Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.222.0

(05/2006)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Multiplexage et synchronisation en transmission

---

Technologies de l'information – Codage générique des images animées et du son associé: Systèmes

Recommandation UIT-T H.222.0

Union internationale des télécommunications

RECOMMANDATIONS UIT-T DE LA SÉRIE H

SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	**H.220–H.229**
Aspects système	H.230–H.239
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
Systèmes et équipements terminaux pour les services audiovisuels	H.300–H.349
Architecture des services d'annuaire pour les services audiovisuels et multimédias	H.350–H.359
Architecture de la qualité de service pour les services audiovisuels et multimédias	H.360–H.369
Services complémentaires en multimédia	H.450–H.499
PROCÉDURES DE MOBILITÉ ET DE COLLABORATION	
Aperçu général de la mobilité et de la collaboration, définitions, protocoles et procédures	H.500–H.509
Mobilité pour les systèmes et services multimédias de la série H	H.510–H.519
Applications et services de collaboration multimédia mobile	H.520–H.529
Sécurité pour les systèmes et services multimédias mobiles	H.530–H.539
Sécurité pour les applications et services de collaboration multimédia mobile	H.540–H.549
Procédures d'interfonctionnement de la mobilité	H.550–H.559
Procédures d'interfonctionnement de collaboration multimédia mobile	H.560–H.569
SERVICES À LARGE BANDE ET MULTIMÉDIAS TRI-SERVICES	
Services multimédias à large bande sur VDSL	H.610–H.619

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

NORME INTERNATIONALE ISO/CEI 13818-1
RECOMMANDATION UIT-T H.222.0

Technologies de l'information – Codage générique des images animées et du son associé: Systèmes

Résumé

La présente Recommandation | Norme internationale spécifie la couche système du codage. Elle a été élaborée en 1994, principalement pour assurer la combinaison et la synchronisation des méthodes de codage vidéo et audio définies dans les parties 2 et 3 de l'ISO/CEI 13818. Depuis 1994, cette norme a été enrichie afin de prendre en charge d'autres spécifications de codage vidéo (ISO/CEI 14496-2 et ISO/CEI 14496-10), d'autres spécifications de codage audio (ISO/CEI 13818-7 et ISO/CEI 14496-3), les flux système (ISO/CEI 14496-1 et ISO/CEI 15938-1), les informations IPMP (ISO/CEI 13818-11) et des métadonnées génériques. La couche système assure six fonctions de base:

1) la synchronisation lors du codage de multiples flux comprimés;
2) l'entrelacement de plusieurs flux comprimés en un seul flux;
3) la réinitialisation des mémoires tampons au début du décodage;
4) la gestion permanente des mémoires tampons;
5) le repérage temporel;
6) le multiplexage et la signalisation des diverses composantes d'un flux système.

Un flux binaire multiplexé conformément à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est soit un **flux de transport** soit un **flux de programme**. Ces deux flux sont construits à partir de **paquet PES** et de paquets contenant les autres informations nécessaires. Ces deux flux permettent le multiplexage de flux vidéo et audio comprimés issus d'un même programme, avec une base de temps commune. Le **flux de transport** permet en outre le multiplexage de flux vidéo et audio comprimés issus de plusieurs programmes, ayant des bases de temps indépendantes. Le **flux de programme** convient en général mieux à des environnements pratiquement exempts d'erreurs, avec possibilité de traitement informatique des informations de programme. Le **flux de transport** convient mieux à des environnements exposés aux erreurs.

Un flux binaire multiplexé conformément à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, que ce soit un flux de transport ou un flux de programme, se compose de deux couches: celle qui est en surface est la couche système, celle qui est la plus profonde est la couche de compression. La couche système remplit les fonctions nécessaires pour utiliser un ou plusieurs flux de données comprimées dans un système. Les parties vidéo et audio de la présente Spécification définissent la couche de compression codée pour les données audio et vidéo. Le codage d'autres types de données n'est pas défini par la présente Recommandation | Norme internationale mais est assuré par la couche système à condition que ces autres types de données soient conformes aux contraintes définies dans la présente Recommandation | Norme internationale.

Source

La Recommandation UIT-T H.222.0 a été approuvée le 29 mai 2006 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Un texte identique est publié comme Norme Internationale ISO/CEI 13818-1.

AVANT-PROPOS

L'Union internationale des télécommunications (UIT) est une institution spécialisée des Nations Unies dans le domaine des télécommunications et des technologies de l'information et de la communication (ICT). Le Secteur de la normalisation des télécommunications (UIT-T) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas des renseignements les plus récents, il est vivement recommandé aux développeurs de consulter la base de données des brevets du TSB sous http://www.itu.int/ITU-T/ipr/.

© UIT 2008

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

Page

SECTION 1 – GÉNÉRALITÉS ... 1
1.1 Domaine d'application ... 1
1.2 Références normatives ... 1
SECTION 2 – ÉLÉMENTS TECHNIQUES ... 2
2.1 Définitions ... 2
2.2 Symboles et abréviations ... 6
2.3 Méthode de description de la syntaxe des flux binaires ... 8
2.4 Prescriptions applicables au flux binaire de transport ... 9
2.5 Prescriptions applicables au flux binaire de programme ... 55
2.6 Descripteurs de programme et d'élément de programme ... 68
2.7 Restrictions appliquées à la sémantique des flux binaires multiplexés ... 99
2.8 Compatibilité avec l'ISO/CEI 11172 ... 104
2.9 Enregistrement des identificateurs de droits d'auteur ... 104
2.10 Enregistrement du format de données privées ... 105
2.11 Acheminement de données ISO/CEI 14496 ... 105
2.12 Acheminement des métadonnées ... 118
2.13 Acheminement des données ISO 15938 ... 128
2.14 Acheminement de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 ... 128
Annexe A – Modèle de décodeur pour le contrôle CRC ... 133
A.0 Décodeur modèle de contrôle CRC ... 133
Annexe B – Commande et contrôle des supports d'enregistrement numérique (DSM-CC) ... 134
B.0 Introduction ... 134
B.1 Eléments généraux ... 135
B.2 Eléments techniques ... 137
Annexe C – Informations spécifiques du programme ... 143
C.0 Explications relatives aux informations spécifiques du programme (PSI) contenues dans un flux de transport ... 143
C.1 Introduction ... 143
C.2 Mécanisme fonctionnel ... 143
C.3 Mappage de sections dans des paquets du flux de transport ... 144
C.4 Fréquences de répétition et accès aléatoire ... 145
C.5 En quoi consiste un programme? ... 145
C.6 Affectation du numéro de programme ... 145
C.7 Utilisation des informations PSI dans un système type ... 146
C.8 Relations entre structures d'informations PSI ... 146
C.9 Affectation spectrale et temps d'acquisition du signal ... 149
Annexe D – Implications de la présente Recommandation | Norme internationale (systèmes) quant au modèle de synchronisation et à ses applications ... 152
D.0 Introduction ... 152
Annexe E – Applications de transmission de données ... 162
E.0 Généralités ... 162
E.1 Suggestion ... 162
Annexe F – Présentation graphique de la syntaxe pour la présente Recommandation | Norme internationale ... 163
F.0 Introduction ... 163
Annexe G – Informations générales ... 169
G.0 Informations générales ... 169
Annexe H – Données privées ... 170
H.0 Données privées ... 170
Annexe I – Conformité des systèmes et interface en temps réel ... 172
I.0 Conformité des systèmes et interface en temps réel ... 172

Page

Annexe J – Interfaçage entre réseaux générateurs de gigue et décodeurs MPEG-2 173
J.0 Introduction 173
J.1 Modèle de conformité de réseau 173
J.2 Spécification de réseau pour le lissage de gigue 174
J.3 Exemples d'implémentations de décodeur 175

Annexe K – Combinaison des flux de transport 177
K.0 Introduction 177
K.1 Les différents types de points de combinaison 177
K.2 Comportement des décodeurs en présence de combinaisons 178

Annexe L – Procédure d'enregistrement (voir 2.9) 180
L.1 Procédure de demande d'un identificateur enregistré (RID, *registered identifier*) 180
L.2 Responsabilités de l'organisme d'enregistrement 180
L.3 Responsabilités des parties demandant un identificateur RID 180
L.4 Procédure d'appel en cas de refus de demande 181

Annexe M – Formulaire de demande d'enregistrement (voir 2.9) 182
M.1 Renseignements de contact sur l'organisation demandant un identificateur enregistré (RID) 182
M.2 Déclaration d'intention d'appliquer l'identificateur RID assigné 182
M.3 Date d'implémentation prévue de l'identificateur RID 182
M.4 Représentant autorisé 182
M.5 Cadre réservé à l'usage officiel de l'organisme d'enregistrement 182

Annexe N 183

Annexe O – Procédure d'enregistrement (voir 2.10) 184
O.1 Procédure de demande d'un identificateur enregistré (RID) 184
O.2 Responsabilités de l'organisme d'enregistrement 184
O.3 Renseignements de contact de l'organisme d'enregistrement 184
O.4 Responsabilités des parties demandant un identificateur RID 184
O.5 Procédure d'appel en cas de refus de demande 184

Annexe P – Formulaire de demande d'enregistrement 186
P.1 Renseignements de contact sur l'organisation demandant un identificateur enregistré (RID) 186
P.2 Demande d'identificateur RID spécifique 186
P.3 Brève description de l'identificateur RID qui est en usage et date de l'implémentation du système 186
P.4 Déclaration d'intention d'appliquer l'identificateur RID assigné 186
P.5 Date d'implémentation prévue de l'identificateur RID 186
P.6 Représentant autorisé 186
P.7 Cadre réservé à l'usage officiel de l'organisme d'enregistrement 186

Annexe Q – Modèles de mémoires tampons T-STD et P-STD pour données audio ADTS ISO/CEI 13818-7 187
Q.1 Introduction 187
Q.2 Débit de débordement de la mémoire tampon de transport 187
Q.3 Capacité de mémoire tampon 187
Q.4 Conclusion 189

Annexe R – Acheminement de scènes ISO/CEI 14496 dans un flux selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 190
R.1 Procédure d'accès au contenu pour des éléments de programme ISO/CEI 14496 dans un flux de programme 190
R.2 Procédure d'accès au contenu pour des éléments de programme ISO/CEI 14496 dans un flux de transport 191

Introduction

La partie relative aux systèmes de la présente Recommandation | Norme internationale traite de la combinaison d'un ou de plusieurs flux élémentaires de données vidéo et audio, ainsi que d'autres types, pour former des flux isolés ou multiples se prêtant à l'enregistrement ou à la transmission. Le codage de ces systèmes suit les règles syntaxiques et sémantiques qui sont imposées par la présente Spécification. Ce codage contient des informations qui permettent un décodage synchronisé du contenu des mémoires des décodeurs, dans une large gamme de conditions d'extraction ou de réception.

Le codage d'un système doit être spécifié sous deux formes: le **flux de transport** et le **flux de programme**. Chacun d'eux est optimisé pour un ensemble d'applications différentes. Ces deux flux, définis dans la présente Recommandation | Norme internationale, acheminent les éléments syntaxiques qui sont nécessaires et suffisants pour synchroniser le décodage et la présentation des informations vidéo et audio, tout en garantissant que les mémoires tampons contenues dans les décodeurs ne subiront ni surremplissage ni sous-remplissage de données. Les informations sont codées sous forme d'éléments syntaxiques faisant appel à des pointeurs temporels (horodateurs) régissant, d'une part, le décodage et la présentation des données codées pour signaux audio et vidéo, d'autre part, l'acheminement du flux de données proprement dit. Les définitions relatives à ces deux flux s'assimilent à des multiplex en mode paquet.

La Figure Intro. 1 illustre la démarche fondamentale – celle du multiplexage – pour les flux élémentaires de données vidéo et audio. Celles-ci sont codées conformément à la Rec. UIT-T H.262 | ISO/CEI 13818-2 et à l'ISO/CEI 13818-3. Les flux élémentaires résultants sont, après compression, mis en **paquets de flux PES**. Les informations nécessaires pour utiliser les paquets de flux PES peuvent être ajoutées au moment de la formation des paquets de flux PES, qu'il s'agisse de flux de transport ou de flux de programme. Ces informations ne sont pas requises et n'ont pas besoin d'être ajoutées si les paquets de flux PES sont recombinés avec des informations de niveau système pour former des **flux de transport** ou des **flux de programme**. La présente norme relative aux systèmes traite des processus situés à droite du trait pointillé vertical de la figure ci-après.

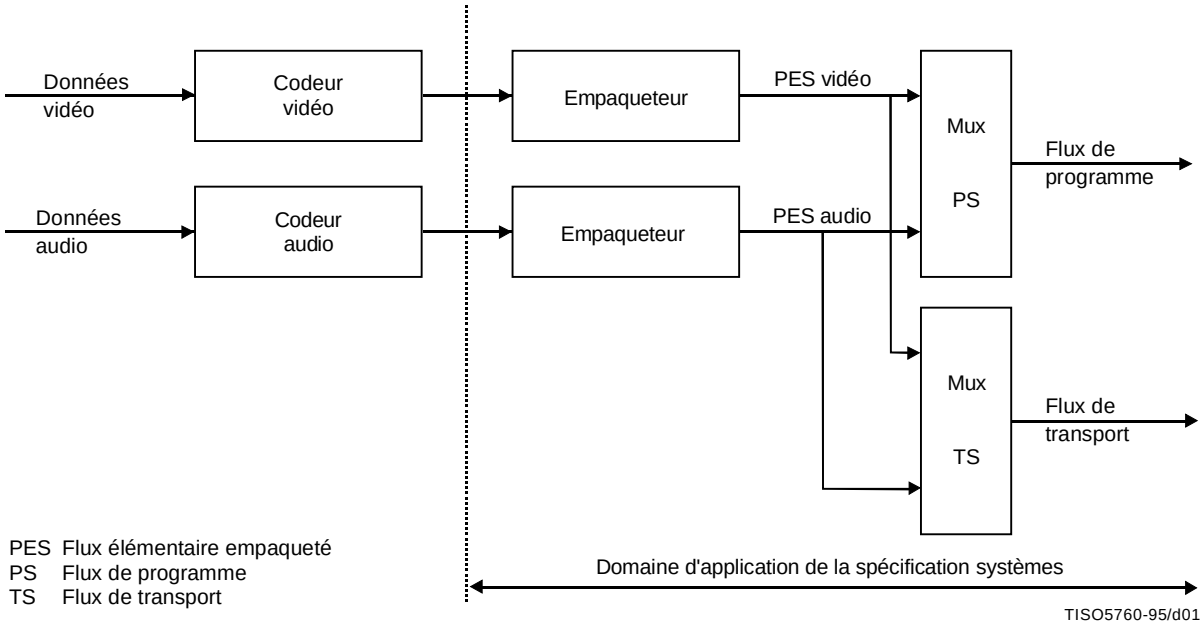


Figure Intro. 1 – Synoptique simplifié du domaine d'application de la présente Recommandation | Norme internationale

Le **flux de programme** est analogue à la couche système de l'ISO/CEI 11172. Il est le résultat de la combinaison, en un seul flux d'un ou de plusieurs flux élémentaires de paquets PES ayant la même base de temps.

Pour les applications qui exigent des flux élémentaires séparés et non multiplexés pour un même programme, on peut également coder ces flux élémentaires sous forme de flux de programme distincts (un par flux élémentaire), possédant la même base de temps. Dans ce cas, les valeurs codées dans les champs de référence temporelle système des divers flux doivent être cohérentes.

De même que le flux de programme isolé, tous les flux élémentaires peuvent être décodés en synchronisme.

Le flux de programme est conçu pour être utilisé dans des environnements relativement exempts d'erreurs. Il convient à des applications pouvant comporter un traitement informatique d'informations du système, comme les applications

multimédias interactives. Les paquets contenus dans le flux de programme peuvent avoir une longueur variable et relativement grande.

Le **flux de transport**, en revanche, combine un ou plusieurs programmes possédant une ou plusieurs bases de temps indépendantes, pour former un flux unique. Les paquets de flux PES contenus dans des flux élémentaires qui forment un programme partagent une base de temps commune. Le flux de transport est conçu pour être utilisé dans des environnements exposés aux erreurs, comme l'enregistrement ou la transmission sur des supports exposés aux erreurs ou aux bruits. Les paquets d'un flux de transport ont une longueur de 188 octets.

Les flux de programme et de transport sont conçus pour différentes applications et leurs définitions ne suivent pas strictement un modèle stratifié. Il est possible et logique de les convertir les uns les autres sans toutefois que l'un soit un sous-ensemble ou un surensemble de l'autre. En particulier, l'extraction du contenu d'un programme dans un flux de transport et la création d'un flux de programme valide sont possibles, grâce au format intermédiaire commun (CIF, *common interchange format*) des paquets de flux PES. Mais le flux de transport ne contient pas tous les champs nécessaires dans un flux de programme et certains doivent être calculés. On peut utiliser le flux de transport pour rassembler une série de couches dans un modèle stratifié. Ce flux est conçu pour être efficace et d'implémentation aisée dans les applications à large bande.

Le domaine d'application des règles exposées dans la présente spécification des systèmes est différent selon qu'elles sont syntaxiques ou sémantiques: les règles syntaxiques ne s'appliquent qu'au codage de la couche système; elles ne s'étendent pas au codage de la couche de compression des données vidéo et audio. En revanche, les règles sémantiques s'appliquent aux combinaisons de flux dans leur ensemble.

La présente spécification des systèmes ne précise pas l'architecture ni l'implémentation des codeurs ou décodeurs, ni celles des multiplexeurs ou démultiplexeurs. Les caractéristiques des flux binaires imposent cependant des exigences relatives au fonctionnement et à la qualité des codeurs, des décodeurs, des multiplexeurs et des démultiplexeurs. Par exemple, les codeurs doivent toujours satisfaire à des exigences minimales en termes de tolérance sur la référence temporelle. Cela, ainsi que d'autres prescriptions, étant mis à part, il existe un degré de liberté considérable pour la conception et l'implémentation des codeurs, décodeurs, multiplexeurs et démultiplexeurs.

Intro. 1 Flux de transport

Le flux de transport est défini de manière à convenir pour la communication ou l'enregistrement d'un ou de plusieurs programmes de données codées, conformément à la Rec. UIT-T H.262 | ISO/CEI 13818-2 et à l'ISO/CEI 13818-3, ainsi que d'autres données dans des environnements exposés à des erreurs notables. De telles erreurs peuvent se manifester sous forme d'erreur sur la valeur d'un bit ou de perte de paquets.

Les flux de transport peuvent être à débit constant ou à débit variable. Dans un cas comme dans l'autre, les flux élémentaires qui les constituent peuvent être à débit constant ou variable. Les contraintes syntaxiques et sémantiques sur le flux sont identiques dans l'un et l'autre cas. Le débit du flux de transport est défini par les valeurs et par les emplacements des champs de la référence temporelle du programme (PCR, *program clock reference*). En général, ces champs sont distincts pour chaque programme.

La construction et l'acheminement d'un flux de transport contenant plusieurs programmes caractérisés par des bases de temps indépendantes, et comportant donc un débit binaire global variable, présentent certaines difficultés. Voir 2.4.2.2.

Le flux de transport peut être construit selon toute méthode qui produit un flux conforme. A partir de flux élémentaires de données codées, de flux de programme ou de flux de transport pouvant eux-mêmes contenir un ou plusieurs programmes, il est possible de construire des flux de transport contenant un ou plusieurs programmes.

Le flux de transport est conçu de telle manière que l'on puisse effectuer sur lui plusieurs opérations avec le moins d'effort possible. Il peut s'agir, par exemple:

1) d'extraire les données codées d'un programme contenu dans le flux de transport, de les décoder et de présenter les résultats comme indiqué sur la Figure Intro. 2;
2) d'extraire, du flux de transport, les paquets de type flux de transport contenus dans un programme donné et de produire, en sortie, un flux de transport différent, ne contenant que ce programme, comme indiqué sur la Figure Intro. 3;
3) d'extraire, d'un ou de plusieurs flux de transport, les paquets de type flux de transport contenus dans un ou plusieurs programmes et de produire, en sortie, un flux de transport différent (non illustré);
4) d'extraire, du flux de transport, le contenu d'un programme isolé et de produire, en sortie, un flux de programme contenant ce programme isolé, comme indiqué sur la Figure Intro. 4;
5) de prendre un flux de programme, de le convertir en un flux de transport pour lui faire traverser un environnement exposé aux pertes puis de récupérer un flux de programme valide et, dans certains cas, identique.

La Figure Intro. 2 et la Figure Intro. 3 montrent des prototypes de systèmes de démultiplexage et de décodage dont l'entrée est un flux de transport. La Figure Intro. 2 illustre le premier cas, dans lequel un flux de transport est directement démultiplexé et décodé. Les flux de transport sont structurés en deux couches:

– une couche système;

– une couche de compression.

Le flux d'entrée dans le décodeur de flux de transport possède une couche système qui est imbriquée dans une couche de compression. Les flux d'entrée dans les décodeurs de données vidéo et de données audio ne possèdent que la couche de compression.

Les opérations effectuées par le décodeur prototype recevant des flux de transport sont applicables soit au flux de transport total (ce sont alors des "opérations à l'échelle du multiplex") soit à des flux élémentaires distincts (ce sont alors des "opérations particulières à un flux"). La couche système du flux de transport se subdivise en deux sous-couches, l'une pour les opérations à l'échelle du multiplex (dite "couche paquet du flux de transport"), l'autre pour les opérations particulières à un flux (dite "couche paquet de flux PES").

Un décodeur prototype pour flux de transport, recevant des données audio et vidéo, est décrit à la Figure Intro. 2 pour montrer la fonction d'un décodeur. L'architecture de celui-ci n'est pas rigide: certaines fonctions de décodeur de couche système, comme la gestion des temps d'un décodeur, peuvent tout aussi bien être réparties entre des décodeurs de flux élémentaire et le décodeur propre au canal. Mais cette Figure Intro. 2 est utile pour poser le problème. De même, on peut effectuer de diverses manières la fonction d'indication – aux décodeurs audio et vidéo particuliers – des erreurs détectées par le décodeur propre au canal: de telles voies de communication ne sont pas représentées sur le schéma. Le prototype de décodeur est conçu de façon à n'impliquer aucune prescription normative quant à la structure d'un décodeur de flux de transport. Des données ni audio ni vidéo peuvent également être présentées en entrée, mais on ne les a pas représentées.

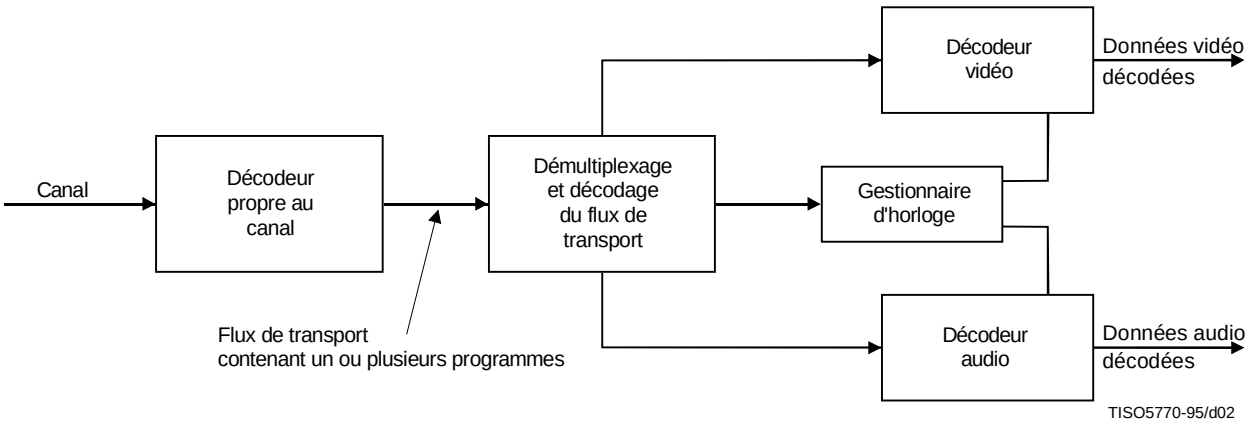


Figure Intro. 2 – Prototype (à titre d'exemple) de multiplexage et de décodage pour le transport

La Figure Intro. 3 illustre le deuxième cas, où un flux de transport contenant plusieurs programmes est converti en un flux de transport ne contenant qu'un seul programme. Dans ce cas, l'opération de remultiplexage exige parfois la correction des valeurs PCR (de référence temporelle programme) afin de tenir compte des changements d'emplacements de référence temporelle programme dans le flux binaire.

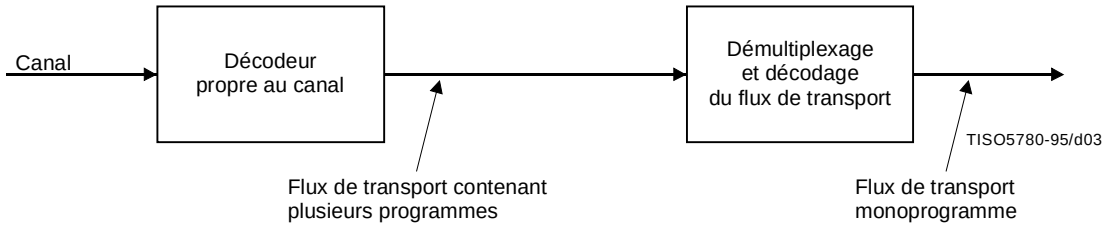


Figure Intro. 3 – Prototype (à titre d'exemple) de multiplexage de flux de transport

La Figure Intro. 4 illustre le cas d'un flux de transport multiprogramme qui est d'abord démultiplexé puis converti en un flux de programme.

Les Figures Intro. 3 et Intro. 4 montrent qu'il est possible et logique de convertir les flux de transport en différents types et configurations. Les syntaxes du **flux de transport** et du **flux de programme** définissent des champs spécifiques qui facilitent les conversions indiquées. Il n'est pas prescrit que des implémentations spécifiques de démultiplexeurs ou de décodeurs comportent toutes ces fonctions.

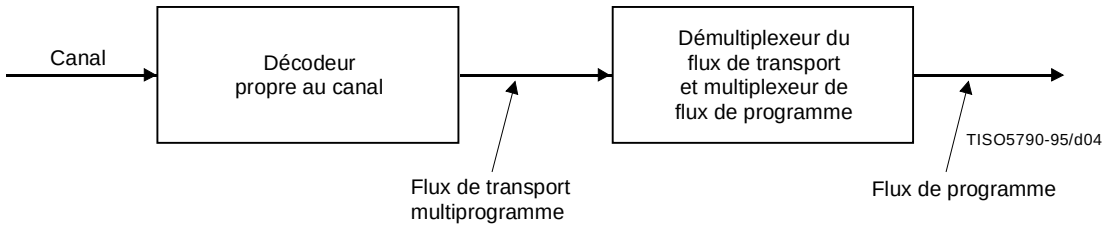


Figure Intro. 4 – Prototype de conversion de flux de transport à flux de programme

Intro. 2 Flux de programme

Le flux de programme est défini de manière à convenir pour la communication ou l'enregistrement d'un seul programme de données codées ou non codées dans des environnements où des erreurs sont très peu probables et où le traitement du codage système, par exemple au moyen d'un logiciel, est un élément important.

Les flux de programme peuvent être à débit constant ou à débit variable. Dans un cas comme dans l'autre, les flux élémentaires qui les constituent peuvent être à débit constant ou variable. Les contraintes syntaxiques et sémantiques sur le flux sont identiques dans l'un et l'autre cas. Le débit du flux de programme est défini par les valeurs et par les emplacements des champs de la référence temporelle du système (SCR, *system clock reference*) et par les champs de structure mux_rate (débit de multiplexage).

Un décodeur prototype pour flux de programme, recevant des données audio et vidéo, est décrit à la Figure Intro. 5. Son architecture n'est pas rigide: certaines fonctions de décodeur de couche système, comme la gestion des temps d'un décodeur, peuvent tout aussi bien être réparties entre des décodeurs de flux élémentaire et le décodeur propre au canal. Mais cette Figure Intro. 5 est utile pour poser le problème. Le prototype de décodeur est conçu de façon à n'impliquer aucune prescription normative quant à la structure d'un décodeur de flux de programme. Des données ni audio ni vidéo peuvent également être présentées en entrée, mais on ne les a pas représentées.

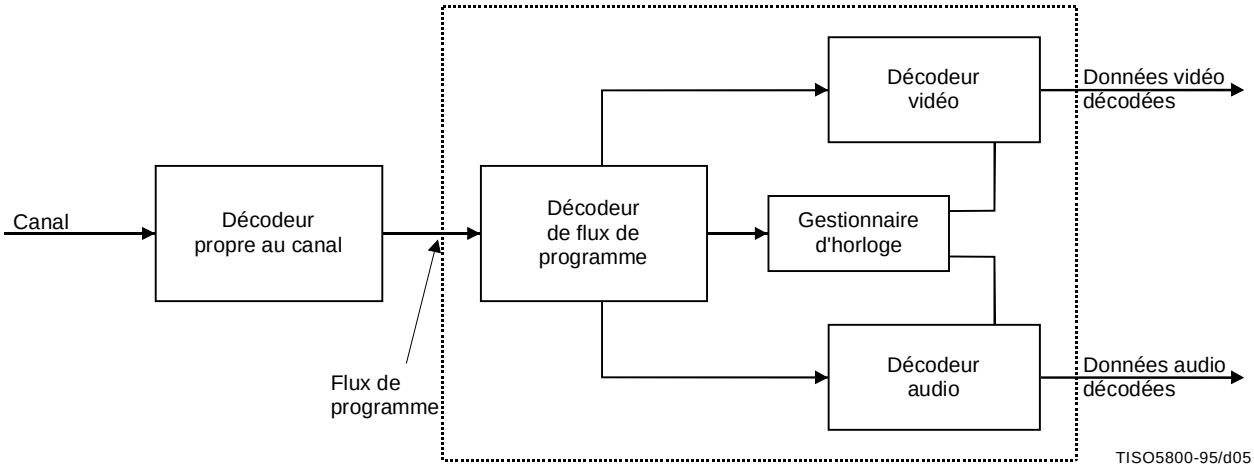


Figure Intro. 5 – Prototype de décodeur pour flux de programme

Le prototype de décodeur pour flux de programme, représenté sur la Figure Intro. 5, se compose de décodeurs de données système, de données vidéo et de données audio conformes, respectivement, aux parties 1, 2 et 3 de ISO/CEI 13818. Dans ce décodeur, la représentation codée et multiplexée d'un ou de plusieurs flux de données audio et vidéo est censée être enregistrée ou communiquée par un canal donné, dans un certain format particulier à ce canal. Ce format n'est pas régi par la présente Recommandation | Norme internationale, non plus que l'étage du décodeur prototype qui est particulier au canal.

Le prototype de décodeur accepte, en entrée, un flux de programme. Il dépend d'un décodeur de flux de programme pour l'extraction, à partir de ce flux, des informations d'ordre temporel. Le décodeur de flux de programme démultiplexe ce flux et les flux élémentaires ainsi obtenus servent d'entrée vidéo et audio dans les décodeurs, dont les sorties se présenteront sous la forme de signaux vidéo et audio décodés. Le système comporte, ce qui n'est pas représenté sur la

figure, le flux d'informations temporelles entre le décodeur de flux de programme, les décodeurs de données vidéo et audio et le décodeur propre au canal. Les décodeurs vidéo et audio sont synchronisés entre eux et avec le canal qui utilise ces informations temporelles.

Les flux de programme sont structurés en deux couches: une couche système et une couche de compression. Le flux d'entrée dans le décodeur de flux de programme possède une couche système qui est imbriquée dans une couche de compression. Les flux d'entrée dans les décodeurs de données vidéo et de données audio ne possèdent que la couche de compression.

Les opérations effectuées par le décodeur prototype recevant des flux de programme sont applicables soit au flux de programme total (ce sont alors des "opérations à l'échelle du multiplex") soit à des flux élémentaires distincts (ce sont alors des "opérations particulières à un flux"). La couche système du flux de programme se subdivise en deux sous-couches, l'une pour les opérations à l'échelle du multiplex (dite "couche bloc de paquets d'un flux de programme"), l'autre pour les opérations particulières à un flux (dite "couche paquet d'un flux PES").

Intro. 3 Conversion entre flux de transport et flux de programme

Conformément aux spécifications des 2.4.1 et 2.5.1 (prescriptions normatives de la présente Recommandation | Norme internationale), il est parfois possible et logique d'effectuer des conversions entre **flux de transport** et **flux de programme** au moyen de paquets de flux PES. Ces paquets peuvent, au prix de certaines contraintes, être mappés directement, du signal de capacité utile d'un certain flux binaire multiplexé, au signal de capacité utile d'un autre flux binaire multiplexé. Il est possible, pour faciliter ce transfert, de repérer l'ordre correct des paquets PES dans un programme, si tous les paquets PES contiennent le compteur de séquence du programme.

Certaines autres informations nécessaires pour effectuer les conversions, par exemple la relation entre flux élémentaires, se trouvent dans des tables et dans des en-têtes des deux flux. Les données de ce type éventuellement disponibles doivent être correctes dans tous les flux, avant et après conversion.

Intro. 4 Flux élémentaire empaqueté

Les **flux de transport** et les **flux de programme** sont chacun composés logiquement de paquets de flux élémentaire empaqueté (PES), comme indiqué dans les définitions d'éléments syntaxiques du 2.4.3.6. Les paquets de flux PES doivent être utilisés pour effectuer des conversions entre flux de transport et flux de programme; dans certains cas de telles conversions, il n'y a pas besoin de modifier les paquets PES. Ceux-ci peuvent avoir une longueur beaucoup plus grande que celle d'un paquet de flux de transport.

Une séquence continue de paquets PES, composant un flux élémentaire donné et possédant un unique identificateur de flux, peut être utilisée afin de construire un flux de type PES. Lorsqu'on utilise des paquets de flux PES pour former un flux PES, ces paquets doivent comporter des champs de référence temporelle du flux élémentaire (ESCR, *elementary stream clock reference*) et des champs de débit de flux élémentaire (ES_rate), avec les contraintes définies au 2.4.3.8. Les données du flux PES doivent se présenter sous la forme d'octets contigus provenant, dans leur ordre initial, du flux élémentaire. Les flux PES ne contiennent pas certaines informations nécessaires au système, qui sont contenues dans les flux de programme et dans les flux de transport. Il s'agira par exemple des informations contenues dans les en-têtes de blocs de paquets, dans les en-têtes système, dans les conditionnements de flux de programme, dans les répertoires de flux de programme, dans les tables de conditionnement du flux de programme et dans certains éléments syntaxiques se rapportant au flux de transport en mode paquet.

Le flux PES est une structure logique qui peut être utile dans certaines implémentations de la présente Recommandation | Norme internationale; il n'est cependant pas défini comme étant un flux destiné aux transferts et aux conversions d'interopérabilité. Les applications qui nécessitent des flux ne contenant qu'un seul flux élémentaire pourront faire appel à des flux de programme ou de transport ne contenant chacun qu'un seul flux élémentaire. Ces flux devront contenir toutes les informations systèmes nécessaires. On pourra construire de multiples flux de programme ou flux de transport contenant chacun un seul flux élémentaire mais ayant une base de temps commune pour acheminer un programme complet, c'est-à-dire avec données audio et données vidéo.

Intro. 5 Modèle temporel

Les parties qui se rapportent aux systèmes, aux données vidéo et aux données audio sont toutes conformes à un modèle temporel dans lequel le retard aux extrémités est constant, depuis l'injection du signal dans un codeur jusqu'à l'extraction de ce signal à la sortie d'un décodeur. Ce retard est la somme des temps nécessaires au codage, à la conservation en mémoire tampon du codeur, au multiplexage, à la communication ou à l'enregistrement, au démultiplexage, à la mise en mémoire tampon du décodeur, au décodage et à la présentation. Dans le cadre de ce modèle de gestion du temps, toutes les images vidéo et tous les échantillons audio sont présentés exactement une seule fois, sauf codage spécifique contraire; d'autre part, l'intervalle interimage et la fréquence des échantillons audio sont identiques dans le codeur et dans le décodeur. Le codage du flux système contient des informations d'horloge qui peuvent être utilisées afin d'implémenter des systèmes impliquant un retard constant aux extrémités. Il est possible d'implémenter des décodeurs qui ne suivent pas exactement ce modèle; dans de tels cas, il appartiendra cependant au décodeur de fonctionner de

manière acceptable. La gestion du temps est incorporée dans les prescriptions normatives de la présente Recommandation | Norme internationale, qui doit être observée par tous les flux binaires valides, quel que soit leur mode de création.

Toutes les informations temporelles sont définies par rapport à une base de temps commune du système, dite "horloge de base du système". Dans le flux de programme, cette horloge peut être en rapport exactement spécifié avec l'horloge vidéo ou l'horloge audio; elle peut également avoir une fréquence de fonctionnement qui diffère légèrement du rapport exact tout en continuant à fournir des pointeurs et des recalages temporels précis de bout en bout.

Dans le flux de transport, la fréquence de l'horloge système est contrainte de manière à toujours être exactement dans le rapport spécifié avec les horloges des échantillons audio et vidéo; l'effet de cette contrainte est de simplifier le recalage sur la fréquence des échantillons à l'intérieur des décodeurs.

Intro. 6 Accès conditionnel

Les flux de données système ici définis autorisent le chiffrement et l'embrouillage afin de créer un accès conditionnel aux programmes codés dans les flux de programme et de transport. Les mécanismes d'accès conditionnel ne sont pas spécifiés ici. Les définitions des flux sont conçues de manière que l'implémentation de systèmes d'accès conditionnel réels soit normalement possible; certains éléments syntaxiques spécifiés fournissent un appui spécifique pour de tels systèmes.

Intro. 7 Opérations à l'échelle du multiplex

Les opérations à l'échelle du multiplex comprennent la coordination de l'extraction des données dans le canal, le réglage des horloges et la gestion des mémoires tampons. Ces tâches sont étroitement liées les unes aux autres. Si le débit d'extraction des données dans le canal est réglable, on peut l'ajuster de manière que les mémoires des décodeurs ne subissent ni surremplissage ni sous-remplissage; mais si le débit binaire n'est pas réglable, les décodeurs de flux élémentaire doivent asservir leur gestion du temps aux données reçues du canal, afin d'éviter tout surremplissage ou sous-remplissage.

Les flux de programme se composent de blocs de données dont les en-têtes facilitent les tâches susmentionnées. Les en-têtes de blocs de données spécifient les instants prévus auxquels chaque octet doit passer du canal au décodeur de flux de programme. Cet ordonnancement des instants d'arrivée prévus servira de référence pour la correction d'horloge et la gestion des mémoires tampons. Il n'est pas nécessaire que les décodeurs suivent exactement cet ordonnancement, mais il faut qu'ils compensent tout écart par rapport à lui.

De la même façon, les flux de transport se composent de paquets de transport dont les en-têtes contiennent des informations spécifiant les instants auxquels chaque octet est censé passer du canal au décodeur de flux de transport. Cet ordonnancement remplit exactement la même fonction que celle qui est spécifiée dans le flux de programme.

Une autre opération à l'échelle du multiplex est l'aptitude d'un décodeur à déterminer quelles sont les ressources requises pour décoder un flux de transport ou un flux de programme. Le premier bloc de paquets de chaque flux de programme apportera des paramètres qui faciliteront l'accomplissement de cette tâche par le décodeur: il s'agira par exemple du débit binaire maximal du flux et du nombre maximal de canaux vidéo présents simultanément dans le canal. De même, le flux de transport contient des informations globalement utiles.

Le flux de transport et le flux de programme contiennent chacun des informations qui indiquent les caractéristiques applicables aux flux élémentaires qui constituent chaque programme, ainsi que les relations entre ces flux. Ces informations peuvent inclure le code de la langue utilisée dans les voies audio ainsi que la relation entre flux vidéo en cas de codage vidéo multicouche.

Intro. 8 Opérations à l'échelle des flux individuels

Les principales opérations à l'échelle des flux individuels sont les suivantes:

1) démultiplexage;
2) lecture synchronisée de plusieurs flux élémentaires.

Intro. 8.1 Démultiplexage

Lors du codage, des flux de programme sont formés par multiplexage de flux élémentaires. Les flux de transport sont formés par multiplexage de flux élémentaires, de flux de programme ou du contenu d'autres flux de transport. Les flux élémentaires peuvent acheminer des données privées, des données réservées et des octets de bourrage, en plus des données audio et vidéo. Ces flux sont subdivisés en paquets qui se succèdent dans le temps. Un paquet de flux PES contient des octets codés issus d'un et d'un seul flux élémentaire.

Le flux de programme peut contenir des paquets de longueur fixe et des paquets de longueur variable, sous réserve des contraintes spécifiées aux 2.5.1 et 2.5.2. Pour les flux de transport, la longueur des paquets est de 188 octets. Des

paquets PES de longueur aussi bien fixe que variable sont autorisés, mais dans la plupart des applications cette longueur sera relativement importante.

Lors du décodage, il est nécessaire de procéder à un démultiplexage afin de reconstituer les flux élémentaires à partir du flux de programme ou de transport multiplexé. Cette opération sera rendue possible par des codes d'identification de flux contenus dans des en-têtes de paquet pour le flux de programme et par des codes d'identification de paquet contenus dans le flux de transport.

Intro. 8.2 Synchronisation

La synchronisation de multiples flux élémentaires est réalisée au moyen d'horodateurs de présentation (PTS, *presentation time stamps*) contenus dans les flux de programme et de transport. Ces horodateurs sont généralement exprimés en unités de 90 kHz mais la référence temporelle du système (SCR), la référence temporelle du programme (PCR) et la référence temporelle du flux élémentaire (ESCR) peuvent faire l'objet d'extensions jusqu'à une résolution de 27 MHz. Le décodage de N flux élémentaires est synchronisé par calage du décodage des flux sur une base de temps principale et commune, plutôt que par recalage d'un flux donné sur un autre. La base de temps principale peut être une des horloges des N décodeurs, l'horloge de la source des données ou une horloge externe définie.

Chaque programme contenu dans un flux de transport, lequel peut contenir plusieurs programmes, peut avoir sa propre base de temps. Les bases de temps de différents programmes contenus dans un même flux de transport peuvent être différentes.

Etant donné que les horodateurs de présentation s'appliquent au décodage de flux élémentaires individuels, ils sont insérés dans la couche paquet de flux PES des flux de programme comme des flux de transport. La synchronisation de bout en bout est réalisée lorsque les codeurs sauvegardent les horodateurs au moment de la capture des données, que ces horodateurs se propagent avec les données codées associées jusqu'aux décodeurs et lorsque ceux-ci les utilisent pour ordonnancer les présentations.

On réalise la synchronisation d'un système de décodage avec un canal en employant la référence SCR dans le flux de programme et son homologue, la référence PCR, dans le flux de transport. Les références SCR et PCR sont des pointeurs temporels qui codent le rythme du flux binaire proprement dit; elles sont calées sur la même base de temps que celle des horodateurs audio et vidéo issus du même programme. Etant donné que chaque programme peut avoir sa propre base de temps, chacun des multiples programmes d'un même flux de transport possédera ses propres champs de référence PCR. Il sera parfois possible que certains programmes aient des champs de référence PCR communs. Le paragraphe 2.4.4: information spécifique du programme (PSI, *program specific information*) indique la méthode de détermination de la référence PCR qui est associée à un programme donné. Chaque programme ne doit avoir qu'une seule base de temps PCR associée.

Intro. 8.3 Relation avec la couche de compression

La couche paquet de flux PES est indépendante de la couche de compression à certains, mais pas à tous égards. Elle en est indépendante en ce sens que les informations utiles contenues dans les paquets PES peuvent ne pas commencer en fonction de codes de déclenchement contenus dans la couche de compression, comme indiqué dans les parties 2 et 3 de ISO/CEI 13818. Par exemple, des codes de déclenchement de données vidéo peuvent apparaître à tout instant dans la capacité utile d'un paquet de flux PES et d'autres codes de déclenchement peuvent être répartis dans des en-têtes de paquet PES. Les horodateurs codés dans des en-têtes de paquet PES s'appliquent cependant aux instants de présentation de structures de la couche de compression (c'est-à-dire aux unités de présentation). En outre, lorsque les données du flux élémentaire sont conformes aux dispositions de la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 13818-3, les données de paquets PES doivent être alignées par les octets en conformité avec la présente Recommandation | Norme internationale.

Intro. 9 Décodeur de référence du système

La première partie de ISO/CEI 13818 fait appel à un "décodeur modèle du système" (STD, *system target decoder*) qui se subdivise en "décodeur modèle du système pour les flux de transport" (T-STD, *transport system target decoder*) (voir 2.4.2) et en "décodeur modèle du système pour les flux de programme" (P-STD, *program system target decoder*) (voir 2.5.2), afin de formaliser les relations de gestion des temps et des mémoires. Etant donné que le décodeur STD est paramétré en termes de champs conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (par exemple pour les capacités de mémoire tampon), chaque flux élémentaire aura son propre paramétrage du décodeur STD. Les codeurs doivent produire des flux binaires qui respectent les contraintes propres aux décodeurs STD correspondants. Les décodeurs physiques pourront partir du principe qu'un flux donné fait l'objet d'une lecture correcte dans son décodeur STD; chaque décodeur physique doit compenser ses écarts de conception par rapport au décodeur STD.

Intro. 10 Applications

Les flux définis dans la présente Recommandation | Norme internationale visent à être aussi utiles que possible dans une large gamme d'applications. Les développeurs d'application sélectionneront simplement le flux le plus approprié.

Les réseaux modernes de communication de données sont sans doute en mesure de prendre en charge des données vidéo UIT-T H.222.0 | ISO/CEI 13818-1 et des données audio ISO/CEI 13818. Un protocole de transport en temps réel est repris. De tels réseaux sont sans doute en mesure de transmettre le flux de programme.

Le flux de programme convient également pour des applications multimédias sur disque compact (CD-ROM). Un traitement informatique du flux de programme sera peut-être approprié.

Le flux de transport conviendra sans doute mieux dans les environnements exposés aux erreurs, comme ceux que l'on rencontre pour acheminer des flux binaires comprimés sur des réseaux à grande distance et dans des systèmes de radiodiffusion.

De nombreuses applications nécessitent l'enregistrement et l'extraction de flux binaires conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 sur divers supports d'enregistrement numérique (DSM, *digital storage media*). Un protocole de commande et contrôle de support d'enregistrement numérique (DSM-CC, *digital storage media command and control*) est spécifié dans l'Annexe B et dans la partie 6 de ISO/CEI 13818 afin de faciliter le pilotage de tels médias.

NORME INTERNATIONALE
RECOMMANDATION UIT-T

Technologies de l'information – Codage générique des images animées et du son associé: Systèmes

SECTION 1 – GÉNÉRALITÉS

1.1 Domaine d'application

La présente Recommandation | Norme internationale spécifie la couche système du codage. Elle a été élaborée principalement pour assurer la combinaison des méthodes de codage vidéo et audio définies dans les parties 2 et 3 de l'ISO/CEI 13818. La couche système assure six fonctions de base:

1) la synchronisation lors du codage de multiples flux comprimés;
2) l'entrelacement de plusieurs flux comprimés en un seul flux;
3) la réinitialisation des mémoires tampons au début du décodage;
4) la gestion permanente des mémoires tampons;
5) le repérage temporel;
6) le multiplexage et la signalisation des diverses composantes d'un flux système.

Un flux binaire multiplexé conformément à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est soit un **flux de transport** soit un **flux de programme**. Ces deux flux sont construits à partir de **paquet PES** et de paquets contenant les autres informations nécessaires. Ces deux flux permettent le multiplexage de flux vidéo et audio comprimés issus d'un même programme, avec une base de temps commune. Le **flux de transport** permet en outre le multiplexage de flux vidéo et audio comprimés issus de plusieurs programmes, ayant des bases de temps indépendantes. Le **flux de programme** convient en général mieux à des environnements pratiquement exempts d'erreurs, avec possibilité de traitement informatique des informations de programme. Le **flux de transport** convient mieux à des environnements exposés aux erreurs.

Un flux binaire multiplexé conformément à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, que ce soit un flux de transport ou un flux de programme, se compose de deux couches: celle qui est en surface est la couche système, celle qui est la plus profonde est la couche de compression. La couche système remplit les fonctions nécessaires pour utiliser un ou plusieurs flux de données comprimées dans un système. Les parties vidéo et audio de la présente Spécification définissent la couche de compression codée pour les données audio et vidéo. Le codage d'autres types de données n'est pas défini par la présente Spécification mais est assuré par la couche système à condition que ces autres types de données soient conformes aux contraintes définies au 2.7.

1.2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations de l'UIT-T en vigueur.

1.2.1 Recommandations | Normes internationales identiques

– Recommandation UIT-T H.262 (2000) | ISO/CEI 13818-2:2000, *Technologies de l'information – Codage des images animées et du son associé: Données vidéo*.

1.2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique

– Recommandation UIT-T.H.264 (2005), *Codage vidéo évolué pour les services audiovisuels génériques*.

ISO/CEI 14496-10:2005, *Technologies de l'information – Codage des objets audiovisuels – Partie 10: Codage visuel avancé.*

– Recommandation UIT-T T.171 (1996), *Protocoles pour les services audiovisuels interactifs: représentation codée des objets multimédias et hypermédias*.

ISO/CEI 13522-1:1997, *Technologies de l'information – Codage de l'information multimédia et hypermédia – Partie 1: Représentation d'objet MHEG – Notation de base (ASN.1)*.

1.2.3 Autres références

– ISO 639-2:1998, *Code pour la représentation des noms de langue – Partie 2: Code alpha-3*.

– ISO 8859-1:1998, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 1: Alphabet latin nº 1*.

– ISO 15706:2002, *Information et documentation – Numéro international normalisé d'œuvre audiovisuelle (ISAN)*.

– ISO/PRF 15706-2, *Information et documentation – Numéro international normalisé d'oeuvre audiovisuelle (ISAN) – Partie 2: Identifiant de version.*

– ISO/CEI 11172-1:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 1: Systèmes*.

– ISO/CEI 11172-2:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 2: Vidéo*.

– ISO/CEI 11172-3:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 3: Audio*.

– ISO/CEI 13818-3:1998, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 3: Son*.

– ISO/CEI 13818-6:1998, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 6: Extensions pour DSM-CC*.

– ISO/CEI 13818-7:2006, *Technologies de l'information – Codage générique des images animées et du son associé – Partie 7: Codage du son avancé (AAC*).

– ISO/CEI 13818-11:2004, *Technologies de l'information – Codage générique des images animées et du son associé – Partie 11: IPMP sur les systèmes MPEG-2*.

– ISO/CEI 14496-1:2004, *Technologies de l'information – Codage des objets audiovisuels – Partie 1: Systèmes*.

– ISO/CEI 14496-2:2004, *Technologies de l'information – Codage des objets audiovisuels – Partie 2: Codage visuel*.

– ISO/CEI 14496-3:2005, *Technologies de l'information – Codage des objets audiovisuels – Partie 3: Codage audio*.

– Recommandation UIT-R BT.601-6 (2007), *Paramètres de codage en studio de la télévision numérique pour des formats standards d'image 4:3 (normalisé) et 16:9 (écran panoramique).*

– Recommandation UIT-R BT.470-7 (2005), *Systèmes de télévision analogique classiques.*

– Recommandation UIT-R BR.648, *Enregistrement numérique des signaux audio*.

– Recommandation UIT-T J.17 (1988), *Préaccentuation utilisée sur les circuits pour transmissions radiophoniques.*

– Publication CEI 60908:1999, *Enregistrement audio – Système audionumérique à disque compact*.

SECTION 2 – ÉLÉMENTS TECHNIQUES

2.1 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent. Si une définition est particulière à une partie, cela est noté entre parenthèses.

2.1.1 unité d'accès (système): représentation codée d'une unité de présentation. Dans le cas de données audio, une unité d'accès est la représentation codée d'une trame audio.

Dans le cas de données vidéo, une unité d'accès comprend toutes les données codées concernant une image, ainsi que tous les octets de bourrage qui la suivent, jusqu'au début (celui-ci non compris) de l'unité d'accès suivante. Si une image n'est pas précédée par un code de déclenchement de groupe ou par un code d'en-tête de séquence, l'unité d'accès commence par le code de déclenchement d'image. Si une image est précédée par un code de déclenchement de groupe et/ou par un code d'en-tête de séquence, l'unité d'accès commence par le premier octet du premier de ces codes de déclenchement. S'il s'agit de la dernière image avant un code de fin de séquence d'un flux de données binaires, tous les octets compris entre le dernier octet de l'image codée et le code de fin de séquence (y compris le code de fin de séquence) font partie de l'unité d'accès.

Concernant la définition d'une unité d'accès pour données vidéo conformes à la Rec. UIT-T H.264 | ISO/CEI 14496-10, voir la définition de l'unité d'accès AVC donnée au 2.1.3.

2.1.2 image 24 heures AVC (système): unité d'accès AVC dont l'instant de présentation se situe au-delà des 24 heures à venir. Dans le cadre de la présente définition, l'unité d'accès AVC n a un instant de présentation qui se situe au-delà des 24 heures à venir si la différence entre l'instant d'arrivée initial $t_{ai}(n)$ et l'instant de sortie du tampon DPB $t_{o,dpb}(n)$ est supérieure à 24 heures.

2.1.3 unité d'accès AVC (système): unité d'accès définie pour les flux d'octets de la Rec. UIT-T H.264 | ISO/CEI 14496-10 avec les contraintes spécifiées au 2.14.1.

2.1.4 tranche AVC (système): structure de données byte_stream_nal_unit définie dans la Rec. UIT-T H.264 | ISO/CEI 14496-10 ayant une valeur nal_unit_type de 1 ou 5, ou structure de données byte_stream_nal_unit ayant une valeur nal_unit_type de 2 ou toute structure de données byte_stream_nal_unit associée ayant une valeur de nal_unit_type de 3 ou 4.

2.1.5 image fixe AVC (système): une image fixe AVC est formée d'une unité d'accès AVC contenant une image IDR, précédée d'unités NAL SPS et PPS qui acheminent suffisamment d'informations pour décoder correctement l'image IDR. Une image fixe AVC sera précédée d'une autre image fixe AVC ou d'une unité NAL de fin de séquence terminant une séquence vidéo codée précédente sauf si l'image fixe AVC correspond à la toute première unité d'accès dans le flux vidéo..

2.1.6 séquence vidéo AVC (système): séquence vidéo codée définie au 3.30 de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

2.1.7 flux vidéo AVC (système): flux conforme à la Rec. UIT-T H.264 | ISO/CEI 14496-10. Un flux vidéo AVC est formé d'une ou de plusieurs séquences vidéo AVC.

2.1.8 débit: débit auquel le flux binaire comprimé est transféré du canal à l'entrée d'un décodeur.

2.1.9 aligné à l'octet: dans un flux binaire codé, un bit est aligné à l'octet si sa position comptée à partir du premier bit de ce flux est un multiple de 8.

2.1.10 canal: support numérique qui enregistre ou transporte un flux conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

2.1.11 image bi-trame à codage B: image B à deux trames ou paire d'images B à trame unique.

2.1.12 image bi-trame codée: image bi-trame à codage I (intra), B (bidirectionnel) ou P (prédictif).

2.1.13 image bi-trame à codage I: image I à deux trames ou paire d'images à trame unique, la première image étant de type I et la deuxième une image de type I ou P.

2.1.14 image bi-trame à codage P: image P à deux trames ou paire d'images P à trame unique.

2.1.15 représentation codée: élément de données qui est représenté sous forme codée.

2.1.16 compression: réduction du nombre de bits utilisés pour représenter une donnée élémentaire.

2.1.17 débit constant: exploitation dans laquelle le débit est constant du début à la fin du flux binaire comprimé.

2.1.18 flux paramétrique contraint du système; CSPS (système) (*constrained system parameter stream*): flux de transport ou de programme auquel s'appliquent les contraintes définies au 2.7.9.

2.1.19 CRC: contrôle de redondance cyclique destiné à vérifier que les données sont correctes.

2.1.20 élément de données: donnée élémentaire telle qu'elle est représentée avant codage et après décodage.

2.1.21 flux décodé: reconstitution décodée d'un flux binaire comprimé.

2.1.22 décodeur: matérialisation d'un processus de décodage.

2.1.23 décodage (processus): processus défini dans la présente Recommandation | Norme internationale, consistant à lire un flux binaire codé entrant et à produire, après décodage, des images ou des échantillons audio.

2.1.24 horodatage de décodage; DTS (système) (*decoding time-stamp*): champ qui peut être présent dans un en-tête de paquet PES et qui indique l'instant du décodage d'une unité d'accès dans le décodeur modèle du système.

2.1.25 support d'enregistrement numérique (DSM, *digital storage media*): dispositif ou système d'enregistrement ou de transmission numérique.

2.1.26 DSM-CC (*digital storage media command and control*): commande et contrôle de support d'enregistrement numérique.

2.1.27 message de commande d'habilitation (ECM, *entitlement control message*): informations privées d'accès conditionnel qui spécifient des mots de commande et éventuellement d'autres paramètres d'embrouillage et de commande qui, en principe, relèvent spécifiquement d'un flux binaire.

2.1.28 message de gestion d'habilitation (EMM, *entitlement management message*): informations privées d'accès conditionnel qui spécifient les niveaux d'autorisation ou les services de décodeurs spécifiques. Elles peuvent concerner des décodeurs simples ou des groupes de décodeurs.

2.1.29 édition: processus par lequel un ou plusieurs flux binaires comprimés sont manipulés afin de produire un nouveau flux binaire comprimé. Les flux binaires édités doivent toujours répondre aux mêmes prescriptions que les flux non édités.

2.1.30 flux élémentaire; ES (système) (*elementary stream*): terme générique désignant un flux binaire de données codées, de type vidéo, audio ou autre contenues dans des paquets PES. Un flux élémentaire est transporté dans une séquence de paquets PES et comporte un seul et unique identificateur de flux.

2.1.31 référence d'horloge de flux élémentaire; ESCR (système) (*elementary stream clock reference*): horodatage contenu dans le flux PES, à partir duquel un décodeur de flux PES peut calculer la base de temps.

2.1.32 codeur: matérialisation d'un processus de codage.

2.1.33 codage (processus): processus, non spécifié dans la présente Recommandation | Norme internationale, qui effectue la lecture d'un train d'images ou d'échantillons audio d'entrée et qui produit un flux binaire valide, tel que défini dans la présente Recommandation | Norme internationale.

2.1.34 codage entropique (ou statistique): codage sans perte à longueur variable de la représentation numérique d'un signal afin de réduire la redondance.

2.1.35 événement: ensemble de flux élémentaires ayant en commun une même base de temps, un même instant de début associé et un même instant de fin associé.

2.1.36 lecture rapide avant (vidéo): processus d'affichage d'une séquence – ou de parties de séquence – d'images dans l'ordre d'affichage, plus rapidement qu'à la fréquence d'image réelle.

2.1.37 interdite (valeur): qualificatif qui, lorsqu'il est utilisé dans les paragraphes de la présente Recommandation | Norme internationale définissant le flux binaire codé, indique que la valeur spécifiée ne doit jamais être utilisée.

2.1.38 métadonnées: informations décrivant le contenu audiovisuel et la nature des données dans un format défini par l'ISO ou tout autre organisme.

2.1.39 unité d'accès aux métadonnées: structure globale interne aux métadonnées qui définit la fraction des métadonnées qui est censée être décodée à un moment donné. La structure interne d'une unité d'accès aux métadonnées est définie par le format des métadonnées.

2.1.40 format d'application des métadonnées: format de l'application qui utilise les métadonnées; indique les informations propres à l'application qui serviront au transport des métadonnées.

2.1.41 information de configuration du décodeur des métadonnées: données dont a besoin un récepteur pour décoder tel ou tel service de métadonnées. Selon le format des métadonnées, l'information de configuration du décodeur peut ou non être nécessaire.

2.1.42 format des métadonnées: format de codage des métadonnées.

2.1.43 service de métadonnées: ensemble cohérent de métadonnées de même format communiquées à un récepteur à une fin précise.

2.1.44 identificateur de service de métadonnées: identificateur d'un service de métadonnées déterminé; utilisé pour certaines méthodes de transport des métadonnées.

2.1.45 flux de métadonnées: concaténation ou ensemble d'unités d'accès aux métadonnées d'un ou de plusieurs services de métadonnées.

2.1.46 flux (multiplexé) (système): flux binaire composé de 0 ou de plus de 0 flux élémentaire(s) combiné(s) de la manière décrite dans la présente Recommandation | Norme internationale.

2.1.47 couche (vidéo et système): un des niveaux dans la hiérarchie de données des spécifications vidéo et système qui est définie dans les parties 1 et 2 de la présente Recommandation | Norme internationale.

2.1.48 bloc (système): un bloc se compose d'un en-tête de bloc, suivi de zéro, un ou plusieurs paquets. Les blocs de données constituent une couche dans la syntaxe du système de codage, décrite au 2.5.3.3.

2.1.49 données en paquet (système): octets de données contigus d'un flux élémentaire présents dans un paquet.

2.1.50 identificateur de paquet; PID (système) (*packet identifier*): valeur unique d'entier qui est utilisée pour associer des flux élémentaires d'un programme afin de constituer un flux de transport monoprogramme ou multiprogramme, comme décrit au 2.4.3.

2.1.51 remplissage (de justification audio): méthode d'ajustement temporel de la longueur moyenne d'une trame audio pour l'aligner sur la durée des échantillons MIC correspondants, consistant à ajouter conditionnellement un intervalle de temps à la trame audio.

2.1.52 charge utile: octets qui suivent les octets d'en-tête dans un paquet. Par exemple, la charge utile d'un paquet de flux de transport comporte les sections d'en-tête de paquet PES et ses octets de données, ou les sections relatives aux champs de pointeur et aux informations spécifiques du programme (PSI), ou les sections relatives aux données privées; mais une capacité utile de paquet PES ne se compose que d'octets de données de paquet PES. L'en-tête de paquet et les champs d'adaptation d'un flux de transport ne constituent pas une charge utile.

2.1.53 PES (système): abréviation du terme Packetized Elementary Stream (flux élémentaire empaqueté).

2.1.54 paquet (de flux) PES (système): structure de données utilisée pour acheminer des données de flux élémentaire. Un tel paquet se compose d'un en-tête de paquet PES, suivi d'un certain nombre d'octets contigus d'un flux élémentaire de données, c'est une couche de la syntaxe de codage du système, comme décrit au 2.4.3.6.

2.1.55 en-tête de paquet de flux PES (système): champs initiaux d'un paquet de flux PES allant jusqu'aux champs d'octets de données du paquet de flux PES, lorsque le flux n'est pas un flux de bourrage. Dans le cas d'un flux de remplissage, l'en-tête de paquet de flux PES est défini également comme étant constitué par les champs initiaux d'un paquet de flux PES, mais n'englobant pas cependant les champs d'octets de bourrage.

2.1.56 flux PES (système): flux composé de paquets PES dont toutes les capacités utiles sont formées de données issues d'un même flux élémentaire et sont désignées par le même identificateur de flux. Des contraintes sémantiques spécifiques sont applicables aux flux PES. Voir Intro. 4.

2.1.57 horodatage de présentation; PTS (système) (*presentation time-stamp*): champ éventuellement présent dans un en-tête de paquet PES qui indique l'heure à laquelle une unité de présentation a été présentée au décodeur modèle du système.

2.1.58 unité de présentation; PU (système) (*presentation unit*): unité d'accès audio décodée ou image décodée.

2.1.59 programme (système): un programme est constitué d'un ensemble d'éléments de programme. Les éléments de programme peuvent être des flux élémentaires. Les éléments de programme ne sont pas nécessairement dotés d'une base de temps définie, ceux qui en sont dotés ont en commun une même base de temps, et sont destinés à une présentation synchronisée.

2.1.60 référence d'horloge de programme; PCR (système) (*program clock reference*): horodatage contenu dans le flux de transport, à partir duquel la base de temps d'un décodeur est calculée.

2.1.61 élément de programme (système): terme générique désignant l'un des flux élémentaires ou l'un des autres flux de données susceptibles d'être inclus dans un programme.

2.1.62 information spécifique du programme; PSI (système) (*program specific information*): données normatives nécessaires au démultiplexage de flux de transport et à la relance concluante de programmes. Elles sont décrites au 2.4.4. Un exemple de PSI privées est le tableau d'information de réseau non obligatoire.

2.1.63 accès aléatoire: processus consistant à commencer la lecture et le décodage du flux binaire codé, à un point arbitraire.

2.1.64 réservée (valeur): dans les articles définissant le flux binaire codé, une valeur est dite "réservée" si elle peut être utilisée dans les futures extensions définies par l'ISO. Sauf indication contraire dans la présente Recommandation | Norme internationale, tous les bits de valeur réservée doivent être mis à '1'.

2.1.65 embrouillage (système): altération des caractéristiques d'un flux de données vidéo, audio ou codées, afin d'empêcher la réception illicite en clair des données en question. Cette altération est un processus spécifié, exécuté sous le contrôle d'un système d'accès conditionnel.

2.1.66 flux d'origine: flux simple non multiplexé d'échantillons avant codage de compression.

2.1.67 combinaison (système): concaténation réalisée au niveau du système de deux flux élémentaires distincts. Le flux système résultant est entièrement conforme aux dispositions de la présente Recommandation | Norme internationale. La combinaison peut créer des discontinuités de la base de temps du compteur de continuité, des informations spécifiques du programme, et du décodage.

2.1.68 codes de déclenchement (système): mots codés à 32 bits incorporés dans le flux binaire codé. Ces codes servent à plusieurs fins, y compris l'identification de certaines des structures syntaxiques de codage. Les codes de déclenchement se composent d'un préfixe de 24 éléments binaires (0x000001) et d'un identificateur de flux sur 8 éléments binaires, comme indiqué dans le Tableau 2-22.

2.1.69 mémoire d'entrée du décodeur modèle de système (système): tampon de type premier entré/premier sorti qui est placé à l'entrée d'un décodeur modèle de système pour enregistrer des données comprimées issues de flux élémentaires, avant leur décodage.

2.1.70 image fixe: une image fixe se compose d'une séquence vidéo, codée selon la Rec. UIT-T H.262 | ISO/CEI 13818-2, l'ISO/CEI 11172-2 ou l'ISO/CEI 14496-2, contenant exactement une image à codage intratrame. Une horodate PTS est associée à cette image et, en cas de codage conforme à l'ISO/CEI 11172-2, la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou l'ISO/CEI 14496-2, l'instant de présentation des images suivantes (si elles existent) est d'au moins deux périodes d'image après celui de l'image fixe.

2.1.71 en-tête de système (système): structure de données qui est définie au 2.5.3.5 et qui achemine des informations résumant les caractéristiques de couche système d'un flux de programme selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

2.1.72 référence d'horloge du système; SCR (système) (*system clock reference*): horodatage contenu dans le flux de programme, à partir duquel la base de temps d'un décodeur est calculée.

2.1.73 décodeur modèle du système; STD (système) (*system target decoder*): modèle fictif de référence représentant un processus de décodage et utilisé pour décrire la sémantique d'un flux binaire multiplexé selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

2.1.74 horodatage (système): élément indiquant l'instant d'une action spécifique telle que l'arrivée d'un octet ou la présentation d'une unité PU.

2.1.75 en-tête de paquet de flux de transport (système): premiers champs d'un paquet de flux de transport, y compris le champ réservé au compteur de continuité.

2.1.76 débit variable: attribut d'un flux de transport ou d'un flux de programme pour lequel le débit du flux binaire entrant dans le décodeur varie avec le temps.

2.2 Symboles et abréviations

Les opérateurs mathématiques utilisés pour décrire la présente Recommandation | Norme internationale sont similaires aux opérateurs employés dans le langage de programmation C. Les divisions d'entiers avec troncature et les arrondissages seront toutefois définis spécifiquement. Les opérateurs au niveau du bit sont définis dans l'hypothèse d'une représentation en complément à deux des nombres entiers. Les boucles de numérotation et de comptage commencent généralement à zéro.

2.2.1 Opérateurs arithmétiques

+	Addition
–	Soustraction (dans le cas d'un opérateur binaire) ou négation (dans le cas d'un opérateur unaire)
+ +	Incrémentation
– –	Décrémentation
* ou ×	Multiplication
^	Puissance
/	Division entière avec troncature. Par exemple, les fractions 7/4 et –7/–4 sont tronquées à 1 et les fractions –7/4 et 7/–4 sont tronquées à –1

// Division entière avec arrondi à l'entier le plus proche. Les valeurs à mi-intervalle sont arrondies par excès, sauf indication contraire. Par exemple, 3//2 sera arrondi à 2 et –3//2 sera arrondi à –2

DIV Division entière avec arrondi à l'entier (algébriquant) inférieur

% Opérateur de congruence, défini seulement pour des nombres positifs

Sign( ) $\text{Sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x == 0 \\ -1 & x < 0 \end{cases}$

NINT( ) Opérateur d'arrondi. Prend un argument réel et envoie en retour l'entier le plus proche de cet argument. Les valeurs à mi-intervalle d'entier sont arrondies par excès

sin Sinus

cos Cosinus

exp Exponentielle

√ Racine carrée

$\log_{10}$ Logarithme base 10

$\log_e$ Logarithme népérien

2.2.2 Opérateurs logiques

|| OU logique

&& ET logique

! NON logique

2.2.3 Opérateurs relationnels

> Supérieur à

≥ Supérieur ou égal à

< Inférieur à

≤ Inférieur ou égal à

= = Egal à

!= Différent de

max [,...,] Valeur maximale dans la liste d'arguments

min [,...,] Valeur minimale dans la liste d'arguments

2.2.4 Opérateurs au niveau des bits

& ET

| OU

>> Décalage à droite avec remplissage par bit de signe

<< Décalage à gauche avec remplissage de zéros

2.2.5 Affectation

= Opérateur d'affectation

2.2.6 Mnémoniques

Les mnémoniques suivants sont définis pour décrire les différents types de données utilisés dans le flux binaire codé.

bslbf Chaîne binaire, bit de gauche en premier, où "gauche" désigne l'ordre dans lequel les chaînes binaires sont écrites dans la présente Recommandation | Norme internationale. Ces chaînes s'écrivent comme une suite de 1 et de 0 entre deux apostrophes, par exemple '1000 0001'. Les espacements à l'intérieur d'une chaîne binaire sont destinés à faciliter la lecture et n'ont pas de signification

ch Canal

gr Granule de 3 * 32 échantillons de sous-bande en couche audio II ou de 18 * 32 échantillons de sous-bande en couche audio III

main_data	La portion main_data du flux binaire contient les facteurs de normalisation, les données à codage de Huffman et des informations auxiliaires
main_data_beg	Indique l'emplacement, dans le flux binaire, du début de la portion main_data pour la trame. Cet emplacement est égal à celui de la fin de la portion main_data de la trame précédente, plus 1 bit. Il est calculé d'après la valeur de la structure main_data_end de la trame précédente
part2_length	Indique le nombre de bits contenus dans la portion main_data pour définir les facteurs de normalisation
rpchof	Coefficients restants du polynôme, dans l'ordre décroissant
sb	Sous-bande
scfsi	Informations pour la sélection du facteur de normalisation
switch_point_l	Numéro de la bande de facteurs de normalisation (bande des facteurs de normalisation en blocs longs) à partir de laquelle s'effectue la commutation de pointage sur fenêtre
switch_point_s	Numéro de la bande de facteurs de normalisation (bande des facteurs de normalisation en blocs courts) à partir de laquelle s'effectue la commutation de pointage sur fenêtre
tcimsbf	Entier signé, en format de complément à deux, bit le plus significatif (signe) en premier
uimsbf	Entier non signé, bit le plus significatif en premier
vlclbf	Code de longueur variable, bit de gauche en premier, où "gauche" désigne l'ordre dans lequel les codes de longueur variable sont écrits
window	Nombre d'intervalles de temps effectifs dans le cas où block_type = = 2, avec $0 \leq$ window ≤ 2.

Dans les mots à plusieurs octets, l'octet le plus significatif est en premier.

2.2.7 Constantes

π	3,14159265359
e	2,71828182845

2.3 Méthode de description de la syntaxe des flux binaires

Les flux binaires récupérés par le décodeur sont décrits au 2.4.1 et au 2.5.1. Chaque donnée élémentaire du flux binaire est écrite en caractères gras. Elle est décrite par son nom, par sa longueur en bits et par un symbole mnémonique indiquant son type et son ordre de transmission.

L'action provoquée par un élément de données décodé dans un flux binaire dépend de la valeur de cet élément et des éléments de données déjà décodés. Le décodage des éléments de données et la définition des variables d'état utilisées pour ce décodage sont décrits dans les articles donnant une présentation sémantique de la syntaxe. Les structures suivantes – en caractères non enrichis – sont créées pour exprimer les conditions affectant les éléments de données présents.

La présente syntaxe utilise la convention du code "C" selon laquelle une variable ou expression évaluée comme différente de zéro est équivalente à un état de véracité.

while (condition) {
data_element
. . .
}
	Si la condition est vraie, le groupe d'éléments de données apparaîtra le premier dans le flux de données, et ce jusqu'à ce que la condition ne soit plus vraie.

do {
data_element
. . . }
while (condition)
	L'élément de données apparaît toujours au moins une fois. L'élément de données est répété jusqu'à ce que la condition ne soit plus vraie.

if (condition) {
data_element
. . .
}
	Si la condition est vraie, le premier groupe d'éléments de données apparaîtra en premier dans le flux de données.

else { **data_element** . . .	Si la condition n'est pas vraie, le deuxième groupe d'éléments de données apparaîtra en premier dans le flux de données.

}

for (i = 0; i < n; i++) {
  data_element
  . . .
}

Le groupe d'éléments de données apparaît n fois. L'insertion de structures conditionnelles dans ce groupe d'éléments de données peut dépendre de la valeur de la variable i de commande d'itérations, qui est mise à zéro pour la première apparition, incrémentée à un pour la deuxième apparition, et ainsi de suite.

Comme indiqué plus haut, le groupe d'éléments de données peut contenir des structures conditionnelles imbriquées. Par souci de concision, les accolades { } seront omises s'il n'y a qu'un seul élément de données en cause.

data_element [] L'élément de données [] est une table de données. Le nombre d'éléments de données est indiqué par le contexte.

data_element [n] L'élément de données [n] est le n+1^{e} élément d'une table de données.

data_element [m][n] L'élément de données [m][n] est le m+1,n+1^{e} élément d'une table de données à deux dimensions.

data_element [1][m][n] L'élément de données [1][m][n] est le 1+1,m+1,n+1^{e} élément d'une table de données à trois dimensions.

data_element [m...n] L'élément de données contient la série allant du bit m au bit n.

Bien que la syntaxe soit exprimée en termes de procédures, il convient de ne pas en déduire que soit la Figure 2-1 soit la Figure 2-2 implémente une procédure de décodage satisfaisante. Ces figures définissent en particulier un flux binaire d'entrée correct et sans erreur. Les décodeurs réels devront toujours comporter la possibilité de détecter les codes de déclenchement et les octets de synchronisation (flux de transport) de façon à commencer correctement le décodage et à identifier les erreurs, les effacements ou les insertions au cours du décodage. Les méthodes de détection de tels états et les mesures à prendre à leur sujet ne sont pas normalisées.

2.4 Prescriptions applicables au flux binaire de transport

2.4.1 Structure de codage et paramètres du flux de transport

La couche de codage du flux de transport de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 permet de combiner un ou plusieurs programmes afin de former un seul flux. Les données issues de chaque flux élémentaire sont multiplexées avec les informations qui permettent une présentation des flux élémentaires en synchronisme avec un programme.

Un flux de transport se compose d'un ou de plusieurs programme(s). Les flux élémentaires sont constitués d'unités d'accès.

Les données de flux élémentaire sont acheminées dans des paquets de flux PES. Un paquet PES se compose d'un en-tête suivi de données de paquet. Les paquets PES s'insèrent dans les paquets du flux de transport. Le premier octet de chaque en-tête de paquet PES est placé au premier emplacement disponible dans la capacité utile d'un paquet de transport.

L'en-tête de paquet PES commence par un préfixe de début de paquet codé sur 32 bits qui identifie également le flux ou le type de flux auquel les données de paquet appartiennent. Cet en-tête de paquet PES peut contenir des horodateurs de décodage et de présentation (DTS et PTS). L'en-tête de paquet PES contient aussi d'autres champs facultatifs. Les données d'un paquet PES contiennent un nombre variable d'octets contigus issus d'un même flux élémentaire.

Les paquets de type flux de transport commencent par un préfixe de 4 octets, qui contient un identificateur de paquet (PID) codé sur 13 éléments binaires (défini dans le Tableau 2-2). L'identificateur PID désigne, au moyen de tables d'informations spécifiques de programme (PSI) le contenu des données se trouvant dans le paquet de transport. Les paquets de transport possédant la même valeur de PID acheminent des données appartenant à un et un seul flux élémentaire.

Les tables d'informations PSI sont acheminées dans le flux de transport. Il existe six tables PSI:

- table d'association de programmes;
- table de conditionnement de programme;
- table d'accès conditionnel;
- table d'informations sur le réseau;
- table de description de flux de transport;
- table d'informations de commande IPMP.

Ces tables contiennent les informations nécessaires et suffisantes pour démultiplexer et présenter les programmes. La table de conditionnement de programme présentée dans le Tableau 2-33 spécifie, entre autres informations, quels identificateurs PID et donc quels flux élémentaires sont associés pour former chaque programme. Cette table indique aussi l'identificateur PID des paquets de flux de transport qui acheminent la référence PCR de chaque programme. La présence de la table d'accès conditionnel est indispensable en cas d'application d'un processus d'embrouillage. L'usage de la table d'informations sur le réseau est facultatif et la présente Recommandation | Norme internationale ne spécifie pas son contenu. La table d'informations de commande IPMP doit être présente lorsque les informations IPMP définies dans l'ISO/CEI 13818-11 sont utilisées par une des composantes du flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Les paquets de transport peuvent être vides. Les paquets vides sont destinés au bourrage de justification des flux de transport. Ils peuvent être insérés ou supprimés par des processus de remultiplexage: on ne peut donc pas partir du principe que la capacité utile des paquets vides sera acheminée jusqu'au décodeur.

La présente Recommandation | Norme internationale ne spécifie pas les données codées qui peuvent être utilisées dans le cadre de systèmes d'accès conditionnel. Elle propose cependant des mécanismes permettant aux diffuseurs de programmes de transporter et d'identifier ces données pour le traitement de décodage et pour repérer correctement les données spécifiées par la présente Spécification. Ce type d'appui est offert, d'une part, au moyen des structures en mode paquet de transport et, d'autre part, au moyen de la table d'accès conditionnel aux informations PSI (voir Tableau 2-32 des informations spécifiques de programme).

2.4.2 Décodeur modèle du système pour le flux de transport

La sémantique du flux de transport (spécifiée au 2.4.3) et les contraintes sur cette sémantique (spécifiées au 2.7) nécessitent des définitions exactes des événements d'arrivée et de décodage des octets ainsi que des moments auxquels ces événements se produisent. Les définitions requises sont développées dans la présente Recommandation | Norme internationale au moyen d'un décodeur théorique appelé décodeur modèle du système pour le flux de transport (T-STD, *transport stream system target decoder*). L'Annexe D (informative) donne des explications complémentaires sur le décodeur T-STD.

Le décodeur T-STD est un modèle théorique qui sert à définir précisément ces événements et à donner une représentation du processus de décodage pendant la construction ou la vérification des flux de transport. Le décodeur T-STD n'est défini qu'à cette fin. Il existe trois types de décodeurs T-STD: le schéma de la Figure 2-1 représente un exemple de décodeur vidéo, audio et systèmes. Ni son architecture, ni l'ordonnancement chronologique décrits n'excluent la possibilité d'une lecture ininterrompue et synchronisée de flux de transport issus de divers décodeurs possédant des architectures différentes et des ordonnancements chronologiques différents.

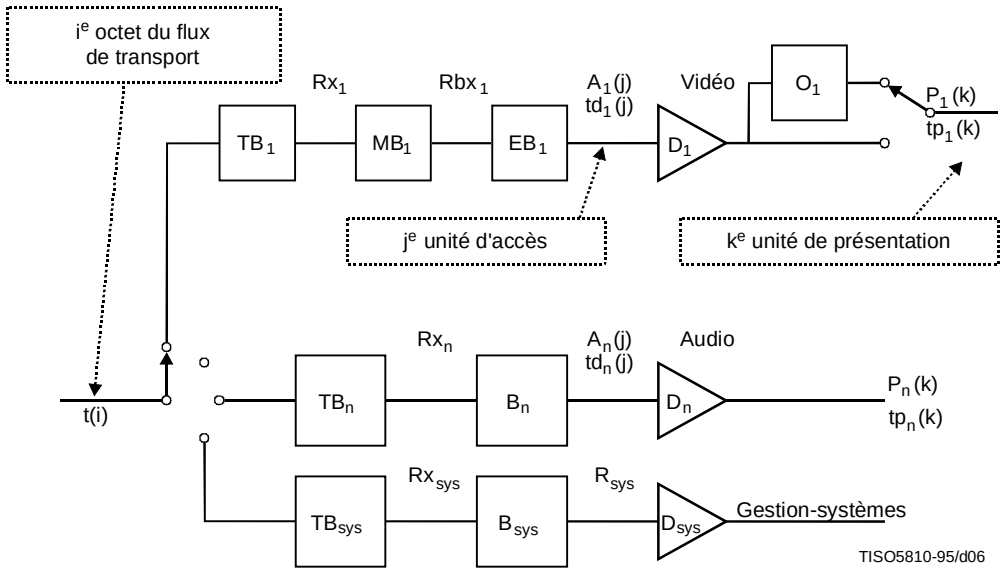


Figure 2-1 – Notation d'un décodeur modèle de système pour un flux de transport

La notation suivante (partiellement illustrée dans la Figure 2-1) est utilisée pour décrire le décodeur modèle du système pour le flux de transport:

i, i′, i″ sont les indices attribués aux octets dans le flux de transport. Le premier octet a l'indice 0.

j est un index pointant sur des unités d'accès dans les flux élémentaires.

k, k′, k″ sont des index pointant sur des unités de présentation dans les flux élémentaires.

n est un index pointant sur les flux élémentaires.

p est un index pointant sur les paquets du flux de transport.

t(i) indique, en secondes, l'instant auquel le n^e octet du flux de transport pénètre dans le décodeur modèle du système. La valeur t(0) est une constante arbitraire.

PCR(i) est le temps codé dans le champ de référence PCR, mesuré en unités de la période d'horloge système de 27 MHz, où i est l'indice désignant l'octet final du champ program_clock_reference_base.

$A_n(j)$ est la j^e unité d'accès dans le flux élémentaire n. Les $A_n(j)$ sont indexées dans l'ordre de décodage.

$td_n(j)$ indique, en secondes, l'instant auquel le décodeur modèle du système décode la j^e unité d'accès dans le flux élémentaire n.

$P_n(k)$ est la k^e unité de présentation dans le flux élémentaire n. $P_n(k)$ résulte du décodage de $A_n(j)$; les $P_n(k)$ sont indexées dans l'ordre de décodage.

$tp_n(k)$ indique, en secondes, l'instant de présentation, dans le décodeur modèle du système, de la k^e unité de présentation dans le flux élémentaire n.

t est le temps, mesuré en secondes.

$F_n(t)$ est le remplissage, mesuré en octets, de la mémoire tampon d'entrée dans le décodeur modèle du système pour le flux élémentaire n à l'instant t.

B_n est la mémoire principale pour le flux élémentaire n, présente uniquement pour les flux élémentaires audio.

BS_n est la capacité, en octets, de la mémoire principale B_n.

B_{sys} est la mémoire principale du décodeur STD destinée aux informations systèmes pour le programme qui est en cours de décodage.

BS_{sys} est la capacité, en octets, de B_{sys}.

MB_n est la mémoire de multiplexage, destinée au flux élémentaire n; présente uniquement pour les flux élémentaires vidéo.

MBS_n est la capacité, en octets, de MB_n.

EB_n est la mémoire de flux élémentaire destinée au flux élémentaire n; présente uniquement dans le cas des flux élémentaires vidéo.

EBS_n est la capacité, en octets, de EB_n.

TB_{sys} est la mémoire tampon de transport destinée aux informations systèmes pour le programme qui est en cours de décodage.

TBS_{sys} est la capacité, mesurée en octets, de TB_{sys}.

TB_n est la mémoire tampon de transport pour le flux élémentaire n.

TBS_n est la capacité, mesurée en octets, de TB_n.

D_{sys} est le décodeur pour les informations systèmes contenues dans le flux de programme n.

D_n est le décodeur pour le flux élémentaire n.

O_n est le tampon de réordonnancement pour le flux élémentaire de données vidéo n.

R_{sys} est le débit d'extraction des données de la mémoire B_{sys}.

Rx_n est le débit d'extraction des données de la mémoire TB_n.

Rbx_n est le débit d'extraction de la mémoire MB_n des données de charge utile des paquets PES, en cas d'utilisation de la méthode du débit de sortie. Il est défini uniquement pour les flux élémentaires vidéo.

$Rbx_n(j)$ est le débit d'extraction de la mémoire MB_n des données de charge utile des paquets PES, en cas d'utilisation de la méthode du délai du vérificateur VBV. Il est défini uniquement pour les flux élémentaires vidéo.

Rx_{sys} est le débit d'extraction des données de la mémoire TB_{sys}.

R_{es} est le débit du flux élémentaire vidéo codé dans un en-tête de séquence.

2.4.2.1 Fréquence de l'horloge système

Les informations de rythme indiquées dans le décodeur T-STD sont acheminées par plusieurs champs de données, définis dans la présente Spécification (voir 2.4.3.4 et 2.4.3.6). Ces informations sont codées dans les champs de référence PCR sous la forme de valeur échantillonnée d'horloge système pour un programme. Les champs de référence PCR sont acheminés dans le champ d'adaptation des paquets de flux de transport avec une valeur d'identificateur de

paquet égale au PID de champ PCR défini dans la section de table de conditionnement de programme pour le flux de transport du programme en cours de décodage.

En pratique, les décodeurs peuvent reconstituer cette base de temps d'après ces valeurs et leurs instants d'arrivée respectifs. On trouvera ci-dessous les contraintes minimales qui s'appliquent à la fréquence d'horloge système pour les flux de programme, telle que représentée par les valeurs des champs de référence PCR lorsqu'elles sont reçues par un décodeur.

La valeur de fréquence d'horloge système est mesurée en hertz et doit répondre aux contraintes suivantes:

$$27\,000\,000 - 810 \leq \text{system_clock_frequency} \leq 27\,000\,000 + 810$$

$$\text{dérive de la fréquence d'horloge système: } \leq 75 \times 10^{-3} \text{ Hz/s}$$

NOTE – Les sources de données codées doivent respecter une tolérance plus stricte afin de faciliter le fonctionnement satisfaisant des appareils domestiques d'enregistrement et de lecture.

Une fréquence d'horloge système pour programme peut être plus précise que prescrit ci-dessus. Une telle précision améliorée peut être transmise au décodeur au moyen du descripteur d'horloge système (voir 2.6.20).

Les débits définis dans la présente Spécification sont mesurés en fonction de la fréquence d'horloge système. Par exemple, un débit de 27 000 000 bits par seconde dans le décodeur T-STD correspondrait au transfert d'un octet de données tous les huit (8) cycles de l'horloge système.

La notation "system_clock_frequency" est utilisée plusieurs fois dans la présente Spécification pour désigner la fréquence d'une horloge répondant à ces prescriptions. Par commodité de notation, les équations mettant en jeu une référence PCR, un pointeur PTS ou un pointeur DTS fournissent des valeurs dont la précision est un multiple entier (en secondes) du rapport (300×2^{33}/system_clock_frequency). Ce rapport est dû au codage sur 33 bits des informations de calage de la référence PCR sur 1/300e de la fréquence d'horloge du système, plus 9 bits pour le reste des informations et au codage sur 33 bits de la fréquence d'horloge du système, divisée par 300 pour les pointeurs PTS et DTS.

2.4.2.2 Entrée dans le décodeur modèle du système pour le flux de transport

Les données d'entrée dans le décodeur modèle du système pour le flux de transport (T-STD) ont la forme d'un flux de transport. Un tel flux peut contenir plusieurs programmes ayant des bases de temps indépendantes. Le décodeur T-STD ne décodera cependant qu'un seul programme à la fois. Dans le modèle T-STD, toutes les indications de rythme se rapportent à la base de temps de ce programme.

Les données issues du flux de transport pénètrent dans le décodeur T-STD à un débit discrètement constant. Le ie octet entre au temps t(i). L'instant auquel cet octet entre dans le décodeur T-STD peut être déduit du flux d'entrée par décodage des champs de référence temporelle de programme (PCR) à l'entrée, codés dans le champ d'adaptation de paquet de transport du programme à décoder, et par comptage des octets du flux de transport complet intercalés entre les champs PCR successifs du programme à décoder. Le champ PCR est codé (voir équation 2-1) en deux parties: d'une part, en unités de période correspondant à 1/300e de la fréquence d'horloge système dite PCR_base (voir équation 2-2) et, d'autre part, en unités de période correspondant à la fréquence d'horloge système dite PCR_extension (voir équation 2-3). Les valeurs codées dans ces deux parties sont calculées par PCR_base(i) (voir équation 2-2) et PCR_ext(i) (voir équation 2-3) respectivement. La valeur codée dans le champ PCR indique l'instant t(i) auquel l'indice i renvoie à l'octet contenant le dernier bit du champ PCR_base.

Plus précisément:

$$PCR(i) = PCR_base(i) \times 300 + PCR_ext(i) \tag{2-1}$$

où:

$$PCR_base(i) = ((system_clock_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-2}$$

$$PCR_ext(i) = ((system_clock_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-3}$$

Pour tous les autres octets, on calcule l'instant d'arrivée à l'entrée, t(i) dans l'équation 2-4 ci-dessous, sur la base de la référence PCR(i″) et du débit d'arrivée des données de transport, ce débit de transport étant déterminé comme étant le nombre d'octets du flux de transport intercalés entre les octets contenant le dernier bit de deux trames PCR successives pour le même programme, divisé par la différence entre les valeurs codées dans deux mêmes champs de référence PCR.

$$t(i) = \frac{PCR(i'')}{system_clock_frequency} + \frac{i - i''}{transport_rate\,(i)} \quad (2\text{-}4)$$

où:

i est l'indice d'un octet du flux de transport pour $i'' < i < i'$;

i″ est l'indice de l'octet contenant le dernier bit du plus récent champ de base program_clock_reference applicable au programme en cours de décodage;

PCR(i″) est le temps codé dans les champs de base et d'extension de type champ de référence temporelle du programme, en unités d'horloge système.

Le débit de transport est donné par le rapport suivant:

$$transport_rate(i) = \frac{((i' - i'') \times system_clock_frequency)}{PCR(i') - PCR(i'')} \quad (2\text{-}5)$$

où:

i′ est l'indice de l'octet contenant le dernier bit du champ de base program_clock_reference immédiatement suivant, applicable au programme en cours de décodage.

NOTE – $i'' < i \leq i'$.

En cas de discontinuité de la base de temps, signalée par l'indicateur de discontinuité dans le champ d'adaptation des paquets de transport, la définition fournie dans les équations 2-4 et 2-5 pour l'instant d'arrivée des octets à l'entrée du décodeur T-STD n'est pas applicable entre le dernier champ PCR de l'ancienne base de temps et le premier champ PCR de la nouvelle base de temps. Dans ce cas, l'instant d'arrivée de ces octets est déterminé à partir de l'équation 2-4, en remplaçant néanmoins le débit de transport par la valeur applicable entre le dernier et l'avant-dernier champ PCR de l'ancienne base de temps.

La tolérance sur la référence PCR est définie comme l'inexactitude maximale admissible dans les références PCR reçues; cette inexactitude est sans doute imputable à l'imprécision des valeurs de PCR et au recalage de PCR en cours de multiplexage. Elle n'inclut pas les erreurs de temps d'arrivée de paquet en raison notamment d'une gigue du réseau. Cette tolérance est de ± 500 ns.

Dans le décodeur modèle T-STD, l'inexactitude sera répercutée dans le calcul du débit de transport par la relation 2-5.

Flux de transport pour plusieurs programmes et à débit variable

Les flux de transport peuvent contenir plusieurs programmes ayant des bases de temps indépendantes. Puisque à chacun de ces programmes indépendants doivent correspondre des séries distinctes de champs PCR, comme l'indiquent les valeurs d'identificateurs PID des champs PCR respectifs, les champs PCR ne peuvent avoir les mêmes coordonnées. Les données d'un programme issues du flux de transport pénètrent dans le décodeur T-STD à un débit constant par paliers. Par conséquent, si le débit du flux de transport varie, ces variations ne peuvent se produire qu'au niveau des champs PCR du programme considéré. Puisque les champs PCR et donc les points du flux de transport correspondant à des variations de débit n'ont pas les mêmes coordonnées, le débit d'entrée des données du flux de transport dans le décodeur T-STD devrait varier en fonction du programme qui y pénètre. Aussi est-il impossible de construire un plan d'acheminement pour la totalité d'un flux de transport, lorsque celui-ci contient plusieurs programmes ayant chacun des bases de temps indépendantes, et lorsque le débit du flux est variable. Par contre, la construction de flux de transport à débit constant et contenant plusieurs programmes à débit variable ne pose pas de difficulté.

2.4.2.3 Mise en mémoire tampon

Les paquets de flux de transport complets, contenant des informations relatives au système, pour le programme choisi pour le décodage, entrent dans le tampon du système de transport, TB_{sys}, avec un débit égal à celui du flux de transport. Parmi ces paquets, il y a les paquets de flux de transport dont les valeurs des identificateurs PID sont 0, 1, 2 ou 3, et tous les paquets du flux de transport identifiés via la table d'association de programmes (voir Tableau 2-30) comme ayant la valeur program_map_PID pour le programme choisi. Les données contenues dans la table d'informations sur le réseau (NIT) spécifiée par l'identificateur PID de la table NIT ne sont pas transférées vers le tampon TB_{sys}.

NOTE 1 – La taille de la table d'informations de commande IPMP peut être importante, et la fréquence de répétition de cette table devrait être ajustée pour respecter les spécifications du tampon.

Tous les octets qui entrent dans la mémoire TB_n sont extraits au débit Rx_n spécifié ci-dessous. Les octets qui font partie du paquet de flux PES ou de son contenu sont injectés dans la mémoire principale B_n. Les autres octets ne le sont pas et peuvent être utilisés pour piloter le système. Les paquets de transport en double ne sont pas injectés dans la mémoire B_n, MB_n, ou B_{sys}.

Le tampon TB_n est vidé comme suit:

- lorsqu'il n'y a pas de données dans le tampon TB_n, le débit Rx_n est égal à zéro;
- sinon, pour des données vidéo:

$$Rx_n = 1,\ 2 \times R_{\max}[profil, niveau]$$

où

R_{max}[profil, niveau] est spécifié en fonction du profil et du niveau déterminés selon le Tableau 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2. Ce tableau spécifie la limite supérieure de chaque flux élémentaire de données vidéo, dans le cadre d'un profil et d'un niveau spécifiques;

Rx_n est égal à 1, 2 × R_{max} pour les flux vidéo à paramètres contraints selon ISO/CEI 11172-2, où R_{max} se rapporte au débit maximal indiqué dans ISO/CEI 11172-2 pour un flux binaire à paramètres contraints.

Pour des données audio ADTS ISO/CEI 13818-7:

Nombre de canaux	**Rx_n [bit/s]**
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Canaux: il s'agit du nombre de canaux de sortie audio pleine bande plus le nombre d'éléments de canaux de couplage commutés indépendamment dans le même flux audio élémentaire. Par exemple, dans le cas type où il n'y a pas d'élément de canaux de couplage commutés indépendamment, le son monophonique correspond à 1 canal, le son stéréophonique à 2 canaux et le son ambiophonique 5.1 à 5 canaux (le canal pour les effets à basses fréquences (LFE) n'est pas compté).

Pour les autres données audio:

$$Rx_n = 2 \times 10^6 \text{ bit/s}$$

Pour des données système:

$$Rx_n = 1 \times 10^6 \text{ bit/s}$$

Le débit Rx_n est mesuré par rapport à la fréquence d'horloge du système.

Les paquets de flux de transport complets, contenant des informations relatives au système, pour le programme choisi pour le décodage, entrent dans le tampon du système de transport, TB_{sys}, avec un débit égal à celui du flux de transport. Parmi ces paquets, il y a les paquets de flux de transport dont les valeurs des identificateurs PID sont 0, 1, 2 et 3 (en cas de présence), et tous les paquets du flux de transport identifiés via la table d'association de programmes (voir Tableau 2-30) comme ayant la valeur program_map_PID pour le programme choisi. Les données contenues dans la table d'informations sur le réseau (NIT) spécifiée par l'identificateur PID de la table NIT ne sont pas transférées vers le tampon TB_{sys}.

Les octets sont extraits de TB_{sys} au débit Rx_{sys} et injectés dans la mémoire B_{sys}. Chaque octet est transféré instantanément.

Les paquets de transport en double ne sont pas injectés dans la mémoire B_{sys}.

Les paquets de transport qui n'entrent ni dans la mémoire TB_n ni dans TB_{sys}, sont perdus.

La capacité de la mémoire tampon de transport est de 512 octets.

Les capacités de mémoires tampons de flux élémentaire EBS_1 à EBS_n sont définies dans le cas des données vidéo comme étant égales à la capacité de la mémoire tampon VBV (vbv_buffer_size) telle qu'elle est acheminée dans l'en-tête de séquence. (Voir le résumé "paramètres contraints" dans l'ISO/CEI 11172-2 et le Tableau 8-14 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.)

Les capacités de mémoires tampons de multiplexage MBS_1 à MBS_n sont définies comme suit pour les données vidéo:

Pour le niveau inférieur et le niveau principal:

$$MBS_n = BS_{mux} + BS_{oh} + VBV_{\max}[profil, niveau] - vbv_buffer_size$$

où BS_{oh}, tampon de surdébit de paquet est défini par:

$$BS_{oh} = (1/750)\,\text{secondes} \times R_{\max}[profil, niveau]$$

et BS_{mux}, tampon de multiplexage supplémentaire par:

$$BS_{mux} = 0{,}004\ \text{secondes} \times R_{\max}[profil, niveau]$$

tandis que VBV_{max}[profil, niveau] est défini au Tableau 8-14 de la Rec. UIT-T H.262 | ISO/CEI 13818-2 et R_{max}[profil, niveau] au Tableau 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, la capacité du tampon VBV étant transportée dans l'en-tête de séquence. (Voir 6.2.2 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.)

Pour le niveau supérieur 1440 et le niveau supérieur:

$$MBS_n = BS_{mux} + BS_{oh}$$

où BS_{oh} est défini par:

$$BS_{oh} = (1/750)\ \text{secondes} \times R_{\max}[profil, niveau]$$

et BS_{mux} est défini par:

$$BS_{mux} = 0{,}004\,\text{secondes} \times R_{\max}[profil, niveau]$$

tandis que R_{max}[profil, niveau] au Tableau 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.

Pour les flux paramétriques binaires contraints de l'ISO/CEI 11172-2:

$$MBS_n = BS_{mux} + BS_{oh} + vbv_\max - vbv_buffer_size$$

où BS_{oh} est défini par:

$$BS_{oh} = (1/750)\,\text{secondes} \times R_{\max}$$

et BS_{mux} est défini par:

$$BS_{mux} = 0{,}004\,\text{secondes} \times R_{\max}$$

tandis que R_{max} et vbv_max désignent respectivement le débit maximal et la capacité maximale de tampon VBV pour un flux paramétrique binaire contraint de l'ISO/CEI 11172-2.

Une portion BS_{mux} = 4 ms × R_{max}[profil, niveau] de la MBS_n, est attribuée à la mémoire de multiplexage. Le reste de cette capacité est disponible pour la capacité $BS_{oh,}$ mais peut aussi être consacré au multiplexage initial.

NOTE 2 – Le remplissage de mémoire par le surdébit de paquets PES est directement retenu dans les flux PES par le décodeur PES-STD défini au 2.5.2.4. Il est possible, mais non nécessaire, de construire des flux de transport à partir de flux PES.

Mémoires principales BS_n

Les capacités des mémoires principales BS_1 à BS_n sont définies comme suit:

Audio

Pour des données audio ADTS ISO/CEI 13818-7:

Nombre de canaux	BS_n [octets]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Canaux: il s'agit du nombre de canaux de sortie audio pleine bande plus le nombre d'éléments de canaux de couplage commutés indépendamment dans le même flux audio élémentaire. Par exemple, dans le cas type où il n'y a pas d'élément de canaux de couplage commutés indépendamment, le son monophonique correspond à 1 canal, le son stéréophonique à 2 canaux et le son ambiophonique 5.1 à 5 canaux (le canal pour les effets à basses fréquences (LFE) n'est pas compté).

Pour les autres données audio:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \ octets$$

Les capacités du tampon d'unité d'accès, BS_{dec}, et du tampon de surdébit de paquets PES, BS_{oh}, sont soumises à la contrainte suivante:

$$BS_{dec} + BS_{oh} \leq 2848 \ octets$$

Une portion (736 octets) de la mémoire de 3584 octets est attribuée au multiplexage. Le reste, 2848 octets, est partagé entre la mémoire tampon d'unité d'accès (BS_{dec}), la mémoire tampon de surdébit de paquets PES (BS_{oh}) et une mémoire de multiplexage supplémentaire.

Systèmes

La mémoire principale (B_{sys}) pour données systèmes a une capacité (BS_{sys}) de 1536 octets.

Vidéo

Dans le cas des flux élémentaires vidéo, le transfert de données de MB_n à EB_n utilise l'une des deux méthodes ci-après: la méthode dite "de débordement" et celle du délai du vérificateur VBV.

Méthode de débordement

La méthode dite "de fuite" ou de débordement a pour effet de transférer les données de MB_n à EB_n, suivant un débit de débordement R_{bx}. Cette méthode est employée dès lors que l'une des conditions suivantes est vérifiée:

- le descripteur de décodeur STD (voir 2.6.32) relatif au flux élémentaire ne figure pas dans le flux de transport;
- le descripteur de décodeur est présent et la valeur du fanion leak_valid (de validité de méthode de débordement) est égale à '1';
- le descripteur de décodeur est présent, la valeur du fanion leak_valid est égale à '0' et la valeur 0xFFFF a été affectée aux champs de délai du vérificateur VBV codés dans le flux vidéo;
- le mode enrichissement est à "Vrai" (voir 2.4.3.7).

Pour le niveau inférieur et le niveau principal:

$$Rbx_n = R_{\max}\,[profil,\ niveau]$$

Pour le niveau 1440 supérieur et le niveau supérieur:

$$Rbx_n = Min\{1{,}05 \times R_{es}, R_{\max}[profil,\ niveau]\}$$

Pour les flux paramétriques binaires contraints de l'ISO/CEI 11172-2:

$$Rbx_n = 1,\ 2 \times R_{\max}$$

où R_{max} désigne le débit maximal pour un flux paramétrique binaire contraint de l'ISO/CEI 11172-2.

Si le tampon MB_n contient des données de capacité utile de paquet PES, et si la mémoire EB_n n'est pas complètement remplie, la capacité utile de paquet PES est transférée de MB_n à EB_n suivant un débit égal à Rbx_n; si la mémoire EB_n est pleine, les données ne sont cependant pas extraites de MB_n. En cas de transfert d'un octet de données de MB_n à EB_n, tous les octets d'en-tête de paquet PES présents dans MB_n et précédant immédiatement l'octet en question sont instantanément extraits et ignorés. Si la mémoire MB_n ne contient aucune donnée de capacité utile de paquet PES, aucune donnée n'en est extraite. Toutes les données introduites dans MB_n en sortent. Tous les octets de données de capacité utile de paquet PES sont instantanément introduits dans EB_n dès qu'ils quittent MB_n.

Méthode du délai de vérificateur VBV

La méthode dite du délai de vérificateur VBV a pour effet de spécifier précisément l'instant du transfert de chaque octet de données codées vidéo de MB_n vers EB_n, d'après les valeurs de délai de vérificateur VBV codées dans le flux élémentaire vidéo. La méthode du délai de vérificateur VBV est utilisée dans tous les cas où le descripteur STD (voir 2.6.32) relatif à ce flux élémentaire est présent dans le flux de transport, la valeur du fanion leak_valid est égale à '0' et la valeur 0xFFFF n'a pas été affectée aux champs de délai de vérificateur VBV codés dans le flux vidéo. Il suffit qu'un seul champ de délai de vérificateur VBV dans une séquence vidéo prenne une valeur différente de 0xFFFF, pour qu'aucun champ de délai de vérificateur VBV de cette séquence ne puisse prendre la valeur 0xFFFF (voir l'ISO/CEI 11172-2 et la Rec. UIT-T H.262 | ISO/CEI 13818-2).

En cas d'utilisation de la méthode du délai de vérificateur VBV, le transfert de MB_n à EB_n du dernier octet du code de déclenchement de l'image vidéo relatif à l'image j s'effectue à l'instant $td_n(j)$ – délai VBV(j), où $td_n(j)$ désigne l'instant de décodage de l'image j, défini plus haut, et délai VBV(j) représente la valeur du délai de vérificateur, exprimée en secondes, indiquée dans le champ de délai VBV de l'image j. Le transfert des données depuis les derniers octets des codes successifs de déclenchement d'image vidéo (y compris le dernier octet du deuxième code de déclenchement) vers la mémoire tampon EB_n s'effectue à raison d'un débit constant par paliers $R_{bx}(j)$, spécifié pour chaque image j. Ainsi, le débit $R_{bx}(j)$ d'introduction des données dans ce tampon est donné par la formule:

$$R_{bx}(j) = NB(j) \ / \ (vbv_delay(j) - vbv_delay(j + 1) + td_n(j + 1) - td_n(j)) \qquad (2\text{-}6)$$

où NB(j) est le nombre d'octets compris entre les derniers octets des codes de déclenchement d'image (y compris le dernier octet du deuxième code de déclenchement) des images j et j + 1, hormis les octets d'en-tête de paquet PES.

NOTE 3 – Délai VBV(j + 1) et $td_n(j + 1)$ peuvent avoir des valeurs différentes de celles qui sont normalement escomptées pour des affichages vidéo périodiques en cas de mise à '1' du fanion de faible délai low_delay dans l'extension de séquence vidéo. L'examen du flux binaire ne permet pas toujours de déterminer les valeurs correctes.

La valeur du débit $R_{bx}(j)$ fournie par l'équation 2-6 doit être inférieure ou égale à R_{max}[profil, niveau] pour les flux élémentaires de type 0x02 (voir Tableau 2-34), dont la définition figure dans la Rec. UIT-T H.262 | ISO/CEI 13818-2, et inférieure ou égale au débit maximal admis dans le jeu de paramètres soumis à des contraintes, défini pour les flux élémentaires de type 0x01 (voir l'ISO/CEI 11172-2).

En cas de transfert d'un octet de données de MB_n à EB_n, tous les octets d'en-tête de paquet PES présents dans MB_n et précédant immédiatement l'octet en question sont instantanément extraits et ignorés. Toutes les données introduites dans MB_n en sortent. Tous les octets de données de capacité utile de paquet PES sont instantanément introduits dans EB_n dès qu'ils quittent MB_n.

Retrait des unités d'accès

Pour chaque mémoire tampon de flux élémentaire EB_n et pour la mémoire principale B_n, toutes les données d'unité d'accès qui sont restées le plus longtemps dans la mémoire d'entrée, $A_n(j)$, et tous les octets de bourrage éventuels qui les précèdent immédiatement, présents dans ladite mémoire d'entrée à l'instant $td_n(j)$ sont retirés aussitôt, à l'instant $td_n(j)$. L'instant de décodage $td_n(j)$ est spécifié dans les champs des horodateurs DTS ou PTS (voir 2.4.3.6). Les instants de décodage, $td_n(j + 1)$, $td_n(j + 2)$, ... des unités d'accès sans champs horodateurs codés DTS ou PTS, suivant directement une unité d'accès j, peuvent être déduits des informations contenues dans le flux élémentaire. Voir l'Annexe C de la Rec. UIT-T H.262 | ISO/CEI 13818-2, ISO/CEI 13818-3 ou ISO/CEI 11172, ainsi que 2.7.5. Dans le cas d'un flux de données audio, tous les en-têtes de paquet PES qui sont mémorisés immédiatement avant l'unité d'accès ou qui sont incorporés dans les données de cette unité d'accès sont simultanément extraits lors du retrait de l'unité d'accès. Lorsque celle-ci est extraite de la mémoire, elle est instantanément décodée pour devenir une unité de présentation.

Données systèmes

Dans le cas des données du système, celles-ci sont extraites de la mémoire principale, B_{sys}, à un débit R_{sys} chaque fois qu'il y a au moins 1 octet disponible dans la mémoire B_{sys}.

$$R_{sys} = \max (80\,000\ bits/s, transport_rate(i) \times 8\ bits/octet / 500) \quad (2\text{-}7)$$

NOTE 4 – L'augmentation du débit R_{sys} en présence de débits de transport élevés a pour objet de permettre d'augmenter le débit de données pour la table PSI d'informations spécifiques de programme.

Faible délai

Lorsque le fanion low_delay est mis à '1' dans l'extension de séquence vidéo (voir 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2), la mémoire EB_n peut être sous-remplie. Dans ce cas et si la mémoire EB_n du flux élémentaire pour le décodeur T-STD est examinée à l'instant spécifié par $td_n(j)$, les données pour l'unité d'accès peuvent ne pas être toutes présentes dans la mémoire EB_n. Si cela se produit, la mémoire doit être réexaminée à intervalles de deux périodes de trame, jusqu'à ce que la mémoire contienne les données pour l'unité d'accès complète. A cet instant, toute l'unité d'accès doit être extraite de la mémoire EB_n, qui ne doit jamais déborder.

En mode de faible délai, et lorsque la valeur du fanion est mise à '1', un sous-remplissage de la mémoire EB_n est autorisé sans limite de durée. Le décodeur T-STD doit extraire de la mémoire EB_n les données d'unité d'accès le plus tôt possible selon l'alinéa précédent et en fonction d'éventuelles valeurs d'horodateur DTS ou PTS codées dans le flux binaire. On notera que le décodeur ne pourra sans doute pas rétablir des temps corrects de décodage et d'affichage, tels qu'indiqués par les horodateurs DTS et PTS, avant que cesse la situation de sous-remplissage de la mémoire EB_n et qu'un horodateur PTS ou DTS soit trouvé dans le flux binaire.

Mode d'enrichissement

Lorsque le fanion DSM_trick_mode (voir 2.4.3.6) est mis à '1' dans l'en-tête de paquet PES contenant le début d'une unité d'accès vidéo à une image de type B (à codage bidirectionnel) et lorsque le champ trick_mode_control est mis à '001' (ralenti) ou à '010' (arrêt sur image) ou à '100' (retour lent), l'unité d'accès aux images de type B n'est pas extraite de la mémoire tampon EB_n de données vidéo avant la dernière fois (s'il y en a plusieurs) où une trame quelconque de l'image est décodée et présentée. La répétition de la présentation de trames et d'images est définie au 2.4.3.8 sous les rubriques relatives au ralenti, au retour lent et au contrôle d'identification de trame (field_id_cntrl). L'unité d'accès est extraite instantanément de la mémoire EB_n à l'instant indiqué, qui dépend de la valeur du champ rep_cntrl.

Lorsque le fanion DSM_trick_mode est mis à '1' dans l'en-tête de paquet PES d'un paquet contenant le premier octet d'un code de déclenchement d'image, le statut du mode d'enrichissement passe à "Vrai" lors de l'extraction de la mémoire EB_n du code de déclenchement d'image du paquet de flux PES. Le statut du mode d'enrichissement reste "Vrai" jusqu'à ce que le décodeur T-STD reçoive un en-tête de paquet PES où le fanion DSM_trick_mode est mis à '0' et jusqu'à l'extraction, de la mémoire EB_n, du premier octet du code de déclenchement d'image qui commence après cet en-tête de paquet PES. Lorsque le mode d'enrichissement est à "Vrai", la mémoire EB_n peut être sous-remplie. Toutes les autres contraintes issues de flux normaux sont conservées lorsque le mode d'enrichissement est activé.

2.4.2.4 Décodage

Les flux élémentaires mémorisés dans les tampons B_1 à B_n et EB_1 à EB_n sont décodés instantanément par les décodeurs D_1 à D_n. Ces flux peuvent être retardés dans les mémoires de réordonnancement O_1 à O_n avant d'être présentés au spectateur en sortie de décodeur T-STD. Les tampons de réordonnancement ne sont utilisés qu'en présence d'un flux élémentaire de données vidéo si certaines unités d'accès ne sont pas acheminées dans l'ordre de présentation. Ces unités d'accès devront être réordonnancées avant d'être présentées. En particulier, si l'image $P_n(k)$ est une image de type I (à codage intratrame) ou une image de type P (à codage prédictif) acheminée avant une ou plusieurs images de type B, cette image doit être retardée dans le tampon de réordonnancement, O_n, du décodeur T-STD avant d'être présentée. Toute image déjà mémorisée dans le tampon O_n sera présentée avant que l'image actuelle puisse être mémorisée. L'image $P_n(k)$ doit normalement être retardée jusqu'au décodage de l'image I ou P suivante. Pendant son stockage dans le tampon de réordonnancement, les images B suivantes seront décodées et présentées.

L'instant de présentation au spectateur d'une unité de présentation $P_n(k)$ est $tp_n(k)$. Pour les unités de présentation qui n'exigent pas de retard pour réordonnancement, l'instant $tp_n(k)$ est égal au temps $td_n(j)$ puisque les unités d'accès sont décodées instantanément; c'est par exemple le cas pour des images bitrames de type B. Pour les unités de présentation qui sont retardées, les instants $tp_n(k)$ et $td_n(j)$ ont pour différence le temps pendant lequel l'image $P_n(k)$ est retardée dans le tampon de réordonnancement, ce temps étant un multiple de la période d'image nominale. Il convient de veiller à utiliser un retard de réordonnancement adéquat à partir du début des flux élémentaires de données vidéo, afin de répondre aux exigences du flux entier. Par exemple, un flux qui n'a initialement que des images de type I et de type P mais qui ultérieurement comporte des images de type B devra normalement avoir un retard de réordonnancement dès son début.

La Rec. UIT-T H.262 | ISO/CEI 13818-2 explique plus en détail le réordonnancement des images vidéo.

2.4.2.5 Présentation

La fonction d'un système de décodage est de reconstruire des unités de présentation à partir de données comprimées et de les présenter en séquence synchronisée, aux instants de présentation corrects. Bien qu'en réalité les appareils de présentation audio et vidéo aient généralement des retards finis et différents, ainsi que d'éventuels retards supplémentaires imposés par le post-traitement ou par des fonctions de sortie, le décodeur modèle du système considère que ces retards sont nuls.

Dans le décodeur T-STD représenté à la Figure 2-1, l'affichage d'une unité de présentation (image) se produit à l'instant de présentation $tp_n(k)$.

Dans le décodeur T-STD, la sortie d'une unité de présentation audio commence à son instant de présentation, $tp_n(k)$, dès que le décodeur présente le premier échantillon. Les échantillons suivants de l'unité de présentation seront présentés en séquence, à la fréquence d'échantillonnage audio.

2.4.2.6 Gestion de la mémoire tampon

Les flux de transport doivent être construits de façon que les conditions énoncées dans le présent paragraphe soient satisfaites. Le présent paragraphe utilise la notation définie pour le décodeur modèle de système.

Il ne doit jamais se produire un surremplissage des mémoires tampons TB_n et TB_{sys}. TB_n et TB_{sys} doivent être vidées au moins une fois par seconde. La mémoire B_n ne doit jamais être ni surremplie, ni sous-remplie. B_{sys} ne doit pas déborder.

La mémoire EB_n ne peut être sous-remplie qu'en mode de faible délai, lorsque le fanion correspondant est mis à '1' dans l'extension de séquence vidéo (voir 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2) ou quand le mode d'enrichissement est à "Vrai".

En cas d'application de la méthode du débit de sortie à la spécification des transferts de données, la mémoire MB_n ne doit jamais déborder et doit être vidée au moins une fois par seconde. EB_n ne doit pas déborder.

En cas d'application de la méthode du délai de vérificateur VBV à la spécification des transferts de données, la mémoire MB_n ne doit ni déborder, ni être sous-remplie, et la mémoire EB_n ne doit pas déborder.

Le temps de passage dans les mémoires tampons du décodeur modèle du système doit être inférieur ou égal à 1 seconde, sauf en ce qui concerne les données vidéo photographiques et les flux ISO/CEI 14496. C'est-à-dire: $td_n(j) - t(i) \leq$ 1 seconde, pour toute valeur de j, et pour tout octet i de l'unité d'accès $A_n(j)$.

Pour les données d'images vidéo fixes, le temps de passage doit vérifier la relation $td_n(j) - t(i) \leq 60$ secondes, pour toute valeur de j et pour tout octet i de l'unité d'accès $A_n(j)$.

Pour les flux ISO/CEI 14496, le temps de passage est contraint par la relation $td_n(j) - t(i) \leq 10$ secondes pour tous les j et pour tous les octets i dans l'unité d'accès $A_n(j)$.

Définition du débordement et du survidage

Soit $F_n(t)$ le remplissage instantané du tampon B_n du décodeur T-STD.

$F_n(t) = 0$ immédiatement avant $t = t(0)$.

Il n'y a pas débordement si pour tout t et pour tout n:

$$F_n(t) \leq BS_n$$

Il n'y a pas survidage pour tout t et pour n:

$$0 \leq F_n(t)$$

2.4.2.7 Extensions du décodeur T-STD pour l'acheminement de données ISO/CEI 14496

Le modèle du décodeur T-STD est étendu pour le décodage des données ISO/CEI 14496 acheminées dans un flux de transport. Les paramètres T-STD utilisés pour le décodage de flux élémentaires distincts ISO/CEI 14496 sont définis au 2.11.2 et ceux qui sont utilisés pour le décodage des scènes et des flux associés ISO/CEI 14496 au 2.11.3.

2.4.2.8 Extensions du décodeur T-STD pour l'acheminement de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10

Pour définir le codage au niveau du décodeur T-STD des flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 acheminés dans un flux de transport, il faut étendre le modèle du décodeur T-STD. L'extension du décodeur T-STD et les paramètres du décodeur T-STD nécessaires au décodage des flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 sont définis au 2.14.3.1.

2.4.3 Spécification de la syntaxe et de la sémantique du flux de transport

La syntaxe suivante décrit un flux d'octets. Les paquets du flux de transport doivent avoir une longueur de 188 octets.

2.4.3.1 Flux de transport

Voir Tableau 2-1.

Tableau 2-1 – Flux de transport

Syntaxe	Nombre de bits	Mnémonique

MPEG_transport_stream() {
do {
transport_packet()
} while (nextbits() == sync_byte)
}		

2.4.3.2 Couche paquet du flux de transport

Voir Tableau 2-2.

Tableau 2-2 – Paquet de transport selon la présente Recommandation | Norme internationale

Syntaxe	Nombre de bits	Mnémonique
transport_packet(){		
sync_byte	**8**	**bslbf**
transport_error_indicator	**1**	**bslbf**
payload_unit_start_indicator	**1**	**bslbf**
transport_priority	**1**	**bslbf**
PID	**13**	**uimsbf**
transport_scrambling_control	**2**	**bslbf**
adaptation_field_control	**2**	**bslbf**
continuity_counter	**4**	**uimsbf**
if(adaptation_field_control == '10' || adaptation_field_control == '11'){		
adaptation_field()		
}		
if(adaptation_field_control == '01' || adaptation_field_control == '11') {		
for (i = 0; i < N; i++){		
data_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.3 Définition sémantique des champs dans la couche paquet du flux de transport

sync_byte (octet de synchronisation) – Cet octet est un champ de longueur fixe, codé sur 8 bits, dont la valeur est '0100 0111' (0x47). Il y a lieu d'éviter, lors du choix de valeurs pour d'autres champs apparaissant régulièrement (comme celui de l'identificateur PID), l'émulation de l'octet de synchronisation.

transport_error_indicator (indicateur d'erreur de transport) – Ce fanion est codé sur 1 bit. S'il est mis à '1', il indique qu'il existe au moins 1 erreur binaire non corrigible dans le paquet de transport associé. Ce bit peut être mis à '1' par des entités extérieures à la couche Transport. Il ne sera pas remis à '0' avant que le ou les bits erronés aient été corrigés.

payload_unit_start_indicator (indicateur de début d'unité de capacité utile) – Ce fanion est codé sur 1 bit. Il a une portée normative pour les paquets de transport qui acheminent des paquets de flux PES (voir 2.4.3.6) ou des données d'information PSI (voir 2.4.4).

Lorsque la capacité utile du paquet de flux de transport contient des données de paquet PES, l'indicateur de début d'unité de capacité utile a la signification suivante: un nombre '1' indique que la capacité utile de ce paquet de transport commencera par le premier octet d'un paquet de flux PES et un '0' indique qu'aucun paquet PES ne doit commencer dans ce paquet de transport. Si l'indicateur de début d'unité de capacité utile est mis à '1', un seul paquet PES commence dans un quelconque paquet de transport. Cela s'applique également aux flux privés de type 6 (voir Tableau 2-34).

Lorsque la capacité utile du paquet de flux de transport contient des données d'information PSI, l'indicateur de début de capacité utile a la signification suivante: si le paquet de transport contient le premier octet d'une section d'informations PSI, l'indicateur de début d'unité de capacité utile a la valeur '1', ce qui signifie que le premier octet de la capacité utile de ce paquet de transport contient le champ de pointeur (pointer_field). Si le paquet de transport ne contient pas le premier octet d'une section d'informations PSI, l'indicateur de début d'unité de capacité utile a la valeur '0', ce qui signifie que la capacité utile ne contient aucun champ de pointeur. Voir 2.4.4.1 et 2.4.4.2. Cela s'applique également aux flux privés de type 5 (voir Tableau 2-34).

Pour les paquets vides, l'indicateur de début d'unité de capacité utile doit être mis à '0'.

La présente Spécification ne définit pas la signification de ce bit pour les paquets de transport ne contenant que des données privées.

transport_priority (priorité de transport) – Ce fanion est codé sur 1 bit. S'il est mis à '1', il indique que le paquet associé possède une priorité plus élevée que d'autres paquets ayant le même identificateur PID mais n'ayant pas ce bit mis à '1'. Le mécanisme de transport peut utiliser ce fanion pour donner une priorité à ses données dans un flux élémentaire. Selon l'application, le champ de priorité peut être codé sans référence à l'identificateur PID ou dans le cadre d'un PID donné seulement. Ce champ peut être modifié par des codeurs ou des décodeurs propres au canal.

PID (identificateur PID) – Ce champ est codé sur 13 bits. Il indique le type de données enregistrées dans la capacité utile du paquet. La valeur de PID 0x0000 est réservée à la table d'association de programmes (voir Tableau 2-30). La valeur de PID 0x0001 est réservée à la table d'accès conditionnel (voir Tableau 2-32). La valeur de PID 0x0002 est réservée à la table de description de flux de transport (voir Tableau 2-36). La valeur de PID 0x0003 est réservée à la table d'informations de commande IPMP (voir ISO/CEI 13818-11). Les valeurs de PID 0x0004 à 0x000F sont réservées. La valeur de PID 0x1FFF est réservée aux paquets vides (voir Tableau 2-3).

Tableau 2-3 – Tableau des valeurs d'identificateur PID

Valeur	Description
0x0000	Table d'association de programmes
0x0001	Table d'accès conditionnel
0x0002	Table de description de flux de transport
0x0003	Table d'informations de commande IPMP
0x0004-0x000F	Valeur réservée
0x0010 … 0x1FFE	Valeur pouvant être attribuée comme identificateur PID de réseau, identificateur PID de paquet de mappage de flux de programme, identificateur PID élémentaire ou pour d'autres fins
0x1FFF	Paquet vide
NOTE – Les paquets de transport avec identificateurs PID 0x0000, 0x0001 et 0x0010-0x1FFE peuvent contenir un champ de référence temporelle PCR.	

transport_scrambling_control (commande d'embrouillage du flux de transport) – Ce champ, codé sur 2 éléments binaires, indique le mode d'embrouillage de la capacité utile d'un paquet de transport. L'en-tête d'un tel paquet ne doit pas être embrouillé, ni son champ d'adaptation éventuel. En cas de paquets vides, la valeur de la commande d'embrouillage du flux de transport doit être mise à '00' (voir Tableau 2-4).

Tableau 2-4 – Valeurs de commande d'embrouillage

Valeur	Description
00	Sans embrouillage
01	Défini par l'utilisateur
10	Défini par l'utilisateur
11	Défini par l'utilisateur

adaptation_field_control (commande de champ d'adaptation) – Ce champ, codé sur 2 éléments binaires, indique si l'en-tête d'un paquet de transport est suivi d'un champ et d'une capacité utile (voir Tableau 2-5).

Tableau 2-5 – Valeurs de commande du champ d'adaptation du flux de transport

Valeur	Description
00	Valeur réservée pour futur usage par l'ISO/CEI
01	Pas de champ d'adaptation – capacité utile seulement
10	Champ d'adaptation seulement – pas de capacité utile
11	Champ d'adaptation suivi d'une capacité utile

Les décodeurs conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 doivent ignorer les paquets de transport dont le champ de commande de champ d'adaptation est mis à la valeur '00'. Dans le cas d'un paquet vide, la valeur de la structure adaptation_field_control doit être '01'.

continuity_counter (compteur de continuité) – Codé sur 4 éléments binaires, ce champ progresse à chaque paquet de transport ayant le même identificateur PID. Ce compteur revient automatiquement à zéro lorsqu'il a atteint sa valeur maximale. Il ne doit pas être activé si le champ adaptation_field_control (commande de champ d'adaptation) du paquet a la valeur '00' ou '10'.

Dans les flux de transport, les paquets dupliqués peuvent être envoyés sous la forme de deux et seulement deux paquets de transport consécutifs sous le même identificateur PID. Les paquets dupliqués doivent avoir la même valeur de compteur de continuité que le paquet initial et leur champ adaptation_field_control doit avoir la valeur '01' ou '11'. Dans les paquets dupliqués, chaque octet du paquet initial doit être dupliqué, sauf qu'une valeur valide doit être codée dans les champs de référence temporelle, s'ils sont présents.

Le compteur de continuité contenu dans un certain paquet du flux de transport indique la continuité s'il diffère d'une unité de progression par rapport à la valeur du compteur de continuité contenu dans le précédent paquet de transport du même identificateur PID, ou en cas d'apparition de conditions de non-progression (structure adaptation_field_control mise à '00' ou à '10' ou paquets dupliqués tels que décrits ci-dessus). Le compteur de continuité peut indiquer une discontinuité si la structure discontinuity_indicator est mise à '1' (voir 2.4.3.4). En cas de paquets vides, la valeur du compteur de continuité n'est pas définie.

data_byte (octet de données) – Les octets de données doivent se présenter en données contiguës, issues des paquets PES (voir 2.4.3.6), des sections d'informations PSI (voir 2.4.4) ou d'octets de bourrage de paquets consécutifs à des sections d'informations PSI, ou de données privées non contenues dans ces structures, selon ce qui est indiqué par l'identificateur PID. En cas de paquets vides ayant une valeur de PID égale à 0x1FFF, les structures data_byte peuvent prendre une valeur quelconque. Le nombre N de structures data_byte est spécifié comme étant égal à 184 moins le nombre d'octets contenus dans le champ d'adaptation (), comme indiqué ci-après au 2.4.3.4.

2.4.3.4 Champ d'adaptation

Voir Tableau 2-6.

Tableau 2-6 – Champ d'adaptation du flux de transport

Syntaxe	Nombre de bits	Mnémonique
adaptation_field() {		
adaptation_field_length	**8**	**uimsbf**
if (adaptation_field_length > 0) {		
discontinuity_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
elementary_stream_priority_indicator	**1**	**bslbf**
PCR_flag	**1**	**bslbf**
OPCR_flag	**1**	**bslbf**
splicing_point_flag	**1**	**bslbf**
transport_private_data_flag	**1**	**bslbf**
adaptation_field_extension_flag	**1**	**bslbf**
if (PCR_flag == '1') {		
program_clock_reference_base	**33**	**uimsbf**
reserved	**6**	**bslbf**
program_clock_reference_extension	**9**	**uimsbf**
}		
if (OPCR_flag == '1') {		
original_program_clock_reference_base	**33**	**uimsbf**
reserved	**6**	**bslbf**
original_program_clock_reference_extension	**9**	**uimsbf**
}		
if (splicing_point_flag == '1') {		
splice_countdown	**8**	**tcimsbf**
}		
if (transport_private_data_flag == '1') {		
transport_private_data_length	**8**	**uimsbf**
for (i = 0; i < transport_private_data_length; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
if (adaptation_field_extension_flag == '1') {		
adaptation_field_extension_length	**8**	**uimsbf**
ltw_flag	**1**	**bslbf**
piecewise_rate_flag	**1**	**bslbf**
seamless_splice_flag	**1**	**bslbf**
reserved	**5**	**bslbf**
if (ltw_flag == '1') {		
ltw_valid_flag	**1**	**bslbf**
ltw_offset	**15**	**uimsbf**
}		
if (piecewise_rate_flag == '1') {		
reserved	**2**	**bslbf**
piecewise_rate	**22**	**uimsbf**
}		
if (seamless_splice_flag == '1') {		
splice_type	**4**	**bslbf**
DTS_next_AU[32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
for (i = 0; i < N; i++) {		
reserved	**8**	**bslbf**
}		
}		
for (i = 0; i < N; i++) {		
stuffing_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.5 Définition sémantique des champs contenus dans la structure adaptation_field

adaptation_field_length (longueur du champ d'adaptation) – Ce champ, codé sur 8 éléments binaires, spécifie le nombre d'octets contenus dans la structure adaptation_field qui suit immédiatement la structure adaptation_field_length. La valeur '0' désigne l'insertion d'un unique octet de bourrage dans un octet du flux de transport. Lorsque la valeur de la commande de champ d'adaptation est '11', celle de la longueur du champ d'adaptation doit être comprise entre 0 et 182. Lorsque la valeur de la commande de champ d'adaptation est '10', celle de la longueur du champ d'adaptation doit être de 183. Pour les paquets de transport acheminant des paquets de flux PES, un bourrage est nécessaire s'il n'y a pas assez de données de paquet PES pour remplir complètement les octets de capacité utile dans ces paquets de transport. Le bourrage s'effectue par définition d'un champ d'adaptation plus long que la somme des longueurs des éléments de données contenus dans ce champ, de manière que les octets de capacité utile restant après le champ d'adaptation puissent recevoir exactement les données disponibles des paquets PES. L'espace restant disponible dans le champ d'adaptation est rempli d'octets de bourrage.

Cette méthode de bourrage est la seule qui soit autorisée pour des paquets de transport contenant des paquets PES. Pour les paquets de transport contenant des informations PSI, une autre méthode de bourrage est décrite (voir 2.4.4).

discontinuity_indicator (indicateur de discontinuité) – Ce fanion codé sur 1 bit indique, s'il est mis à '1', que l'état de discontinuité est "Vrai" pour le paquet de transport actuel. Lorsque l'indicateur de discontinuité est mis à '0', ou lorsqu'il n'est pas présent, l'état de discontinuité est "Faux". L'indicateur de discontinuité sert à indiquer deux types de discontinuités: les discontinuités de base de temps du système et celles du compteur de continuité.

Une discontinuité de la base de temps du système est indiquée par l'utilisation de l'indicateur de discontinuité dans les paquets de transport d'un identificateur PID de référence PCR (voir 2.4.4.9). Si l'état de discontinuité est "Vrai" pour un paquet de transport d'un identificateur de référence PCR, la prochaine référence PCR contenue dans un paquet de transport avec le même PID représentera un échantillon de nouvelle base de temps du système pour le programme associé. Le point de discontinuité de la base de temps du système est défini comme l'instant d'arrivée, à l'entrée du décodeur T-STD, du premier octet d'un paquet contenant la référence PCR d'une nouvelle base de temps. L'indicateur de discontinuité doit être mis à '1' dans le paquet contenant la discontinuité de la base de temps. Le bit du fanion indicateur de discontinuité peut également être mis à '1' dans les paquets de transport (ayant le même identificateur PID de référence PCR) qui précèdent le paquet contenant la référence PCR de la nouvelle base de temps. Dans ce cas, une fois que l'indicateur de discontinuité a été mis à '1', il doit continuer à être mis à '1' dans tous les paquets de transport ayant le même identificateur de PCR, jusqu'à et y compris le paquet de transport contenant la première référence PCR de la nouvelle base de temps. Après l'apparition d'une discontinuité, pas plus de deux références PCR relatives à la nouvelle base de temps du système ne doivent être reçues avant qu'une autre discontinuité de la base de temps du système ne puisse se produire. De plus, et sauf lorsque le mode d'enrichissement est à l'état "Vrai", les mémoires tampons T-STD pour un même programme ne doivent pas contenir à un instant donné quelconque de données provenant de plus de deux bases de temps de système.

Avant l'apparition d'une discontinuité de base de temps de système, le premier octet du paquet de transport contenant un horodateur PTS ou DTS qui se rapporte à la nouvelle base de temps ne doit pas arriver à l'entrée du décodeur T-STD. Après l'apparition d'une discontinuité de base de temps de système, le premier octet du paquet de transport contenant un horodateur PTS ou DTS qui se rapporte à la base de temps précédente ne doit pas arriver à l'entrée du décodeur T-STD.

Une discontinuité du compteur de continuité est indiquée par l'utilisation de l'indicateur de discontinuité dans un paquet de transport quelconque. Si l'état de discontinuité est "Vrai" pour un paquet de transport ne contenant pas d'identificateur de référence PCR, la structure continuity_counter qu'il contient peut présenter une discontinuité par rapport au précédent paquet de transport du même identificateur PID. Si l'état de discontinuité est "Vrai" pour un paquet de transport contenant un identificateur PID de référence PCR, la structure continuity_counter qu'il contient ne peut présenter de discontinuité que dans le paquet où apparaît une discontinuité de la base de temps du système. Un point de discontinuité du compteur de continuité apparaît lorsque l'état de discontinuité est "Vrai" dans le paquet de transport et lorsque le compteur de continuité du même paquet présente une discontinuité par rapport au paquet de transport précédent dont l'identificateur PID est identique. Un point de discontinuité du compteur de continuité ne doit pas apparaître plus d'une fois entre le début et la fin de l'état de discontinuité. En outre pour tous les identificateurs PID de référence PCR, lorsque la structure discontinuity_indicator d'un paquet correspondant à un indicateur PID particulier est mise à '1', la structure discontinuity_indicator du paquet de transport suivant correspondant au même PID peut être mise à '1', mais ne peut être mise à '1' dans plus de deux paquets de transport consécutifs de même PID.

Aux fins de ce paragraphe, la définition d'un point d'accès de flux élémentaire est la suivante:

- données vidéo ISO/CEI 11172-2 et Rec. UIT-T H.262 | ISO/CEI 13818-2 – Le premier octet d'un en-tête de séquence vidéo;
- données visuelles ISO/CEI 14496-2 – Le premier octet de l'en-tête de séquence d'objet visuel;
- données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 – Le premier octet d'une unité d'accès AVC. Les ensembles de paramètres SPS et PPS référencés dans cette unité d'accès AVC et dans toutes les unités

d'accès AVC suivantes du flux vidéo codé seront introduits dans le flux d'octets après ce point d'accès et ce, avant leur activation;

- données audio – Le premier octet d'une trame audio.

A la suite d'une discontinuité du compteur de continuité dans un paquet de transport identifié comme contenant des données de flux élémentaire, le premier octet de données de flux élémentaire d'un paquet de flux de transport ayant le même identificateur PID doit être le premier octet d'un point d'accès de flux élémentaire. Dans le cas de données vidéo ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 14496-2, le premier octet d'un point d'accès de flux élémentaire peut aussi être le premier octet d'un code de fin de séquence suivi d'un point d'accès de flux élémentaire.

Chaque paquet de transport contenant des données de flux élémentaire dont l'identificateur PID ne désigne pas une référence PCR, dans lequel se produit une discontinuité du compteur de continuité et contenant un horodateur PTS ou DTS, doit arriver à l'entrée du décodeur T-STD après la discontinuité de base de temps du système pour le programme associé. Lorsque l'état de discontinuité est "Vrai", et si pour deux paquets de transport consécutifs de même identificateur PID et contenant la même valeur du compteur de continuité, les valeurs de la structure adaptation_field_control sont mises à '01' ou '11', alors le deuxième paquet peut être ignoré. Un flux de transport ne doit pas être construit de telle sorte que la mise à l'écart d'un paquet de ce type provoque la perte de données de capacité utile de paquet PES ou de données PSI d'informations spécifiques de programme.

A la suite d'un indicateur de discontinuité mis à la valeur '1' dans un paquet de transport contenant des informations PSI, il ne peut se produire qu'une seule discontinuité du numéro de version des sections PSI. Dès l'apparition d'une discontinuité de ce type, une version des sections du flux de transport relatives au conditionnement du programme approprié doit être envoyée avec un champ section_length mis à la valeur 13 et une valeur de l'indicateur d'applicabilité de la structure suivante égale à '1', de manière qu'il n'y ait pas de descripteur de flux de programme, ni de descripteur de flux élémentaire. Cette version doit être suivie d'une version de la section du flux de transport relative au conditionnement de chaque programme affecté, chaque numéro de version contenant une définition complète de programme étant majoré d'une unité et la valeur actuelle de l'indicateur d'applicabilité de la structure suivante étant mise à '1'. Cette discontinuité indique un changement de version de données d'information PSI.

random_access_indicator (indicateur d'accès aléatoire) – Ce fanion est codé sur 1 bit. Il indique que le paquet de transport actuel et les paquets suivants de même identificateur PID contiennent certaines informations destinées à faciliter l'accès aléatoire en ce point.

En particulier, lorsque ce bit est mis à '1', le paquet PES suivant du début de capacité utile des paquets de transport de même identificateur PID doit contenir un point d'accès de flux élémentaire tel que défini dans la sémantique du champ discontinuity_indicator. De plus, dans le cas de données vidéo, une horodate de présentation doit être présente pour la première image qui suit le point d'accès de flux élémentaire.

En outre, et dans ces différents cas, la présence d'un horodateur de présentation est indispensable dans ce paquet PES ou dans un paquet ultérieur pour la première image suivant l'en-tête de séquence ou pour la trame audio. Lorsque l'identificateur PID contient une référence PCR, le fanion de référence PCR doit être mis à '1'.

elementary_stream_priority_indicator (indicateur de priorité de données de flux élémentaire) – Ce fanion est codé sur 1 bit. Il indique la priorité, entre paquets de même identificateur PID, des données de flux élémentaire contenues dans la capacité utile d'un paquet de transport. Une valeur '1' indique que cette capacité utile a une priorité plus élevée que les capacités utiles d'autres paquets de transport.

Dans le cas de données vidéo ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 14496-2, ce champ ne peut être mis à '1' que si la capacité utile contient un ou plusieurs octets issus d'une tranche à codage intratrame.

Dans le cas de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10, ce champ ne peut être mis à '1' que si la capacité utile contient un ou plusieurs octets issus d'une tranche dont le type est mis à 2, 4, 7 ou 9.

Une valeur '0' indique que la capacité utile a la même priorité que tous les autres paquets qui n'ont pas ce bit mis à '1'.

PCR_flag (fanion de référence PCR) – Ce fanion est codé sur 1 bit. Une valeur '1' indique que le champ d'adaptation contient un champ de référence PCR codé en deux parties. Une valeur '0' indique que le champ d'adaptation ne contient aucun champ de référence PCR.

OPCR_flag (fanion de référence OPCR) – Ce fanion est codé sur 1 bit. Une valeur '1' indique que le champ d'adaptation contient un champ de référence temporelle initiale de programme (OPCR) codé en deux parties. Une valeur '0' indique que le champ d'adaptation ne contient aucun champ de référence OPCR.

splicing_point_flag (fanion de point de combinaison) – Ce fanion est codé sur 1 bit. S'il est mis à '1', il indique qu'un champ de décompte avant combinaison doit être présent dans le champ d'adaptation associé, pour spécifier l'arrivée d'un point de combinaison. Une valeur '0' indique qu'un champ de décompte avant combinaison n'est pas présent dans le champ d'adaptation.

transport_private_data_flag (fanion de données privées dans le flux de transport) – Ce fanion est codé sur 1 bit. S'il est mis à '1', il indique que le champ d'adaptation contient un ou plusieurs octets de données privées. Une valeur '0' indique que le champ d'adaptation ne contient aucun octet de données privées.

adaptation_field_extension_flag (fanion d'extension de champ d'adaptation) – Fanion codé sur 1 bit. Sa valeur '1' indique la présence d'une extension de champ d'adaptation. Une valeur '0' indique qu'une extension de champ d'adaptation n'est pas présente dans le champ d'adaptation.

program_clock_reference_base; program_clock_reference_extension (base de la référence temporelle de programme; extension de la référence temporelle de programme) – Le champ de référence PCR est codé en deux parties sur 42 éléments binaires. La première partie, la base de référence temporelle de programme, constitue un champ de 33 éléments binaires dont la valeur est indiquée par PCR base(i), comme indiqué dans l'équation 2-2. La deuxième partie, dite extension de référence temporelle du programme, est un champ codé sur 9 éléments binaires, dont la valeur est donnée par PCR_ext(i) d'après l'équation 2-3. Le champ de référence PCR donne l'instant prévu d'arrivée de l'octet contenant le dernier élément binaire de la base de référence temporelle du programme à l'entrée du décodeur modèle du système.

original_program_clock_reference_base; original_program_clock_reference_extension (base de la référence temporelle de programme initiale; extension de la référence temporelle de programme initiale) – La référence facultative OPCR est codée sur 42 éléments binaires répartis en deux parties. Ce codage est identique à celui des deux parties correspondantes du champ de référence PCR. La présence de la référence OPCR est indiquée par le fanion de référence OPCR. Le champ de référence OPCR ne doit être codé que dans les paquets de transport contenant le champ de référence PCR. Les références OPCR sont autorisées aussi bien dans les flux de transport monoprogrammes que dans les flux de transport multiprogrammes.

La référence OPCR facilite la reconstruction d'un flux de transport monoprogramme à partir d'un autre flux de transport. Lors de la reconstruction du flux de transport monoprogramme initial, la référence OPCR peut être copiée dans le champ de référence PCR. La valeur de référence PCR résultante n'est valide que si le flux de transport monoprogramme initial est exactement reconstruit dans son intégralité. Ce processus mettra au moins en jeu tous les paquets d'informations PSI et de données privées qui étaient présents dans le flux de transport initial et nécessitera peut-être d'autres dispositions d'ordre privé. Cela implique également que la référence OPCR devra toujours être une copie à l'identique de sa référence PCR associée dans le flux de transport monoprogramme initial.

La référence OPCR est exprimée comme suit:

$$OPCR(i) = OPCR_base(i) \times 300 + OPCR_ext(i) \tag{2-8}$$

où:

$$OPCR_base(i) = ((system_clock_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-9}$$

$$OPCR_ext(i) = ((system_clock_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-10}$$

Le champ de la référence OPCR est ignoré par le décodeur. Il ne doit être modifié par aucun multiplexeur ou décodeur.

splice_countdown (décompte avant combinaison) – Ce champ, codé sur 8 éléments binaires, représente une valeur qui peut être positive ou négative. Une valeur positive spécifie le nombre de paquets de transport restants, sous le même identificateur PID, à la suite du paquet de transport associé, avant qu'un point de combinaison soit atteint. Les paquets de transport dupliqués, ainsi que les paquets de transport ne contenant que des champs d'adaptation, sont exclus. Le point de combinaison apparaît immédiatement après le dernier octet du paquet de transport dont le champ splice_countdown atteint la valeur zéro. Dans le paquet du flux de transport dont le champ splice_countdown atteint la valeur zéro, le dernier octet de données de la capacité utile doit être le dernier octet d'une trame audio codée ou d'une image codée. Dans le cas de données vidéo, l'unité d'accès correspondante n'est pas nécessairement terminée par un code de fin de séquence. Les paquets de transport de même PID qui suivent peuvent contenir des données issues d'un autre flux élémentaire, du même type.

La capacité utile du prochain paquet de transport de même PID (hormis les paquets dupliqués et les paquets sans capacité utile) commence par le premier octet d'un paquet de flux PES. Dans le cas des données audio, la capacité utile de ce paquet PES doit commencer par un point d'accès; dans le cas de données vidéo, la capacité utile du paquet PES doit commencer par un point d'accès ou par un code de fin de séquence suivi d'un point d'accès. La précédente trame audio codée ou image vidéo codée sera donc alignée avec la limite de paquet ou sera "bourrée" pour qu'il en soit ainsi. Au-delà du point de combinaison, le champ de décompte peut également être présent. Lorsque le champ

splice_countdown a une valeur négative de type "moins n" (–n), ce champ signifie que le paquet de transport associé est le n^{e} paquet qui suit le point de combinaison (à l'exclusion des paquets dupliqués et des paquets sans capacité utile).

Pour la définition du point d'accès de flux élémentaire, voir la sémantique de discontinuity_indicator.

transport_private_data_length (longueur de données privées de transport) – Champ codé sur 8 éléments binaires qui spécifie le nombre d'octets de données privées venant immédiatement à la suite du champ de longueur de données privées de transport. Le nombre d'octets de données privées ne doit pas être tel que les données privées aillent au-delà des limites du champ d'adaptation.

private_data_byte (octet de données privées) – Champ codé sur 8 éléments binaires qui ne doit pas être spécifié par l'UIT-T | ISO/CEI.

adaptation_field_extension_length (longueur d'extension du champ d'adaptation) – Champ codé sur 8 éléments binaires qui indique le nombre d'octets des données d'extension du champ d'adaptation ajoutées à la fin de celui-ci, y compris les éventuels octets réservés.

ltw_flag (legal_time_window_flag) (fanion de fenêtre sur temps légal) – Fanion codé sur 1 bit dont la valeur '1' indique la présence du champ de décalage par rapport au temps légal (ltw_offset).

piecewise_rate_flag (fanion de débit discret) – Fanion codé sur 1 bit dont la valeur '1' indique la présence du champ de débit discret.

seamless_splice_flag (fanion de combinaison transparente) – Fanion codé sur 1 bit indiquant, lorsqu'il est mis à '1', la présence de champs réservés au type de combinaison et à l'unité d'accès suivante indiquée par l'horodateur de décodage. La valeur '0' indique que ni le champ réservé au type de combinaison ni le champ réservé à l'unité d'accès suivante indiquée par l'horodateur de décodage ne sont présents. Ce champ ne doit pas prendre la valeur '1' dans des paquets de transport dont le fanion de combinaison transparente n'est pas mis à '1'. Lorsqu'il a été mis à '1' dans un paquet de transport dont le décompte avant combinaison est positif, il doit être mis à '1' dans tous les paquets de transport suivants de même identificateur PID ayant le fanion de point de combinaison mis à '1', jusqu'au paquet dans lequel le décompte avant combinaison s'annule (ce paquet inclus).

Lorsque ce fanion est mis à '1' et que le flux élémentaire transporté sous cet identificateur PID n'est pas un flux vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2, le champ splice_type doit être mis à '0000'; si le flux élémentaire transporté sous cet identificateur PID est un flux vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2, il doit vérifier les contraintes correspondant à la valeur de splice_type.

ltw_valid_flag (legal time window_valid_flag) (fanion de décalage valide par rapport au temps légal) – Fanion codé sur 1 bit dont la valeur '1' prescrit la validité de la valeur du décalage ltw_offset. Une valeur '0' indique que ce décalage est indéfini.

ltw_offset (legal time window_offset) (décalage par rapport à la fenêtre sur temps légal) – Champ codé sur 15 éléments binaires dont la valeur n'est définie que si le fanion ltw_valid_flag a la valeur '1'. Lorsqu'il est défini, ce fanion est exprimé en unités de $(300/f_s)$ secondes, où f_s désigne la fréquence d'horloge du système du programme correspondant à ce PID et vérifie la relation:

$$offset = t_1(i) - t(i)$$

$$ltw_offset = offset // 1$$

où i désigne l'indice du premier octet de ce paquet de transport, offset la valeur de codage de ce champ, t(i) l'instant d'arrivée de l'octet i dans le décodeur T-STD et $t_1(i)$ la limite supérieure dans le temps d'une période de temps appelée la "fenêtre sur temps légal" associée à ce paquet de transport.

La fenêtre sur temps légal se caractérise par la propriété suivante: si ce flux de transport arrive à l'entrée d'un décodeur T-STD à partir de l'instant $t_1(i)$, c'est-à-dire à la fin de sa fenêtre sur temps légal, et si tous les autres paquets de transport du même programme arrivent à la fin de leurs fenêtres sur temps légal, alors:

- dans le cas de données vidéo – La mémoire tampon MB_n pour cet identificateur PID à l'intérieur du décodeur T-STD contiendra moins de 184 octets de données de flux élémentaire à l'instant où y entrera le premier octet de la capacité utile de ce paquet de transport et il ne se produira aucune violation de mémoire tampon à l'intérieur du décodeur T-STD;
- dans le cas de données audio – La mémoire tampon B_n pour cet identificateur PID contiendra moins de $BS_{dec} + 1$ octet de données de flux élémentaire à l'instant où y entrera le premier octet de ce paquet de transport et il ne se produira aucune violation de mémoire tampon à l'intérieur du décodeur T-STD.

En fonction de différents facteurs, notamment la capacité de la mémoire tampon MB_n et la valeur du débit de transfert entre MB_n et EB_n, il est possible de déterminer un autre instant $t_0(i)$ tel que, si ce paquet arrive à un moment quelconque de l'intervalle $[t_0(i), t_1(i)]$, il ne se produira aucune violation de mémoire tampon du décodeur T-STD. Cette période de temps est appelée la fenêtre sur temps légal. La définition de la valeur de t_0 ne figure pas dans la présente Recommandation | Norme internationale.

Les informations codées dans ce champ sont destinées à des dispositifs tels que des remultiplexeurs; elles peuvent leur servir à rétablir l'état antérieur de leur tampon MB_n.

piecewise_rate (débit discret) – La signification de ce champ codé sur 22 éléments binaires n'est définie que si les deux fanions ltw_flag et ltw_valid_flag sont mis à '1'. Le champ en question représente alors un entier positif spécifiant un débit R hypothétique qui est utilisé pour définir les instants finals des fenêtres sur temps légal des paquets de transport de même PID qui suivent ce paquet mais ne contiennent pas le champ de décalage par rapport à la fenêtre sur temps légal.

Supposons que le premier octet de ce paquet de transport et les paquets de transport suivants de même PID correspondent respectivement aux indices A_i, A_{i+1}, ..., A_{i+N}, et que le champ legal_time_window_offset des N derniers paquets ne soit pas codé. Alors les valeurs $t_1(A_{i+j})$ seront déterminées par la relation:

$$t_1(A_{1+j}) = t_1(A_i) + j \times 188 \times 8 \; bits / octet / R$$

où j va de 1 à N.

Tous les paquets compris entre ce paquet et le paquet de transport suivant de même PID contenant un champ legal_time_window_offset doivent être traités comme s'ils contenaient la valeur:

$$offset = t_1(A_i) - t(A_i)$$

correspondant à la valeur $t_1(.)$ calculée d'après la relation ci-dessus, codée dans le champ legal_time_window_offset. A noter que t(j) désigne l'instant d'arrivée de l'octet j dans le décodeur T-STD.

La signification du champ piecewise_rate n'est pas définie lorsqu'il se trouve dans un paquet de transport ne contenant pas de champ legal_time_window_offset.

splice_type (type de combinaison) – Ce champ est codé sur 4 éléments binaires; à compter de sa première occurrence, il doit conserver une valeur inchangée dans tous les paquets de transport suivants de même identificateur PID et à l'intérieur desquels il est présent, et ce jusqu'au paquet de transport contenant un champ splice_countdown dont la valeur s'annule (y compris le paquet en question). Si le flux élémentaire transporté sous cet identificateur PID n'est pas un flux vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2, ce champ doit prendre la valeur '0000'. Si le flux élémentaire transporté sous cet identificateur PID est un flux vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2, ce champ indique les conditions de combinaison que doit vérifier ce flux élémentaire. Les Tableaux 2-7 à 2-20 définissent ces conditions en fonction du profil, du niveau et du type de combinaison.

Dans ces tableaux, l'attribution d'une valeur au "délai de décodage de combinaison", ainsi qu'au "débit maximal du flux après combinaison" impose les conditions suivantes au flux élémentaire vidéo:

1) le dernier octet de l'image codée se terminant dans le paquet de transport à l'intérieur duquel la valeur du champ splice_countdown s'annule, doit rester dans la mémoire tampon du vérificateur VBV pendant une durée égale à (délai de décodage de combinaison $t_{n+1} - t_n$), où aux fins du présent paragraphe:
   - n est l'index de l'image codée dont la fin se trouve dans le paquet de transport contenant le champ splice_countdown dont la valeur s'annule, c'est-à-dire l'image codée mentionnée plus haut;
   - t_n est défini au C.3.1 de la Rec. UIT-T H.262 | ISO/CEI 13818-2;
   - $(t_{n+1} - t_n)$ est défini aux C.9 à C.12 de la Rec. UIT-T H.262 | ISO/CEI 13818-2;

     NOTE – t_n est l'instant auquel l'image codée n est supprimée du tampon VBV et $(t_n + 1 - t_n)$ est la durée pendant laquelle l'image n est présente.
2) la mémoire d'entrée du vérificateur VBV modèle ne doit pas déborder si son entrée est commutée au point de combinaison vers un flux de débit constant égal au "débit maximal du flux après combinaison" pendant une durée égale au "délai de décodage de combinaison".

Tableau 2-7 – Premier tableau de paramètres de combinaison
Profil simple/niveau principal, profil principal/niveau principal,
profil SNR/niveau principal (les deux couches),
profil spatial/niveau supérieur-1440 (dans la couche de base),
profil supérieur/niveau principal (couche centrale + couche de base),
profil multiformat/niveau principal (couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $15{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 150 ms; débit maximal du flux après combinaison = $12{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 225 ms; débit maximal du flux après combinaison = $8{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $7{,}2 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-8 – Deuxième tableau de paramètres de combinaison
Profil principal/niveau inférieur, profil SNR – Niveau inférieur (les deux couches),
profil supérieur/niveau principal (couche de base),
profil multiformat/niveau principal (couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 115 ms; débit maximal du flux après combinaison = $4{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 155 ms; débit maximal du flux après combinaison = $3{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 230 ms; débit maximal du flux après combinaison = $2{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $1{,}8 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-9 – Troisième tableau de paramètres de combinaison
Profil principal/niveau supérieur-1440, profil spatial/niveau supérieur-1440 (toutes les couches),
profil supérieur/niveau supérieur-1440 (couche centrale + couche de base),
profil multiformat/niveau supérieur-1440 (dans la couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $60{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 160 ms; débit maximal du flux après combinaison = $45{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 240 ms; débit maximal du flux après combinaison = $30{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $28{,}5 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-10 – Quatrième tableau de paramètres de combinaison
Profil principal/niveau supérieur, profil supérieur/niveau supérieur-1440 (toutes les couches),
profil supérieur/niveau supérieur (couche centrale + couche de base),
profil multiformat/niveau supérieur (couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $80{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 160 ms; débit maximal du flux après combinaison = $60{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 240 ms; débit maximal du flux après combinaison = $40{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $38{,}0 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-11 – Cinquième tableau de paramètres de combinaison
Profil SNR/niveau inférieur (couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 115 ms; débit maximal du flux après combinaison = $3{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 175 ms; débit maximal du flux après combinaison = $2{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $1{,}4 \times 10^6$ bit/s
0011-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-12 – Sixième tableau de paramètres de combinaison
Profil SNR/niveau principal (couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 115 ms; débit maximal du flux après combinaison = $10{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 145 ms; débit maximal du flux après combinaison = $8{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 235 ms; débit maximal du flux après combinaison = $5{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $4{,}7 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-13 – Septième tableau de paramètres de combinaison
Profil spatial/niveau supérieur-1440
(couche centrale + couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $40{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 160 ms; débit maximal du flux après combinaison = $30{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 240 ms; débit maximal du flux après combinaison = $20{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $19{,}0 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-14 – Huitième tableau de paramètres de combinaison Profil supérieur/niveau principal (toutes les couches), profil supérieur/niveau supérieur-1440 (couche de base) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $20{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 160 ms; débit maximal du flux après combinaison = $15{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 240 ms; débit maximal du flux après combinaison = $10{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $9{,}5 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-15 – Neuvième tableau de paramètres de combinaison Profil supérieur/niveau supérieur (couche de base), profil multiformat/niveau principal (les deux couches) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $25{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 165 ms; débit maximal du flux après combinaison = $18{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $12{,}0 \times 10^6$ bit/s
0011-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-16 – Dixième tableau de paramètres de combinaison Profil supérieur/niveau supérieur (toutes les couches), profil multiformat/niveau superieur-1440 (les deux couches) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $100{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 160 ms; débit maximal du flux après combinaison = $75{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 240 ms; débit maximal du flux après combinaison = $50{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $48{,}0 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-17 – Onzième tableau de paramètres de combinaison Profil 4:2:2/niveau principal – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 45 ms; débit maximal du flux après combinaison = $50{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 90 ms; débit maximal du flux après combinaison = $50{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 180 ms; débit maximal du flux après combinaison = $50{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 225 ms; débit maximal du flux après combinaison = $40{,}0 \times 10^6$ bit/s
0100	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $36{,}0 \times 10^6$ bit/s
0101-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-18 – Douzième tableau de paramètres de combinaison
Profil multiformat/niveau inférieur (les deux couches) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 115 ms; débit maximal du flux après combinaison = $8{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 155 ms; débit maximal du flux après combinaison = $6{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 230 ms; débit maximal du flux après combinaison = $4{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $3{,}7 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-19 – Treizième tableau de paramètres de combinaison
Profil multiformat/niveau supérieur (les deux couches) – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 120 ms; débit maximal du flux après combinaison = $130{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 150 ms; débit maximal du flux après combinaison = $104{,}0 \times 10^6$ bit/s
0010	Délai de décodage de combinaison = 240 ms; débit maximal du flux après combinaison = $65{,}0 \times 10^6$ bit/s
0011	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $62{,}4 \times 10^6$ bit/s
0100-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

Tableau 2-20 – Quatorzième tableau de paramètres de combinaison
Profil 4:2:2/niveau supérieur – Données vidéo

Type de combinaison	Conditions
0000	Délai de décodage de combinaison = 45 ms; débit maximal du flux après combinaison = $300{,}0 \times 10^6$ bit/s
0001	Délai de décodage de combinaison = 90 ms; débit maximal du flux après combinaison = $300{,}0 \times 10^6$ bit/s
0010-0011	Valeur réservée
0100	Délai de décodage de combinaison = 250 ms; débit maximal du flux après combinaison = $180{,}0 \times 10^6$ bit/s
0101-1011	Valeur réservée
1100-1111	Valeur définie par l'utilisateur

DTS_next_AU (unité d'accès suivante indiquée par l'horodatage de décodage) – champ codé sur 33 bits et comportant trois parties. Il indique l'instant de décodage de la première unité d'accès à la suite d'un point de combinaison, dans le cas où aucune combinaison (ou combinaison transparente) n'a été réalisée au point considéré. Cet instant est défini d'après la base de temps valide dans le paquet de transport contenant le champ splice_countdown dont la valeur s'annule. A partir de sa première apparition, ce champ doit conserver la même valeur dans tous les paquets de transport suivants dont le PID est identique à celui du paquet dans lequel il est présent, jusqu'au paquet contenant le champ splice_countdown dont la valeur devient égale à zéro (y compris le paquet en question).

stuffing_byte (octet de remplissage) – Valeur fixe de 8 bits, égale à '1111 1111' pouvant être ingérée par le codeur. Elle est ignorée par le décodeur.

2.4.3.6 Paquet de flux PES

Voir Tableau 2-21.

Tableau 2-21 – Paquet de flux PES

Syntaxe	Nombre de bits	Mnémonique
PES_packet() {		
packet_start_code_prefix	**24**	**bslbf**
stream_id	**8**	**uimsbf**
PES_packet_length	**16**	**uimsbf**
if (stream_id != program_stream_map		
&& stream_id != padding_stream		
&& stream_id != private_stream_2		
&& stream_id != ECM		
&& stream_id != EMM		
&& stream_id != program_stream_directory		
&& stream_id != DSMCC_stream		
&& stream_id != ITU-T Rec. H.222.1 type E stream) {		
'10'	**2**	**bslbf**
PES_scrambling_control	**2**	**bslbf**
PES_priority	**1**	**bslbf**
data_alignment_indicator	**1**	**bslbf**
Copyright	**1**	**bslbf**
original_or_copy	**1**	**bslbf**
PTS_DTS_flags	**2**	**bslbf**
ESCR_flag	**1**	**bslbf**
ES_rate_flag	**1**	**bslbf**
DSM_trick_mode_flag	**1**	**bslbf**
additional_copy_info_flag	**1**	**bslbf**
PES_CRC_flag	**1**	**bslbf**
PES_extension_flag	**1**	**bslbf**
PES_header_data_length	**8**	**uimsbf**
if (PTS_DTS_flags == '10') {		
'0010'	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (PTS_DTS_flags == '11') {		
'0011'	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
'0001'	**4**	**bslbf**
DTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
DTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
DTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (ESCR_flag == '1') {		
reserved	**2**	**bslbf**
ESCR_base[32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
ESCR_base[29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
ESCR_base[14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
ESCR_extension	**9**	**uimsbf**
marker_bit	**1**	**bslbf**
}		
if (ES_rate_flag == '1') {		
marker_bit	**1**	**bslbf**
ES_rate	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
}		

Tableau 2-21 – Paquet de flux PES

Syntaxe	Nombre de bits	Mnémonique
if (DSM_trick_mode_flag == '1') {		
trick_mode_control	**3**	**uimsbf**
if (trick_mode_control == fast_forward) {		
field_id	**2**	**bslbf**
intra_slice_refresh	**1**	**bslbf**
frequency_truncation	**2**	**bslbf**
}		
else if (trick_mode_control == slow_motion) {		
rep_cntrl	**5**	**uimsbf**
}		
else if (trick_mode_control == freeze_frame) {		
field_id	**2**	**uimsbf**
reserved	**3**	**bslbf**
}		
else if (trick_mode_control == fast_reverse) {		
field_id	**2**	**bslbf**
intra_slice_refresh	**1**	**bslbf**
frequency_truncation	**2**	**bslbf**
else if (trick_mode_control == slow_reverse) {		
rep_cntrl	**5**	**uimsbf**
}		
Else		
reserved	**5**	**bslbf**
}		
if (additional_copy_info_flag == '1') {		
marker_bit	**1**	**bslbf**
additional_copy_info	**7**	**bslbf**
}		
if (PES_CRC_flag == '1') {		
previous_PES_packet_CRC	**16**	**bslbf**
}		
if (PES_extension_flag == '1') {		
PES_private_data_flag	**1**	**bslbf**
pack_header_field_flag	**1**	**bslbf**
program_packet_sequence_counter_flag	**1**	**bslbf**
P-STD_buffer_flag	**1**	**bslbf**
reserved	**3**	**bslbf**
PES_extension_flag_2	**1**	**bslbf**
if (PES_private_data_flag == '1') {		
PES_private_data	**128**	**bslbf**
}		
if (pack_header_field_flag == '1') {		
pack_field_length	**8**	**uimsbf**
pack_header()		
}		
if (program_packet_sequence_counter_flag == '1') {		
marker_bit	**1**	**bslbf**
program_packet_sequence_counter	**7**	**uimsbf**
marker_bit	**1**	**bslbf**
MPEG1_MPEG2_identifier	**1**	**bslbf**
original_stuff_length	**6**	**uimsbf**
}		
if (P-STD_buffer_flag == '1') {		
'01'	**2**	**bslbf**
P-STD_buffer_scale	**1**	**bslbf**
P-STD_buffer_size	**13**	**uimsbf**
}		
if (PES_extension_flag_2 == '1') {		
marker_bit	**1**	**bslbf**
PES_extension_field_length	**7**	**uimsbf**
stream_id_extension_flag	**1**	**bslbf**
If (stream_id_extension_flag == '0') {		
stream_id_extension	**7**	**uimsbf**
for (i = 0; i < PES_extension_field_length; i++){		
reserved	**8**	**bslbf**
}		
}		
}		
}		
for (i < 0; i < N1; i++) {		

Tableau 2-21 – Paquet de flux PES

Syntaxe	Nombre de bits	Mnémonique
stuffing_byte	**8**	**bslbf**
}		
for (i < 0; i < N2; i++) {		
PES_packet_data_byte	**8**	**bslbf**
}		
}		
else if (stream_id == program_stream_map		
|| stream_id == private_stream_2		
|| stream_id == ECM		
|| stream_id == EMM		
|| stream_id == program_stream_directory		
|| stream_id == DSMCC_stream		
|| stream_id == ITU-T Rec. H.222.1 type E stream) {		
for (i = 0; i < PES_packet_length; i++) {		
PES_packet_data_byte	**8**	**bslbf**
}		
}		
else if (stream_id == padding_stream) {		
for (i < 0; i < PES_packet_length; i++) {		
padding_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.7 Définition sémantique des champs contenus dans un paquet de flux PES

packet_start_code_prefix (préfixe codé de début de paquet) – Ce champ, codé sur 24 éléments binaires, constitue – de concert avec la structure stream_id qui le suit – un code de début de paquet qui indique le point de départ d'un paquet. Ce préfixe est la chaîne binaire '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id (identificateur de flux) – Dans les flux de programme le champ stream_id spécifie le type et le nombre de flux élémentaires, suivant les définitions indiquées dans le Tableau 2-22. Dans le cas des flux de transport, il est possible d'affecter à l'identificateur stream_id une valeur valide quelconque indiquant correctement le type de flux élémentaire conformément aux définitions du Tableau 2-22. En ce qui concerne les flux de transport, le type de flux élémentaire est spécifié dans les informations spécifiques de programme comme indiqué au 2.4.4.

PES_packet_length (longueur de paquet PES) – Ce champ, codé sur 16 éléments binaires, spécifie le nombre d'octets contenus dans le paquet PES qui suit le dernier octet de ce champ. Une valeur 0 indique que la longueur du paquet PES n'est ni spécifiée ni délimitée. Cette valeur n'est autorisée que dans les paquets PES dont la capacité utile est un flux élémentaire de données vidéo contenu dans des paquets de flux de transport.

PES_scrambling_control (commande d'embrouillage de flux PES) – Ce champ, codé sur 2 éléments binaires, indique le mode d'embrouillage de la capacité utile d'un paquet de flux PES. L'en-tête de paquet PES ne doit pas être embrouillé, non plus que les (éventuels) champs facultatifs (voir Tableau 2-23).

Tableau 2-22 – Affectations de l'identificateur stream_id

Identificateur de flux	Note	Codage du flux
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		numéro de flux audio x xxxx ISO/CEI 13818-3 ou ISO/CEI 11172-3 ou ISO/CEI 13818-7 ou ISO/CEI 14496-3
1110 xxxx		numéro de flux vidéo xxxx de la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 11172-2 ou ISO/CEI 14496-2 ou Rec. UIT-T H.264 | ISO/CEI 14496-10
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	Annexe A de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 ou ISO/CEI 13818-6_DSMCC_stream
1111 0011	2	ISO/CEI_13522_stream
1111 0100	6	Rec. UIT-T H.222.1 type A
1111 0101	6	Rec. UIT-T H.222.1 type B
1111 0110	6	Rec. UIT-T H.222.1 type C
1111 0111	6	Rec. UIT-T H.222.1 type D
1111 1000	6	Rec. UIT-T H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/CEI 14496-1_SL-packetized_stream
1111 1011		ISO/CEI 14496-1_FlexMux_stream
1111 1100		flux de métadonnées
1111 1101	8	extended_stream_id
1111 1110		flux de données réservé
1111 1111	4	program_stream_directory

La notation x signifie que la valeur '0' ou la valeur '1' est autorisée et aboutit au même type de flux. Le numéro de flux est donné par les valeurs prises par les x.

NOTE 1 – Les paquets PES de type program_stream_map ont une syntaxe unique, qui est spécifiée au 2.5.4.1.

NOTE 2 – Les paquets de type private_stream_1 et ISO/CEI_13552_stream suivent la même syntaxe de paquet PES que pour les flux vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2 et que pour les flux audio ISO/CEI 13818-3.

NOTE 3 – Les paquets PES de type private_stream_2, ECM_stream et EMM_stream sont analogues au champ private_stream_1 sauf qu'aucune syntaxe n'est spécifiée après le champ PES_packet_length.

NOTE 4 – Les paquets PES de type program_stream_directory ont une syntaxe unique, spécifiée au 2.5.5.

NOTE 5 – Les paquets PES de type DSM-CC_stream ont une syntaxe unique, qui est spécifiée dans l'ISO/CEI 13818-6.

NOTE 6 – Cet identificateur stream_id est associé au type de flux 0x09 indiqué au Tableau 2-34.

NOTE 7 – Cet identificateur stream_id est utilisé uniquement dans les paquets PES, qui transportent des données provenant d'un flux de programme ou d'un flux de système ISO/CEI 11172-1, dans un flux de transport (voir 2.4.3.8).

NOTE 8 – L'utilisation de l'identificateur stream_id 0xFD (extended_stream_id) indique que le paquet PES emploie une syntaxe étendue pour permettre d'identifier d'autres types de flux.

Tableau 2-23 – Valeurs de commande de l'embrouillage du flux PES

Valeur	Description
00	Sans embrouillage
01	Défini par l'utilisateur
10	Défini par l'utilisateur
11	Défini par l'utilisateur

PES_priority (priorité au flux PES) – Ce champ, codé sur 1 bit, indique la priorité de la capacité utile contenue dans ce paquet PES. Une valeur '1' indique une priorité de capacité utile PES plus élevée que celle d'un paquet PES dont ce fanion est à '0'. Un multiplexeur peut utiliser le bit de priorité au flux PES pour avantager ses propres données dans un flux élémentaire. Ce fanion ne doit pas être modifié par le mécanisme de transport.

data_alignment_indicator (indicateur d'alignement de données) – Indicateur logique codé sur un bit. La valeur '1' indique que l'en-tête de paquet PES est immédiatement suivi de l'élément de syntaxe vidéo ou du mot de synchronisation audio indiqué dans le descripteur d'alignement de flux de données défini au 2.6.10, si ce descripteur est présent. Si la valeur est à '1' et que le descripteur est absent, il faut utiliser l'alignement de type '01' tel que défini dans les Tableaux 2-53, 2-54 ou 2-55. La valeur '0' indique que la question de savoir si un tel alignement est exécuté ou non n'est pas définie.

copyright (droit d'auteur) – Si ce champ codé sur 1 bit est mis à '1', il indique que les données contenues dans la capacité utile du paquet PES associé sont protégées par un droit d'auteur. Si ce fanion est mis à '0', la protection par droit d'auteur est indéterminée. Lorsqu'un descripteur de droit d'auteur (voir 2.6.24) est associé au flux élémentaire contenant ce paquet PES et que le fanion de droit d'auteur est mis à '1', le descripteur s'applique aux données contenues dans ce paquet PES.

original_or_copy (original ou copie) – Si ce champ est mis à '1', le contenu de capacité utile du paquet PES associé est un original. S'il est mis à '0', il s'agit d'une copie.

PTS_DTS_flags (fanions PTS et DTS) – Champ codé sur 2 éléments binaires. Si sa valeur est '10', les champs PTS sont présents dans l'en-tête de paquet PES. Si sa valeur est '11', l'en-tête de paquet PES contient à la fois des champs PTS et DTS. Si sa valeur est '00', l'en-tête de paquet PES ne contient ni champ PTS ni champ DTS. La valeur '01' est interdite.

ESCR_flag (fanion de référence ESCR) – Si ce fanion codé sur 1 bit est mis à '1', les champs de base et d'extension de référence ESCR sont présents dans l'en-tête de paquet PES. S'il a la valeur '0', aucun champ de référence ESCR n'est présent.

ES_rate_flag (fanion de débit de flux élémentaire) – Si ce fanion codé sur 1 bit est mis à '1', le champ ES_rate est présent dans l'en-tête de paquet PES. S'il a la valeur '0', aucun champ de débit de flux élémentaire n'est présent.

DSM_trick_mode_flag (fanion de mode d'enrichissement de support DSM) – Si ce fanion codé sur 1 bit est mis à '1', le champ de mode d'enrichissement, codé sur 8 éléments binaires, est présent. S'il a la valeur '0', ce champ n'est pas présent.

additional_copy_info_flag (fanion d'information supplémentaire sur le droit d'auteur) – Si ce fanion codé sur 1 bit est mis à '1', le champ d'information supplémentaire sur le droit d'auteur est présent. S'il a la valeur '0', ce champ n'est pas présent.

PES_CRC_flag (fanion de contrôle CRC du flux PES) – Si ce fanion codé sur 1 bit est mis à '1', un champ de contrôle CRC est présent dans le paquet PES. S'il a la valeur '0', ce champ n'est pas présent.

PES_extension_flag (fanion d'extension de paquet PES) – Si ce fanion codé sur 1 bit est mis à '1', un champ d'extension est présent dans l'en-tête de ce paquet PES. S'il a la valeur '0', ce champ n'est pas présent.

PES_header_data_length (longueur des données d'en-tête de paquet PES) – Champ codé sur 8 éléments binaires qui spécifie le nombre total d'octets occupés par les champs facultatifs et par d'éventuels octets de bourrage contenus dans cet en-tête de paquet PES. La présence de champs facultatifs est indiquée par l'octet qui précède le champ de longueur des données d'en-tête de paquet PES.

marker_bit (bit marqueur) – Champ codé sur 1 bit qui a la valeur '1'.

PTS (presentation time stamp) (horodateur de présentation) – Les instants de présentation doivent être liés de la façon suivante aux instants de décodage. L'horodateur PTS est un nombre de 33 éléments binaires codés en trois champs séparés. Il indique l'instant $tp_n(k)$ de présentation, dans le décodeur modèle du système, d'une unité de présentation k du flux élémentaire n. La valeur du champ PTS est exprimée en nombre de périodes de l'horloge système divisé par 300 (de façon à obtenir une fréquence de 90 kHz). L'instant de décodage est calculé d'après l'horodateur PTS au moyen de l'équation 2-11 ci-dessous. Voir 2.7.4 pour les contraintes applicables à la fréquence de codage des horodateurs de présentation.

$$PTS(k) = ((system_clock_frequency \times tp_n(k)) \; DIV \; 300) \; \% \; 2^{33} \qquad (2\text{-}11)$$

où $tp_n(k)$ est l'instant de présentation de l'unité de présentation $P_n(k)$.

Dans le cas de données audio, si l'en-tête de paquet PES contient un horodateur PTS, celui-ci doit se rapporter à la première unité d'accès qui commence dans ledit paquet. Une unité d'accès audio commence dans un paquet PES si le premier octet de cette unité est contenu dans les données du paquet.

Si, dans le cas de données vidéo ISO/CEI 11172-2 ou ISO/CEI 14496-2, l'en-tête de paquet PES contient une horodate PTS, celle-ci doit se rapporter à l'unité d'accès contenant le premier code de déclenchement d'image qui commence dans le paquet PES. Un code de déclenchement d'image commence dans un paquet PES si le premier octet de ce code est présent dans le paquet PES. Pour les images à codage de type I ou P contenues dans des séquences autres qu'à faible délai et s'il n'y a pas de discontinuité de décodage entre les unités d'accès (AU, *access unit*) k et k′, l'instant de présentation $t_{pn}(k)$ doit être égal à l'instant de décodage $t_{dn}(k')$ de la prochaine image I ou P transmise (voir 2.7.5). S'il y a une discontinuité de décodage, ou si le flux se termine, la différence entre les instants $t_{pn}(k)$ et $t_{dn}(k)$ doit être la même que si le flux initial avait continué sans discontinuité, ni terminaison.

NOTE 1 – Une séquence à faible délai est une séquence vidéo ISO/CEI 14496-2 dans laquelle le fanion low_delay est mis à '1' (voir 6.2.3 de l'ISO/CEI 14496-2).

Si, dans le cas de données vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2, l'en-tête de paquet PES contient une horodate PTS, celle-ci doit se rapporter à l'unité d'accès contenant le premier code de déclenchement d'image qui commence dans le paquet PES. Un code de déclenchement d'image commence dans un paquet PES si le premier octet de ce code est présent dans le paquet PES. Pour les images bi-trames à codage de type I ou P contenues dans des séquences autres qu'à faible délai et s'il n'y a pas de discontinuité de décodage entre les unités d'accès (AU, *access unit*) k et k′, l'instant de présentation $t_{pn}(k)$ doit être égal à l'instant de décodage $t_{dn}(k')$ de la prochaine image bi-trame à codage de type I ou P transmise (voir 2.7.5). S'il y a une discontinuité de décodage, ou si le flux se termine, la différence entre les instants $t_{pn}(k)$ et $t_{dn}(k)$ doit être la même que si le flux initial avait continué sans discontinuité, ni terminaison.

NOTE 2 – Une séquence à faible délai est une séquence vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2 dans laquelle le fanion low_delay est mis à '1' (voir 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2). Par ailleurs, il est à noter que pour les images monotrames l'heure de présentation se rapporte à la première image monotrame de l'image bi-trame codée.

Si, dans le cas de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10, l'en-tête de paquet PES contient une horodate PTS, celle-ci doit se rapporter à la première unité d'accès AVC qui commence dans ce paquet PES. Une unité d'accès AVC commence dans un paquet PES si le premier octet de cette unité est présent dans le paquet PES. Dans un souci de cohérence entre le décodeur STD et le décodeur HRD défini dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10, pour chaque unité d'accès AVC décodée, la valeur de PTS donnée par le décodeur STD doit, dans les limites de la précision des horloges respectives, indiquer le même instant que l'instant de sortie nominal du tampon DPB donné par le décodeur HRD, défini ici comme étant $t_{o,n,dpb}(n) = t_{r,n}(n) + t_c$ * dpb_output_delay(n), où $t_{r,n}(n)$, t_c, et dpb_output_delay(n) sont définis dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

NOTE 3 – On peut utiliser des horloges différentes pour déterminer l'horodate PTS et $t_{o,n,dpb}(n)$.

L'instant de présentation $t_{pn}(k)$ sera identique à l'instant de décodage $t_{dn}(k)$ pour:

- les unités d'accès audio;
- les unités d'accès des séquences vidéo à faible délai Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 14496-2;
- les images B des flux vidéo ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 14496-2.

S'il y a filtrage des données audio, le modèle du système part du principe que ce filtrage n'introduit aucun retard, de sorte que l'échantillon indiqué par un horodateur PTS lors du codage est le même échantillon que celui qui est indiqué par cet horodateur PTS lors du décodage. En cas de codage échelonnable, voir 2.7.6.

DTS (decoding time stamp) (horodateur de décodage) – L'horodateur DTS est un nombre de 33 éléments binaires codés en trois champs séparés. Il indique l'instant de décodage $td_n(j)$, dans le décodeur modèle du système, d'une unité d'accès j du flux élémentaire n. La valeur du champ DTS est exprimée en nombre de périodes de l'horloge système divisé par 300 (de façon à obtenir une fréquence de 90 kHz). L'instant de décodage est calculé d'après l'horodateur DTS au moyen de la relation 2-12 ci-dessous:

$$DTS(j) = ((system_clock_frequency \times td_n(j)) \; DIV \; 300) \; \% \; 2^{33} \qquad (2\text{-}12)$$

où $td_n(j)$ est l'instant de décodage de l'unité d'accès $A_n(j)$.

Si, dans le cas de données vidéo ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 14496-2, l'en-tête de paquet PES contient une horodate DTS, celle-ci doit se rapporter à l'unité d'accès contenant le premier code de déclenchement d'image qui commence dans ce paquet PES. Un code de déclenchement d'image commence dans un paquet PES si le premier octet de ce code est présent dans le paquet.

Si, dans le cas de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10, l'en-tête de paquet PES contient une horodate DTS, celle-ci doit se rapporter à la première unité d'accès AVC qui commence dans ce paquet PES. Une unité d'accès AVC commence dans un paquet PES si le premier octet de cette unité est présent dans le paquet PES. Dans un souci de cohérence entre le décodeur STD et le décodeur HRD défini dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10, pour chaque unité d'accès AVC, la valeur de DTS donnée par le décodeur STD doit, dans les limites de la précision des horloges respectives, indiquer le même instant que l'instant de sortie nominal du tampon CPB $t_{r,n}(n)$ donné par le décodeur HRD, défini dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

NOTE 4 – On peut utiliser des horloges différentes pour déterminer l'horodate DTS et $t_{r,n}(n)$.

Dans le cas d'un codage échelonnable, voir 2.7.6.

ESCR_base; ESCR_extension – La référence temporelle du flux élémentaire ESCR est un champ codé sur 42 éléments binaires comprenant deux parties: la première (ESCR_base) est codée sur 33 éléments et sa valeur ESCR_base(i) est fournie par la relation 2-14; la seconde (ESCR_ext) est un champ codé sur 9 éléments, dont la valeur ESCR_ext(i) est donnée par la relation 2-15. Le champ de référence temporelle du flux élémentaire indique l'instant d'arrivée prévu de l'octet contenant le dernier bit du champ de base ESCR_base à l'entrée du décodeur PES-STD pour flux PES (voir 2.5.2.4).

Plus précisément:

$$ESCR(i) = ESCR_base(i) \times 300 + ESCR_ext(i) \tag{2-13}$$

où:

$$\text{ESCR_base(i)} = ((\text{system_clock_ frequency} \times \text{t(i)}) \text{ DIV } 300) \% 2^{33} \tag{2-14}$$

$$ESCR_ext(i) = ((system_clock_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-15}$$

Le champ ESCR de référence temporelle de flux élémentaire et le champ ES_rate de débit de flux élémentaire (voir les définitions ci-après) contiennent des données de synchronisation concernant la séquence de flux PES. Ils doivent satisfaire aux contraintes définies au 2.7.3.

ES_rate (elementary stream rate) (débit de flux élémentaire) – Ce champ est un entier arithmétique spécifiant le débit auquel le décodeur modèle du système reçoit des octets en paquets PES, dans le cas d'un flux PES. Le débit de flux élémentaire est valide dans le paquet PES qui le contient et dans les paquets PES suivants jusqu'à l'apparition d'un nouveau champ ES_rate. La valeur de ce champ est mesurée en unités de 50 octets par seconde. La valeur 0 est interdite; elle sert à définir l'instant d'arrivée des octets à l'entrée d'un décodeur P-STD pour des flux PES tels que définis au 2.5.2.4. La valeur codée dans le champ ES_rate peut varier de paquet PES en paquet PES.

trick_mode_control (commande de mode d'enrichissement) – Ce champ, codé sur 3 éléments binaires, indique le mode d'enrichissement qui est appliqué au flux vidéo associé. Lorsqu'il s'agit d'autres types de flux élémentaire, ce champ ainsi que les cinq éléments binaires suivants n'ont pas de significations définies. La définition de l'état trick_mode figure au 2.4.2.3 concernant le **mode d'enrichissement**.

Si le mode d'enrichissement prend la valeur "Faux", le nombre de fois N qu'une image est présentée à la sortie du processus de décodage applicable à des séquences d'images à balayage progressif, est spécifié pour chaque image, d'une part, par la valeur des champs repeat_first_field et top_field qui sont les premiers dans le cas de flux vidéo selon la Rec. UIT-T H.262 | ISO/CEI 13818-2 et, d'autre part, par l'en-tête de séquence pour les flux vidéo selon ISO/CEI 11172-2.

Pour les séquences d'images à balayage entrelacé lorsque le mode d'enrichissement prend la valeur "Faux", le nombre de fois N qu'une image est présentée à la sortie du processus de décodage applicable à des séquences d'images à balayage, est spécifié pour chaque image par la valeur des champs repeat_first_field et progressive_frame, dans le cas de flux vidéo selon la Rec. UIT-T H.262 | ISO/CEI 13818-2.

Si le mode d'enrichissement prend la valeur "Vrai", le nombre de fois qu'une image doit être affichée dépend de N.

On notera que si la valeur de ce champ change ou en cas de désactivation du mode d'enrichissement, une combinaison quelconque des phénomènes ci-dessous est susceptible de se produire:

- discontinuité de la base de temps;
- discontinuité de décodage;
- discontinuité du compteur de continuité.

Tableau 2-24 – Valeurs de commande du mode d'enrichissement

Valeur	Description
'000'	Avance rapide
'001'	Ralenti
'010'	Image figée
'011'	Retour rapide
'100'	Retour lent
'101'-'111'	Valeur réservée

Dans le contexte du mode d'enrichissement, une anomalie de vitesse de décodage et de présentation peut provoquer une erreur dans les valeurs de certains champs définis dans les données vidéo de flux élémentaire. De même, la validité de la contrainte sémantique sur la structure en tranches peut être affectée. Les éléments syntaxiques pour données vidéo auxquels cette exception est applicable sont les suivants:

- bit_rate;
- vbv_delay;
- repeat_first_field;
- v_axis_positive;
- field_sequence;
- subcarrier;
- burst_amplitude;
- subcarrier_phase.

Un décodeur ne peut pas se fonder sur les valeurs codées dans les champs ci-après lorsque le mode d'enrichissement est activé.

Le décodage du champ trick_mode_control ne constitue pas une exigence normative applicable aux décodeurs. Toutefois les exigences normatives suivantes sont applicables aux décodeurs qui assurent effectivement le décodage de ce champ.

fast forward (avance rapide) – La valeur '000' dans le champ trick_mode_control indique un flux vidéo en avance rapide et définit la signification des cinq éléments binaires suivants de l'en-tête de paquet PES. Le fanion intra_slice_refresh peut être mis à '1' pour indiquer que certains macroblocs peuvent manquer et que le décodeur peut les remplacer par des macroblocs ayant les mêmes coordonnées spatiales dans des images déjà décodées. Le champ field_id, défini dans le Tableau 2-25, indique le ou les champs qu'il y a lieu d'afficher. Le champ frequency_truncation indique qu'un ensemble limité de coefficients peut être présent. Le Tableau 2-26 donne la signification des valeurs de ce champ.

slow motion (ralenti) – La valeur '001' dans le champ trick_mode_control indique un flux de données vidéo en ralenti et définit la signification des cinq éléments binaires suivants de l'en-tête de paquet PES. Dans le cas de séquences à balayage progressif, l'image doit être affichée N × rep_cntrl fois, avec N défini plus haut.

Lorsqu'il s'agit de séquences vidéo ISO/CEI 11172-2 et de séquences vidéo à balayage progressif Rec. UIT-T H.262 | ISO/CEI 13818-2, le temps d'affichage de l'image doit être de N × rep_cntrl picture.

Dans le cas de séquences vidéo à balayage entrelacé selon la Rec. UIT-T H.262 | ISO/CEI 13818-2, le temps d'affichage de l'image doit être de N × rep_cntrl field. Si l'image est une image tramée, la première trame à afficher est soit la trame supérieure si le champ top_field_first est d'abord égal à 1, soit la trame inférieure si le champ top_field_first est d'abord égal à '0' (voir la Rec. UIT-T H.262 | ISO/CEI 13818-2). Cette trame est affichée pendant un temps égal à N × rep_cntrl / 2 field. L'autre trame de l'image est ensuite affichée pendant un temps égal à N – N × rep_cntrl / 2 field.

freeze frame (arrêt sur image) – La valeur '010' dans le champ trick_mode_control indique un flux de données vidéo en arrêt sur image et définit la signification des cinq éléments binaires suivants de l'en-tête de paquet PES. Le champ field_id est défini dans le Tableau 2-25 – Valeurs de commande du champ d'identification field_id. Le champ field_id se rapporte à la première unité d'accès vidéo qui commence dans le paquet PES qui contient le champ field_id, à moins que le paquet PES ne contienne aucun octet de capacité utile; dans ce cas le champ field_id se rapporte à la précédente unité d'accès de données vidéo la plus récente.

fast reverse (retour rapide) – La valeur '011' dans le champ trick_mode_control indique un flux de données vidéo en retour rapide et définit la signification des cinq éléments binaires suivants de l'en-tête de paquet PES. Le bit du fanion intra_slice_refresh peut être mis à '1' pour indiquer que certains macroblocs peuvent manquer et que le décodeur peut

les remplacer par des macroblocs ayant les mêmes coordonnées spatiales dans des images déjà décodées. Le champ field_id, défini dans le Tableau 2-25, indique le ou les champs qu'il y a lieu d'afficher. Le champ frequency_truncation indique qu'un ensemble limité de coefficients peut être présent. La signification des valeurs de ce champ est donnée dans le Tableau 2-26.

slow reverse (retour lent) – La valeur '100' dans le champ trick_mode_control indique un flux de données vidéo en retour lent et définit la signification des cinq éléments binaires suivants de l'en-tête de paquet PES. Lorsqu'il s'agit de séquences vidéo selon ISO/CEI 11172-2 et de séquences vidéo à balayage progressif Rec. UIT-T H.262 | ISO/CEI 13818-2, le temps d'affichage de l'image doit être de N × rep_cntrl picture, avec N défini plus haut.

Dans le cas de séquences vidéo à balayage entrelacé selon la Rec. UIT-T H.262 | ISO/CEI 13818-2, le temps d'affichage de l'image doit être de N × rep_cntrl field. Si l'image est une image tramée, la première trame à afficher est soit la trame inférieure si le champ top_field_first est d'abord égal à 1, soit la trame supérieure si le champ top_field_first est d'abord égal à '0' (voir Rec. UIT-T H.262 | ISO/CEI 13818-2). Cette trame est affichée pendant un temps égal à N × rep_cntrl / 2 field. L'autre trame de l'image est ensuite affichée pendant un temps égal à N – N × rep_cntrl / 2 field.

field_id (identificateur de trame) – Ce champ, codé sur 2 éléments binaires, indique quelle(s) trame(s) il faut afficher conformément au Tableau 2-25.

Tableau 2-25 – Valeurs de commande du champ d'identification field_id

Valeur	Description
'00'	Affichage à partir de la trame supérieure seulement
'01'	Affichage à partir de la trame inférieure seulement
'10'	Affichage de l'image bitrame complète
'11'	Valeur réservée

intra_slice_refresh (remplacement de macroblocs manquants dans une tranche) – Le bit de ce fanion, s'il est mis à '1', indique que certains macroblocs peuvent manquer entre des tranches codées de données vidéo de ce paquet PES. S'il est mis à '0', cela ne peut pas se produire. Pour plus de détails, voir Rec. UIT-T H.262 | ISO/CEI 13818-2. Le décodeur peut remplacer les macroblocs manquants par des macroblocs de mêmes coordonnées spatiales dans des images déjà décodées.

frequency_truncation (limitation de la série de coefficients fréquentiels) – Ce champ, codé sur 2 éléments binaires, indique qu'un ensemble limité de coefficients a peut-être été utilisé pour coder les données vidéo dans ce paquet PES. Ses valeurs sont définies dans le Tableau 2-26.

Tableau 2-26 – Valeurs de sélection des coefficients

Valeur	Description
'00'	Seuls les coefficients DC sont différents de zéro
'01'	Seuls les trois premiers coefficients sont différents de zéro
'10'	Seuls les six premiers coefficients sont différents de zéro
'11'	Tous les coefficients peuvent être différents de zéro

rep_cntrl (commande de répétition de trame) – Codé sur 5 éléments binaires, ce champ indique le nombre de fois qu'il y a lieu d'afficher chaque trame d'une image à balayage entrelacé, ou d'afficher une image à balayage progressif. Dans le cas d'images à balayage entrelacé, c'est la valeur du champ trick_mode_control, ainsi que le premier octet du champ top_field_first de l'en-tête de séquence vidéo, qui indique s'il y a lieu d'afficher d'abord la trame supérieure ou la trame inférieure. La valeur '0' est interdite.

additional_copy_info (information supplémentaire sur le droit d'auteur) – Ce champ, codé sur 7 éléments binaires, contient des données privées se rapportant aux informations de droit d'auteur.

previous_PES_packet_CRC (contrôle CRC du paquet PES précédent) – Ce champ, codé sur 16 éléments binaires, contient la valeur de contrôle CRC qui se traduit par une sortie zéro des 16 registres d'un décodeur semblable à celui qui est défini dans l'Annexe A mais avec le polynôme:

$$x^{16} + x^{12} + x^5 + 1$$

après traitement des octets de données du précédent paquet PES, à l'exclusion de l'en-tête de paquet PES.

NOTE 5 – Ce contrôle CRC est destiné à la maintenance des réseaux, comme la localisation de la source d'erreurs intermittentes. Il n'est pas destiné à être utilisé par des décodeurs de flux élémentaire. Il n'est calculé que sur les octets de données car les données d'un en-tête de paquet PES peuvent être modifiées en cours de transport.

PES_private_data_flag (fanion de données privées en paquets PES) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que l'en-tête de paquet PES contient des données privées. Lorsqu'il est mis à une valeur de '0', il indique que des données privées ne sont pas présentes dans l'en-tête de paquet PES.

pack_header_field_flag (fanion de présence d'un en-tête de bloc de paquets) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que l'en-tête de paquet PES contient un en-tête de bloc de paquets conforme à ISO/CEI 11172-1 ou un en-tête de bloc de paquets pour flux de programme. Si ce fanion est inséré dans un paquet PES lui-même contenu dans un flux de programme, il doit être mis à '0'. S'il est mis à '0' dans un flux de transport, ce fanion indique qu'aucun en-tête de bloc de paquets n'est présent dans l'en-tête de paquet PES.

program_packet_sequence_counter_flag (fanion de compteur de paquets en séquence dans un flux de programme) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que les champs program_packet_sequence_counter, identificateurs MPEG1 et MPEG2, et original_stuff_length sont présents dans le paquet PES. Lorsqu'il est mis à '0', ce fanion indique que ces champs ne sont pas présents dans l'en-tête de paquet PES.

P-STD_buffer_flag (fanion de mémoire du décodeur P-STD) – Lorsqu'il est mis à '1', ce fanion indique que les champs P-STD_buffer_scale et P-STD_buffer_size sont présents dans l'en-tête de paquet PES. S'il est mis à '0', ce fanion indique que ces champs ne sont pas présents dans l'en-tête de paquet PES.

PES_extension_flag_2 (2^e fanion d'extension de paquet PES) – Lorsqu'il est mis à '1', ce fanion indique la présence du champ PES_extension_field_length et des champs associés. S'il est mis à '0', ce fanion signale l'absence du champ PES_extension_field_length et de tous les champs associés.

PES_private_data (données privées de flux PES) – Ce champ, codé sur 16 octets, contient des données privées. Celles-ci, en combinaison avec les champs qui les précèdent et qui les suivent, ne doivent pas émuler le champ packet_start_code_prefix (0x000001).

pack_field_length (longueur du champ d'en-tête de bloc de paquets) – Codé sur 8 éléments binaires, ce champ indique la longueur, en octets, du champ pack_header_field().

program_packet_sequence_counter (compteur de paquets en séquence dans un flux de programme) – Ce champ est codé sur 7 éléments binaires. Il désigne un compteur facultatif qui progresse à chaque paquet PES successif issu d'un flux de programme ou d'un flux de données conformes à ISO/CEI 11172-1 ou avec chaque paquet PES associé à une définition de programme unique dans un flux de transport. Sa fonction est assimilable à celle d'un compteur de continuité (voir 2.4.3.2). Il permet à une application de retrouver la séquence originale de paquets PES d'un flux de programme ou la séquence originale de paquets du flux de données original selon ISO/CEI 11172-1. Le compteur reviendra de lui-même à zéro une fois sa valeur maximale atteinte. La répétition d'un paquet PES ne doit pas se produire. Il en découle que deux paquets PES consécutifs dans le multiplex de programmes ne peuvent pas avoir la même valeur de champ program_packet_sequence_counter.

MPEG1_MPEG2_identifier (identificateur de format MPEG1 ou MPEG2) – S'il est mis à '1', ce fanion codé sur 1 bit indique que le paquet PES contient des informations issues d'un flux de données conforme à ISO/CEI 11172-1. S'il est mis à la valeur '0', ce fanion indique que le paquet PES contient des informations issues d'un flux de programme.

original_stuff_length (longueur du bourrage original) – Codé sur 6 éléments binaires, ce champ spécifie le nombre d'octets de bourrage utilisés dans l'en-tête de paquet PES original, conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 ou dans l'en-tête de paquet PES original conforme à ISO/CEI 11172-1.

P-STD_buffer_scale (échelle de capacité mémoire du décodeur P-STD) – Ce fanion est un champ codé sur un bit, dont la signification n'est définie que si le paquet PES est contenu dans un flux de programme. Il indique le facteur de normalisation utilisé pour interpréter le champ P-STD_buffer_size suivant. Si le précédent identificateur de flux a indiqué un flux de données audio, le champ P-STD_buffer_scale doit avoir la valeur '0'. Si le précédent identificateur de flux a indiqué un flux de données vidéo, le champ P-STD_buffer_scale doit avoir la valeur '1'. Pour tous les autres types de flux, sa valeur peut être '1' ou '0'.

P-STD_buffer_size (capacité mémoire du décodeur P-STD) – Ce champ est codé sur 13 éléments binaires par un entier arithmétique dont la signification n'est définie que si le paquet PES est contenu dans un flux de programme; il définit la capacité de la mémoire tampon d'entrée dans le décodeur P-STD, BS_n. Si ce champ a la valeur '0', la capacité mémoire est exprimée en unités de 128 octets. S'il a la valeur '1', la capacité est exprimée en unités de 1024 octets. On a donc:

$$\text{si } (P\text{-}STD_buffer_scale == 0)$$
$$BS_n = P\text{-}STD_buffer_size \times 128 \quad (2\text{-}16)$$

sinon:

$$BS_n = P\text{-}STD_buffer_size \times 1024 \quad (2\text{-}17)$$

La valeur codée du champ P-STD_buffer_size est applicable dès que le décodeur modèle du système selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 reçoit ce champ. Voir 2.7.7.

La taille BS_n sera supérieure ou égale à la taille du tampon d'image codée (CPB) signalée par la taille CpbSize[cpb_cnt_minus1] spécifiée par les paramètres NAL hrd_parameters() dans le flux vidéo AVC. Si les paramètres NAL hrd_parameters() sont absents du flux vidéo AVC, la taille BS_n sera supérieure ou égale à la taille du tampon CPB NAL pour le format de flux d'octets défini à l'Annexe A de la Rec. UIT-T H.264 | ISO/CEI 14496-10 comme étant de 1200 × MaxCPB pour le niveau appliqué.

PES_extension_field_length (longueur du champ d'extension de paquet PES) – Codé sur 7 éléments binaires, ce champ spécifie la longueur, en octets, des données qui suivent ce champ dans le champ PES_extension jusqu'aux éventuels octets correspondant à des valeurs réservées et y compris ces derniers.

stream_id_extension_flag – Fanion occupant un bit indiquant, lorsqu'il est mis à 0 la présence d'un champ stream_id_extension dans l'en-tête du paquet PES. La valeur 1 pour ce fanion est une valeur réservée.

stream_id_extension – Dans les flux de programme, stream_id_extension indique le type et le numéro du flux élémentaire tel que défini par le Tableau 2-27. Dans les flux de transport, stream_id_extension peut prendre une valeur quelconque valide qui décrit correctement le flux élémentaire tel que défini dans le Tableau 2-27. Dans les flux de transport, le type de flux élémentaire est spécifié dans l'information spécifique du programme définie au § 2.4.4. A noter que ce champ est utilisé comme extension de l'identificateur stream_id défini plus haut. Ce champ ne doit pas être utilisé sauf si la valeur de l'identificateur stream_id est 1111 1101.

Tableau 2-27 – Affectation de l'extension stream_id_extension

stream_id_extension	Note	Codage du champ
000 0000	1	Flux d'informations de commande IPMP
000 0001	2	Flux IPMP
000 0010 … 011 1111		reserved_data_stream
100 0000 … 111 1111		private_stream

NOTE 1 – Les paquets PES de l'extension stream_id_extension 0b000 0000 (Flux d'informations de commande IPMP) ont une syntaxe unique spécifiée dans l'ISO/CEI 13818-11 (MPEG-2 IPMP).

NOTE 2 – Les paquets PES de l'extension stream_id_extension 0b000 0001 (Flux IPMP) ont une syntaxe unique spécifiée dans l'ISO/CEI 13818-11 (MPEG-2 IPMP).

stuffing_byte (octet de bourrage) – Codé sur 8 éléments binaires, ce champ a une valeur fixe égale à '1111 1111' qui peut être insérée par le codeur, pour répondre par exemple aux nécessités du canal. Ce champ est ignoré par le décodeur. Chaque en-tête de paquet PES ne peut contenir plus de 32 octets de bourrage.

PES_packet_data_byte (octet de données contenues dans un paquet PES) – Ce champ se compose d'octets contigus de données issues du flux élémentaire indiqué par l'identificateur de flux ou par l'identificateur de paquet. Si les données du flux élémentaire sont conformes à la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 13818-3, les octets de données contenues dans un paquet PES doivent être alignés par les octets, sur les octets conformes à la présente Recommandation | Norme internationale. L'ordre des octets dans le flux élémentaire doit être conservé. Le nombre N d'octets de données contenues dans un paquet PES est spécifié dans le champ PES_packet_length. N doit être égal à la valeur indiquée dans le champ PES_packet_length, moins le nombre d'octets se trouvant entre le dernier octet du champ PES_packet_length et le premier octet de données contenues dans un paquet PES.

Dans le cas de private_stream_1, private_stream_2, ECM_stream ou EMM_stream, le contenu du champ PES_packet_data_byte peut être défini par l'utilisateur et ne sera spécifié par l'UIT-T | ISO/CEI dans le futur.

padding_byte (octet de bourrage) – Codé sur 8 éléments binaires, ce champ a une valeur fixe égale à '1111 1111'. Ce champ est ignoré par le décodeur.

2.4.3.8 Acheminement de flux de programme et de flux de système ISO/CEI 11172-1 dans le flux de transport

Le flux de transport contient des champs facultatifs destinés à faciliter l'acheminement des flux de programme et des flux de systèmes ISO/CEI 11172-1, en simplifiant la reconstruction des flux respectifs dans le décodeur.

Lorsqu'on place un flux de programme dans un flux de transport, les paquets PES de flux de programme dont le champ stream_id (identificateur de flux) prend des valeurs de flux de données privées 1, de données vidéo selon la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 11172-2 et de données audio selon ISO/CEI 13818-3 ou ISO/CEI 11172-3, sont acheminés dans des paquets de flux de transport.

Lors de la reconstruction du flux de programme au niveau du décodeur de flux de transport, les données de ces paquets PES sont copiées sur le flux de programme reconstruit.

En ce qui concerne les paquets PES de flux de programme dont le champ stream_id prend les valeurs program_stream_map, padding_stream, private_stream_2, ECM, EMM, DSM-CC_stream ou program_stream_directory, tous les octets du paquet de flux de programme, sauf ceux du code de début de bloc de paquets, sont placés dans les champs réservés aux octets de données d'un nouveau paquet PES. L'identificateur stream_id de ce nouveau paquet PES prend la valeur ancillary_stream (flux auxiliaire) (voir Tableau 2-22). Ce nouveau paquet PES est ensuite acheminé dans des paquets de flux de transport.

Lors de la reconstruction du flux de programme au niveau du décodeur de flux de transport, le préfixe du code de déclenchement de paquet des paquets PES contenant un identificateur stream_id dont la valeur est ancillary_stream, est copié sur le flux de programme reconstruit, suivi des champs d'octets de données provenant de ces paquets PES de flux de transport.

Les flux ISO/CEI 11172-1 sont acheminés dans des flux de transport en remplaçant tout d'abord les en-têtes de paquet ISO/CEI 11172-1 par des en-têtes de paquet PES UIT-T H.262 | ISO/CEI 13818-2, puis en copiant les valeurs du champ d'en-tête de paquet ISO/CEI 11172-1 sur les champs équivalents d'en-tête de paquet PES UIT-T H.262 | ISO/CEI 13818-2.

Le champ du compteur de séquence de paquet de programme figure dans l'en-tête de chaque paquet PES contenant des données provenant d'un flux de programme ou d'un flux de système ISO/CEI 11172-1. Cela permet de restituer au niveau du décodeur l'ordre des paquets PES du flux de programme d'origine ou celui des paquets du flux de système ISO/CEI 11172-1.

Le champ destiné à l'en-tête de bloc d'un flux de programme ou d'un flux de système ISO/CEI 11172-1 est acheminé dans le flux de transport à l'intérieur de l'en-tête du paquet immédiatement suivant.

2.4.4 Informations spécifiques du programme

Les informations spécifiques du programme (PSI, *program specific information*) comprennent aussi bien les données normatives selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 et les données privées qui permettent le démultiplexage de programmes par des décodeurs. Les programmes se composent d'un ou de plusieurs flux élémentaires, chacun étant désigné par un identificateur PID. Les flux élémentaires de programmes, ou certaines parties d'entre eux, peuvent être embrouillés de manière à faire l'objet d'un accès conditionnel. Les informations spécifiques du programme ne doivent cependant pas être embrouillées.

Dans les flux de transport, les informations spécifiques du programme sont classées en six structures tabulaires, comme indiqué dans le Tableau 2-28. Bien que ces structures puissent être considérées comme de simples tables, elles doivent être segmentées en sections et être insérées dans des paquets du flux de transport, certaines avec des identificateurs PID prédéfinis, d'autres avec des identificateurs PID sélectionnables par l'utilisateur.

Tableau 2-28 – Informations spécifiques du programme

Nom de la structure	Type de flux	Identificateur PID	Description
Table d'association de programmes	Rec. UIT-T H.222.0 | ISO/CEI 13818-1	0x00	Associe le numéro de programme et l'identificateur PID de la table de mappage de programme
Table de mappage de programme	Rec. UIT-T H.222.0 | ISO/CEI 13818-1	Valeur assignée dans la table d'association de programmes	Spécifie les valeurs d'identificateur PID pour des composantes d'un ou de plusieurs programmes
Table d'informations relatives au réseau	Données privées	Valeur assignée dans la table d'association de programmes	Paramètres physiques du réseau tels que fréquences MRF, numéros de répéteur, etc.
Table d'accès conditionnel	Rec. UIT-T H.222.0 | ISO/CEI 13818-1	0x01	Associe un ou plusieurs flux de messages EMM (privés) ayant chacun une unique valeur d'identificateur PID
Table de descripteurs de flux de transport	Rec. UIT-T H.222.0 | ISO/CEI 13818-1	0x02	Associe un ou plusieurs descripteurs du Tableau 2-45 pour former un flux de transport complet
Table d'informations de commande IPMP	Rec. UIT-T H.222.0 | ISO/CEI 13818-1	0x03	Contient la liste d'outils, le conteneur de droits et le conteneur d'outils IPMP définis dans l'ISO/CEI 13818-11

Les tables PSI définies par la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 doivent être segmentées en une ou plusieurs sections acheminées dans les paquets de transport. Une section est une structure syntaxique qui sera utilisée pour mapper les tables PSI dans les paquets de transport.

De concert avec les tables d'informations PSI définies dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, il est possible d'acheminer des tables de données privées. La présente Spécification ne définit pas les moyens d'acheminement d'informations privées dans des paquets de flux de transport. Ces paquets pourront être structurés comme ceux qui sont utilisés pour transporter les tables d'informations PSI définies dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, de manière que la syntaxe de mappage de ces données privées soit la même que pour le mappage des tables d'informations PSI définies dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Une section relative à des données privées est définie à cette fin. Si les données privées sont acheminées dans des paquets de transport possédant la même valeur d'identificateur PID que les paquets de transport acheminant des tables de mappage de programme (identifiées dans la table d'association de programmes), la syntaxe et la sémantique du champ private_section doivent être utilisées. Les données acheminées dans les octets de données privées peuvent être embrouillées; par contre aucun autre champ de la section de données privées ne doit l'être. Cette section de données privées permet de transmettre des données au moyen d'une structure réduite au minimum. Si cette structure n'est pas utilisée, le mappage des données privées sur les paquets du flux de transport n'est pas défini par la présente Recommandation | Norme internationale.

La longueur des sections peut varier. Le début d'une section est indiqué par un champ de pointeur contenu dans la capacité utile d'un paquet de transport. La syntaxe du champ pointer_field est spécifiée dans le Tableau 2-29.

Les paquets de flux de transport acheminant des sections peuvent contenir des champs d'adaptation.

Dans un flux de transport, on peut trouver des octets de bourrage de paquet de valeur 0xFF dans la capacité utile des paquets de flux de transport acheminant des PSI ou des sections privées après le dernier octet d'une section uniquement. Dans ce cas, tous les octets doivent, jusqu'à la fin du paquet de transport, être aussi des octets de bourrage de valeur 0xFF. Ces octets peuvent être ignorés par un décodeur. En pareil cas, la capacité utile du paquet suivant de flux de transport ayant la même valeur d'identification PID doit commencer par un champ de pointeur de valeur 0x00, indiquant que la section suivante commencera immédiatement après.

Chaque flux de transport doit contenir un ou plusieurs paquets de transport identifiés par un PID de valeur 0x0000. L'ensemble de ces paquets de transport doit contenir une liste complète de tous les programmes contenus dans le flux de transport. La version la plus récemment transmise de la table dont le champ current_next_indicator est mis à la valeur '1' doit toujours s'appliquer aux données actuelles du flux de transport. Toute modification éventuelle des programmes acheminés dans le flux de transport doit être décrite dans une version actualisée de la table d'association de programmes contenue dans des paquets de transport ayant une identification PID de valeur 0x0000. Ces sections doivent toutes utiliser la valeur 0x00 pour identifier la table. Seules les sections dont le champ table_id a cette valeur sont autorisées à l'intérieur des paquets de transport ayant un PID de valeur 0x0000. Pour qu'une nouvelle version de la table d'association de programmes devienne valide, toutes les sections (d'après les indications du champ last_section_number) dont les champs new version_number et current_next_indicator sont mis à la valeur '1' doivent sortir de la mémoire B_{sys} définie dans le décodeur T-STD (voir 2.4.2). La table PAT est valide dès que le dernier octet de la section nécessaire pour compléter la table a quitté B_{sys}.

Dès qu'au moins un flux élémentaire contenu dans un flux de transport est embrouillé, les paquets de transport ayant un PID de valeur 0x0001 doivent être transmis avec des sections d'accès conditionnel contenant des descripteurs d'accès conditionnel appropriés à ce flux embrouillé. Les paquets de transport émis doivent, ensemble, former une même version de la table d'accès conditionnel. La version transmise le plus récemment de la table dont le champ current_next_indicator est mis à la valeur '1' doit toujours s'appliquer aux données en cours dans le flux de transport. Toute modification éventuelle de l'embrouillage, rendant la table existante non valide ou incomplète, doit être décrite dans une version actualisée de la table d'accès conditionnel. Ces sections doivent toutes utiliser la valeur d'identification de table (table_id) 0x01. Seules les sections ayant cette valeur d'identification sont autorisées à l'intérieur des paquets de transport dont l'identificateur PID a la valeur 0x0001. Pour qu'une nouvelle version de la table d'accès conditionnel devienne valide, toutes les sections (d'après les indications du champ last_section_number) dont les champs new version_number et current_next_indicator sont mis à la valeur '1' doivent sortir de la mémoire B_{sys}. La table CAT est valide dès que le dernier octet de la section nécessaire pour compléter la table a quitté B_{sys}.

Chaque flux de transport doit contenir un ou plusieurs paquets de transport dont les identificateurs PID ont des valeurs définies conformément à la table d'association de programmes comme désignant des paquets de transport contenant des sections de mappage de programme pour un flux de transport. Chaque programme cité dans la table d'association de programmes doit être décrit dans une seule et unique section de mappage de programme pour un flux de transport. Chaque programme doit être défini entièrement à l'intérieur du flux de transport proprement dit. Les données privées comportant un champ identificateur PID de paquet de flux élémentaire associé dans la section appropriée de la table de mappage de programme font partie du programme. Le flux de transport peut contenir d'autres données privées, sans qu'elles soient nécessairement énumérées dans la section de la table de mappage de programme. La version transmise le plus récemment de la section de mappage de programme pour flux de transport, dont le champ current_next_indicator est mis à la valeur '1', doit toujours s'appliquer aux données en cours dans le flux de transport. Toute modification éventuelle de la définition d'un des programmes acheminés dans le flux de transport doit être décrite dans une version actualisée de la section correspondante de la table de mappage de programme acheminée dans les paquets de transport dont la valeur d'identificateur PID est définie comme étant l'identificateur de mappage pour ce programme spécifique. Tous les paquets de transport qui acheminent une section donnée de mappage de programme de flux de transport doivent avoir la même valeur d'identificateur PID. L'identificateur de paquet pour mappage de flux de programme (champ program_map_PID) ne doit pas changer pendant toute la durée d'existence d'un programme, y compris tous ses événements associés. Une définition de programme ne doit pas couvrir plusieurs sections de mappage de programme de flux de transport. Pour qu'une nouvelle version d'une section de mappage de programme de flux de transport devienne valide, le dernier octet de la section dont les champs new version_number et current_next_indicator sont mis à la valeur '1' doit être sorti de la mémoire B_{sys}.

Les sections ayant une valeur d'identification de table égale à 0x02 doivent contenir des informations relatives à la table de mappage de programme. De telles sections peuvent être acheminées dans des paquets de flux de transport ayant des valeurs d'identificateur PID différentes.

La table d'informations sur le réseau est facultative et son contenu est privé. Si elle est présente, elle est acheminée dans des paquets de flux de transport ayant la même valeur d'identification PID de réseau (champ appelé network_PID). La valeur du champ network_PID est définie par l'utilisateur et, si elle est présente, doit se trouver dans la table d'association de programmes, au numéro de programme réservé 0x0000. Si la table d'informations sur le réseau existe, elle doit prendre la forme d'une ou de plusieurs sections privées.

Le nombre maximal d'octets contenus dans une section de table d'informations PSI conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est de 1024 octets. Le nombre maximal d'octets contenus dans une section privée est de 4096 octets.

La table de descripteurs de flux de transport est facultative. Lorsqu'elle est présente, la description du flux de transport est acheminée dans des paquets de flux de transport ayant une valeur d'identificateur PID égale à 0x0002, comme indiqué dans le Tableau 2-28. Elle s'applique à l'ensemble du flux de transport. Les sections de la table de descripteurs de flux de transport utilisent un identificateur de table 0x03, spécifié dans le Tableau 2-31; son contenu est limité aux descripteurs spécifiés dans le Tableau 2-45. La section TS_description_section devient valable quand le dernier octet de la section qui doit terminer le tableau sort de B_{sys}.

Il n'y a pas de restrictions quant à l'apparition de codes de déclenchement, d'octets de synchronisation ou d'autres schémas binaires dans des données d'information PSI, qu'elles soient conformes à la présente Recommandation | Norme internationale ou privées.

2.4.4.1 Pointeur

La syntaxe du champ de pointeur est définie dans le Tableau 2-29 ci-dessous.

Tableau 2-29 – Pointeur sur les informations spécifiques du programme

Syntaxe	Nombre de bits	Mnémonique
pointer_field	**8**	**uimsbf**

2.4.4.2 Définition sémantique des champs dans la syntaxe du pointeur

pointer_field (champ de pointeur) – Codée sur 8 éléments binaires, la valeur de ce champ représente le nombre d'octets qui viennent immédiatement à la suite de ce champ, jusqu'à l'apparition du premier octet de la première section dans la capacité utile du paquet de flux de transport (de telle sorte qu'une valeur 0x00 du champ pointer_field indiquera que la section commence immédiatement après ce champ). Lorsque au moins une section commence dans un paquet de transport donné, l'indicateur de début d'unité de capacité utile (voir 2.4.3.2) doit être mis à 1 et le premier octet de la capacité utile de ce paquet de transport doit contenir le pointeur. Si aucune section ne commence dans un paquet de transport donné, l'indicateur de début d'unité de capacité utile doit être mis à 0 et aucun pointeur ne doit être envoyé dans la capacité utile de ce paquet.

2.4.4.3 Table d'association de programmes

La table d'association de programmes fournit la correspondance entre un numéro de programme et la valeur d'identificateur PID des paquets du flux de transport acheminant la définition du programme. Le numéro de programme est l'étiquette numérique associée à un programme.

L'ensemble de la table d'association de programmes doit être subdivisé en une ou plusieurs sections ayant la syntaxe ci-après (voir Tableau 2-30).

Tableau 2-30 – Section relative aux associations de programmes

Syntaxe	Nombre de bits	Mnémonique
program_association_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
transport_stream_id	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
program_number	**16**	**uimsbf**
reserved	**3**	**bslbf**
if (program_number == '0') {		
network_PID	**13**	**uimsbf**
}		
else {		
program_map_PID	**13**	**uimsbf**
}		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.4 Affectation des identificateurs de table

Le champ table_id identifie le contenu d'une section d'informations PSI pour flux de transport, comme indiqué dans le Tableau 2-31 ci-dessous.

Tableau 2-31 – Valeurs d'affectation des identificateurs de table

Valeur	Description
0x00	program_association_section
0x01	conditional_access_section (CA_section)
0x02	TS_program_map_section
0x03	TS_description_section
0x04	ISO_IEC 14496_scene_description_section
0x05	ISO_IEC 14496_object_descriptor_section
0x06	Metadata_sections
0x07	IPMP_Control_Information_section (défini dans l'ISO/CEI 13818-11)
0x08-0x3F	Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (valeur réservée)
0x40-0xFE	Valeur privée d'utilisateur
0xFF	Valeur interdite

2.4.4.5 Définition sémantique des champs contenus dans une section d'association de programmes

table_id (identificateur de table) – Codé sur 8 éléments binaires, ce champ doit être mis à la valeur 0x00 comme indiqué dans le Tableau 2-31.

section_syntax_indicator (indicateur de syntaxe de section) – Ce fanion codé sur 1 bit doit être mis à '1'.

section_length (longueur de section) – Codé sur 12 éléments binaires (dont les deux premiers doivent être '00'), ce champ spécifie le nombre d'octets de la section qui commence immédiatement après le champ section_length, y compris le contrôle CRC. La valeur codée dans ce champ ne doit pas dépasser 1021 (0x3FD).

transport_stream_id (identificateur de flux de transport) – Codé sur 16 éléments binaires, ce champ sert d'étiquette pour identifier un flux de transport parmi d'autres multiplex quelconques d'un réseau. Sa valeur est définie par l'utilisateur.

version_number (numéro de version) – Codé sur 5 éléments binaires, ce champ est le numéro de version de l'ensemble de la table d'association de programmes. Le numéro de version sera incrémenté de la valeur 1 à chaque modification de la définition de la table d'association de programmes, puis ramené à sa valeur modulo 32. Lorsque le champ current_next_indicator est mis à '1', le numéro de version doit être celui de la table d'association de programmes actuellement applicable. Lorsque le champ current_next_indicator est mis à '0', le numéro de version doit être celui de la prochaine table d'association de programmes applicable.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la table d'association de programmes envoyée est actuellement applicable. Lorsqu'il est mis à '0', il indique que la table envoyée n'est pas encore applicable et doit être la prochaine table à devenir valide.

section_number (numéro de section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la section. Le numéro de la première section dans la table d'association de programmes doit être 0x00. Il doit augmenter de 1 à chaque nouvelle section contenue dans la table d'association de programmes.

last_section_number (numéro de la dernière section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la dernière section (c'est-à-dire celle dont le numéro de section est le plus élevé) de la table d'association de programmes complète.

program_number (numéro de programme) – Codé sur 16 éléments binaires, ce champ spécifie le programme auquel l'identificateur de paquet de conditionnement de flux de programme est applicable. S'il est mis à la valeur 0x0000, la prochaine référence PID doit être celle du réseau. Pour tous les autres cas, la valeur de ce champ est définie par l'utilisateur. Ce champ ne doit pas prendre plus d'une seule fois une valeur quelconque dans la même version de la table d'association de programmes.

NOTE – Le numéro de programme peut être utilisé pour désigner, par exemple, un canal de diffusion.

network_PID (identificateur de réseau) – Codé sur 13 éléments binaires et utilisé seulement conjointement avec la valeur du program_number mise à 0x0000, ce champ spécifie l'identificateur PID des paquets de transport qui doivent

contenir la table d'association de programmes. La valeur de ce champ est définie par l'utilisateur mais ne doit pas prendre d'autres valeurs que celles spécifiées au Tableau 2-3. La présence de ce champ est facultative.

program_map_PID (identificateur de paquet de mappage de flux de programme) – Codé sur 13 éléments binaires, ce champ spécifie l'identificateur PID des paquets de transport qui doivent contenir la section de mappage de programme applicable aux programmes spécifiés par le champ program_number. Aucun numéro de programme ne doit être affecté à plus d'un seul identificateur de paquet de mappage de flux de programme. La valeur de ce champ est définie par l'utilisateur mais ne doit pas prendre d'autres valeurs que celles spécifiées au Tableau 2-3.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 éléments binaires, ce champ contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe A après traitement de toute la section d'association de programmes.

2.4.4.6 Table d'accès conditionnel

La table d'accès conditionnel (CA, *conditional access*) fournit l'association entre un ou plusieurs systèmes d'accès conditionnel, leurs flux de messages EMM et tous paramètres spéciaux pouvant leur être associés. Voir 2.6.16, qui contient une définition du champ descriptor() figurant dans le Tableau 2-32.

Le tableau peut être subdivisé en une ou plusieurs sections, avant insertion dans des paquets de transport, avec la syntaxe suivante.

Tableau 2-32 – Section relative à l'accès conditionnel

Syntaxe	Nombre de bits	Mnémonique
CA_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
reserved	**18**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.7 Définition sémantique des champs dans une section d'accès conditionnel

table_id (identificateur de table) – Codé sur 8 éléments binaires, ce champ doit toujours être mis à la valeur 0x01, comme indiqué dans le Tableau 2-31.

section_syntax_indicator (indicateur de syntaxe de section) – Ce fanion codé sur 1 bit doit toujours être mis à '1'.

section_length (longueur de section) – Codé sur 12 éléments binaires (dont les deux premiers doivent être '00'), ce champ spécifie le nombre d'octets de la section qui commence immédiatement après le champ section_length, y compris le contrôle CRC. La valeur codée dans ce champ ne doit pas dépasser 1021 (0x3FD).

version_number (numéro de version) – Codé sur 5 éléments binaires, ce champ est le numéro de version de l'ensemble de la table d'accès conditionnel. Le numéro de version doit augmenter d'une unité modulo 32 à chaque modification des informations acheminées dans la table d'accès conditionnel. Lorsque le champ current_next_indicator est mis à '1', le numéro de version doit être celui de la table d'accès conditionnel actuellement applicable. Lorsque le champ current_next_indicator est mis à '0', le numéro de version doit être celui de la prochaine table d'accès conditionnel applicable.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la table d'accès conditionnel envoyée est actuellement applicable. Lorsqu'il est mis à '0', il indique que la table envoyée n'est pas encore applicable et doit être la prochaine table d'accès conditionnel à devenir valide.

section_number (numéro de section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la section. Le numéro de la première section dans la table d'accès conditionnel doit être 0x00. Il doit augmenter de 1 à chaque nouvelle section contenue dans la table d'accès conditionnel.

last_section_number (numéro de la dernière section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la dernière section (c'est-à-dire celle dont le numéro de section est le plus élevé) de la table d'accès conditionnel complète.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 éléments binaires, ce champ contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe A après traitement de toute la section d'accès conditionnel.

2.4.4.8 Table de mappage de programme

La table de mappage de programme fournit les mappages entre les numéros de programme et les éléments de programme qui les acheminent. Une seule occurrence d'un tel mappage est appelée "définition de programme". La table de mappage de programme est l'ensemble complet de toutes les définitions de programme pour un flux de transport. Cette table doit être transmise sous forme de paquets, dont les identificateurs PID ont des valeurs sélectionnées par le codeur. Plusieurs valeurs de PID peuvent être utilisées, le cas échéant. Cette table doit être segmentée en une ou plusieurs sections, avant d'être insérée dans les paquets de flux de transport, avec la syntaxe suivante. Dans chaque section le champ du numéro de section doit être mis à '0'. Les sections sont identifiées par la valeur affectée au champ numéro de programme.

La description pour les champs descriptor() se trouve au 2.6 (voir Tableau 2-33).

Tableau 2-33 – Section relative au mappage du flux de programme pour le flux de transport

Syntaxe	Nombre de bits	Mnémonique
TS_program_map_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
program_number	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
reserved	**3**	**bslbf**
PCR_PID	**13**	**uimsbf**
reserved	**4**	**bslbf**
program_info_length	**12**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
for (i = 0; i < N1; i++) {		
stream_type	**8**	**uimsbf**
reserved	**3**	**bslbf**
elementary_PID	**13**	**uimsbf**
reserved	**4**	**bslbf**
ES_info_length	**12**	**uimsbf**
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.9 Définition sémantique des champs contenus dans la section de mappage de flux de programme pour le flux de transport

table_id (identificateur de table) – Codé sur 8 éléments binaires, ce champ doit toujours – dans le cas d'une section de mappage de programme pour le flux de transport – être mis à la valeur 0x02, comme indiqué dans le Tableau 2-31.

section_syntax_indicator (indicateur de syntaxe de section) – Ce fanion codé sur 1 bit doit toujours être mis à '1'.

section_length (longueur de section) – Codé sur 12 éléments binaires (dont les deux premiers doivent être '00'), ce champ spécifie le nombre d'octets de la section qui commence immédiatement après le champ section_length, y compris le contrôle CRC. La valeur de codage de ce champ ne doit pas dépasser 1021 (0x3FD).

program_number (numéro de programme) – Codé sur 16 éléments binaires, ce champ spécifie le programme auquel l'identificateur de paquet de mappage de flux de programme est applicable. Une seule définition de programme doit être acheminée dans chaque section de mappage de programme pour le flux de transport. Cela implique que la longueur d'une définition de programme ne dépasse jamais 1016 octets (0x3F8). Voir l'Annexe C (informative), qui indique des moyens de traiter les cas où cette longueur serait insuffisante. Le numéro de programme peut être utilisé pour désigner, par exemple, un canal de diffusion. Pour décrire les différents éléments de programme appartenant à un programme, les données issues de différentes sources (par exemple des événements séquentiels) peuvent être concaténées afin de former une suite continue de flux utilisant le même numéro de programme. Voir en Annexe C des exemples d'applications.

version_number (numéro de version) – Codé sur 5 éléments binaires, ce champ est le numéro de version de la section de mappage de programme pour le flux de transport. Le numéro de version doit augmenter d'une unité modulo 32 à chaque modification des informations acheminées dans cette section. Le numéro de section correspond à la définition d'un programme donné et donc à une section et une seule. Lorsque le champ current_next_indicator est mis à '1', le numéro de version doit être celui de la section de mappage de programme pour le flux de transport actuellement applicable. Lorsque le champ current_next_indicator est mis à '0', le numéro de version doit être celui de la prochaine section de mappage de programme pour le flux de transport applicable.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la section de mappage de programme pour le flux de transport envoyée est actuellement applicable. Lorsqu'il est mis à '0', il indique que la table envoyée n'est pas encore applicable et doit être la prochaine section de mappage de programme pour le flux de transport à devenir valide.

section_number (numéro de section) – Codé sur 8 éléments binaires, ce champ doit avoir la valeur 0x00.

last_section_number (numéro de la dernière section) – Codé sur 8 éléments binaires, ce champ doit avoir la valeur 0x00.

PCR_PID (identificateur de paquet pour la référence PCR) – Codé sur 13 éléments binaires, ce champ indique le PID des paquets de transport qui doivent contenir les champs PCR pour le programme spécifié par la structure program_number. Si aucune référence PCR n'est associée à une définition de programme pour des flux privés, ce champ doit prendre la valeur 0x1FFF. Se reporter à la définition sémantique de la référence PCR donnée au 2.4.3.5 et au Tableau 2-3 en ce qui concerne les restrictions applicables au choix de la valeur affectée à l'identificateur PCR_PID.

program_info_length (longueur des descripteurs de flux de programme) – Codé sur 12 éléments binaires (dont les 2 premiers doivent être '00'), ce champ spécifie sur les 10 bits restants le nombre d'octets dont se composent les descripteurs suivant immédiatement le champ program_info_length.

stream_type (type de flux) – Codé sur 8 éléments binaires, ce champ spécifie le type d'élément de programme acheminé dans les paquets dont le PID a une valeur spécifiée par la structure elementary_PID. Les valeurs de ce champ sont spécifiées dans le Tableau 2-34.

NOTE – Un flux auxiliaire de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est disponible pour les types de données définis par la présente Spécification autres que audio, vidéo, DSM-CC, tels que "Program Stream Directory" et "Program Stream Map".

Tableau 2-34 – Assignation des types de flux

Valeur	Description
0x00	Réservé à l'UIT-T | ISO/CEI
0x01	Vidéo ISO/CEI 11172-2
0x02	Vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2 ou flux vidéo à paramètres contraints ISO/CEI 11172-2
0x03	Audio ISO/CEI 11172-3
0x04	Audio ISO/CEI 13818-3
0x05	Private_sections Rec. UIT-T H.222.0 | ISO/CEI 13818-1
0x06	Paquets PES contenant des données privées Rec. UIT-T H.222.0 | ISO/CEI 13818-1
0x07	MHEG ISO/CEI 13522
0x08	Annexe A DSM-CC Rec. UIT-T H.222.0 | ISO/CEI 13818-1
0x09	Rec. UIT-T H.222.1
0x0A	ISO/CEI 13818-6 type A

Tableau 2-34 – Assignation des types de flux

Valeur	Description
0x0B	ISO/CEI 13818-6 type B
0x0C	ISO/CEI 13818-6 type C
0x0D	ISO/CEI 13818-6 type D
0x0E	Auxiliaire Rec. UIT-T H.222.0 | ISO/CEI 13818-1
0x0F	Audio ISO/CEI 13818-7 avec syntaxe de transport ADTS
0x10	Visuel ISO/CEI 14496-2
0x11	Audio ISO/CEI 14496-3 avec syntaxe de transport LATM telle que définie dans l'ISO/CEI 14496-3
0x12	Flux de paquets SL ISO/CEI 14496-1 ou flux FlexMux acheminé dans des paquets PES
0x13	Flux de paquets SL ISO/CEI 14496-1 ou flux FlexMux acheminé dans ISO/CEI 14496_sections
0x14	Protocole de téléchargement synchronisé ISO/CEI 13818-6
0x15	Métadonnées acheminées dans des paquets PES
0x16	Métadonnées acheminées dans metadata_sections
0x17	Métadonnées acheminées dans un carrousel de données ISO/CEI 13818-6
0x18	Métadonnées acheminées dans un carrousel d'objets ISO/CEI 13818-6
0x19	Métadonnées acheminées dans le protocole de téléchargement synchronisé ISO/CEI 13818-6
0x1A	Flux IPMP (défini dans l'ISO/CEI 13818-11, IPMP MPEG-2)
0x1B	Flux vidéo AVC tel que défini dans la Rec. UIT-T H.264 | ISO/CEI 14496-10
0x1C-0x7E	Réservés à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1
0x7F	Flux IPMP
0x80-0xFF	Réservés à l'utilisateur

elementary_PID (identificateur de paquet de flux élémentaire) – Codé sur 13 éléments binaires, ce champ spécifie l'identificateur de paquets de flux élémentaire acheminant l'élément de programme associé.

ES_info_length (longueur des descripteurs de flux élémentaire) – Codé sur 12 éléments binaires (dont les 2 premiers doivent être '00'), ce champ spécifie sur les 10 bits restants le nombre d'octets dont se composent les descripteurs de l'élément de programme associé suivant immédiatement le champ ES_info_length.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 éléments binaires, ce champ contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe B après traitement de toute la section de mappage de programme pour le flux de transport.

2.4.4.10 Syntaxe de la section relative aux données privées

L'utilisation du champ de section relative aux données privées doit être faite lorsque des données privées sont envoyées dans des paquets de transport dont le PID a une valeur désignée, dans la table d'association de programmes, comme identifiant un paquet de table de mappage de flux de programme. Cette section privée permet de transmettre des données avec le minimum absolu de structure permettant à des décodeurs d'analyser la syntaxe du flux. On peut utiliser ces sections de deux manières: si l'indicateur de syntaxe de section est à '1', on doit utiliser l'ensemble de la structure syntaxique commune à toutes les tables; s'il est à '0', seuls les champs 'table_id' jusqu'à 'private_section_length' doivent être conformes à la syntaxe et à la sémantique structurelles communes, le reste de la section privée pouvant prendre toute forme déterminée par l'utilisateur. On trouvera dans l'Annexe C (informative) des exemples d'utilisation étendue de cette syntaxe.

Une table privée peut se composer de plusieurs sections privées, ayant toutes le même identificateur de table (voir Tableau 2-35).

Tableau 2-35 – Section relative aux données privées

Syntaxe	Nombre de bits	Mnémonique
private_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
private_section_length	**12**	**uimsbf**
if (section_syntax_indicator == '0') {		
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
else {		
table_id_extension	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < private_section_length-9; i++) {		
private_data_byte	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		
}		

2.4.4.11 Définition sémantique des champs contenus dans une section de données privées

table_id (identificateur de table) – Codé sur 8 éléments binaires, ce champ spécifie la table de données privées à laquelle la section appartient. Certaines valeurs sont réservées (voir Tableau 2-31).

section_syntax_indicator (indicateur de syntaxe de section) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la section privée suit la syntaxe générique de section au-delà du champ private_section_length. S'il est mis à '0', il indique que les octets de données privées suivent immédiatement le champ private_section_length.

private_indicator (indicateur de données privées) – Ce fanion codé sur 1 bit, définissable par l'utilisateur, ne doit pas faire l'objet d'une future spécification par l'UIT-T | ISO/CEI.

private_section_length – Codé sur 12 éléments binaires, ce champ indique le nombre d'octets restants dans la section de données privées suivant immédiatement le champ private_section_length, jusqu'à la fin de la section privée. La valeur de ce champ ne doit pas dépasser 4093 (0xFFD).

private_data_byte (octet de données privées) – Ce champ est définissable par l'utilisateur et ne doit pas faire l'objet d'une future spécification par l'UIT-T | ISO/CEI.

table_id_extension (extension d'identificateur de table) – Ce champ est codé sur 16 éléments binaires. Son usage et sa valeur sont définis par l'utilisateur.

version_number (numéro de version) – Codé sur 5 éléments binaires, ce champ est le numéro de version de la section de données privées. Le numéro de version doit augmenter d'une unité modulo 32 à chaque modification des informations acheminées dans la section de données privées. Lorsque le champ current_next_indicator est mis à '0', le numéro de version doit être celui de la prochaine section de données privées applicable, ayant le même identificateur de table et le même numéro de section.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la section de données privées envoyée est actuellement applicable et que le numéro de version doit être celui de la section privée actuellement applicable. Lorsqu'il est mis à '0', il indique que la table envoyée n'est pas encore applicable et doit être la prochaine section de données privées, ayant le même numéro de section et le même identificateur de table, à devenir valide.

section_number (numéro de section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la section de données privées. Pour la première section d'une table privée, ce numéro doit être 0x00. Le numéro augmente d'une unité à chaque nouvelle section de la table de données privées.

last_section_number (numéro de la dernière section) – Codé sur 8 éléments binaires, ce champ spécifie le numéro de la dernière section (c'est-à-dire celle dont le numéro est le plus élevé) de la table de données privées à laquelle cette section appartient.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 éléments binaires, ce champ contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe A après traitement de toute la section de données privées.

2.4.4.12 Syntaxe de la section du flux de transport

Les flux de bits conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 peuvent acheminer l'information définie dans le Tableau 2-36 qui suit et les décodeurs conformes à cette même Recommandation | Norme internationale peuvent la décoder.

La table de descripteurs de flux de transport est définie pour prendre en charge l'acheminement de descripteurs tels que ceux qui sont décrits au 2.6 pour un flux de transport complet. Ces descripteurs s'appliqueront au flux de transport complet. Ce tableau utilise une valeur d'identificateur de table égale à 0x03, spécifiée dans le Tableau 2-31, qui est acheminée dans des paquets de flux de données dont la valeur d'identificateur PID est égale à 0x0002, comme spécifié dans le Tableau 2-3.

Tableau 2-36 – Table de descripteurs de flux de transport

Syntaxe	Nombre de bits	Mnémonique
TS_description_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
reserved	**18**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.13 Définition sémantique des champs contenus dans une section de flux de transport

table_id (identificateur de table) – Codé sur 8 éléments binaires, ce champ doit être mis à la valeur '0x03' comme indiqué dans le Tableau 2-31.

section_length (longueur de section) – Codé sur 12 éléments binaires (dont les deux premiers doivent être '00'), ce champ spécifie le nombre d'octets de la section qui commence immédiatement après le champ section_length, y compris le contrôle CRC. La valeur codée dans ce champ ne doit pas dépasser 1021 (0x3FD).

version_number (numéro de version) – Codé sur 5 éléments binaires, ce champ est le numéro de version de l'ensemble de la table de descripteurs de flux de transport. Le numéro de version doit augmenter d'une unité modulo 32 à chaque modification de la définition de la table de descripteurs de flux de transport. Lorsque le champ current_next_indicator est mis à '1', le numéro de version doit être celui de la table de descripteurs de flux de transport actuellement applicable. Lorsque le champ current_next_indicator est mis à '0', le numéro de version doit être celui de la prochaine table de descripteurs de flux de transport applicable.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la table de descripteurs de flux de transport envoyée est actuellement applicable. Lorsqu'il est mis à '0', il indique que la table envoyée n'est pas encore applicable et doit être la prochaine table à devenir valide.

section_number (numéro de section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la section. Le numéro de la première section dans la table de descripteurs de flux de transport doit être 0x00. Il doit augmenter de 1 à chaque nouvelle section contenue dans la table de descripteurs de flux de transport.

last_section_number (numéro de la dernière section) – Codé sur 8 éléments binaires, ce champ indique le numéro de la dernière section (c'est-à-dire celle dont le numéro de section est le plus élevé) de la table de descripteurs de flux de transport complète.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 éléments binaires, ce champ contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini dans l'Annexe A après traitement de toute la section de description du flux de transport.

2.5 Prescriptions applicables au flux binaire de programme

2.5.1 Structure de codage et paramètres du flux de programme

La couche de codage du flux de programme de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 permet de combiner un ou plusieurs flux élémentaires afin de former un seul flux. Les données issues de chaque flux élémentaire sont multiplexées et codées avec les informations qui permettent une présentation des flux élémentaires en synchronisme avec un programme.

Un flux de programme se compose d'un ou de plusieurs flux élémentaires issus d'un même programme et multiplexés ensemble. Les flux élémentaires audio et vidéo sont composés d'unités d'accès.

Les données issues de flux élémentaires sont insérées dans des paquets PES. Un paquet PES se compose d'un en-tête suivi de données de paquet. Les paquets PES sont insérés dans des lots de flux de programme.

L'en-tête de paquet PES commence par un préfixe de début de paquet codé sur 32 bits qui identifie également le flux auquel les données de paquet appartiennent (voir Tableau 2-22). Cet en-tête de paquet PES peut contenir soit simplement un horodateur de présentation (PTS, *presentation time stamp*), soit des horodateurs de décodage (DTS, *decoding time stamp*) et de présentation. L'en-tête de paquet PES contient aussi un certain nombre de fanions annonçant une série de champs facultatifs. Les données d'un paquet PES contiennent un nombre variable d'octets contigus issus d'un même flux élémentaire.

Dans un flux de programme, les paquets de flux PES sont organisés en blocs. Un bloc de paquets commence par un en-tête de bloc et est suivi par zéro, un ou plusieurs paquets PES. L'en-tête de bloc de paquets commence par un préfixe de début de bloc codé sur 32 bits. L'en-tête de bloc de paquets est utilisé pour insérer des informations relatives à la gestion du temps et au débit.

Le flux de programme commence par un en-tête système qui peut, sur option, être répété. Cet en-tête système contient un répertoire des paramètres systèmes qui sont définis dans le flux.

La présente Recommandation | Norme internationale ne spécifie pas les données codées qui peuvent être utilisées dans le cadre de systèmes d'accès conditionnel. La présente Recommandation | Norme internationale propose cependant des mécanismes permettant aux diffuseurs de programmes de transporter et d'identifier ces données pour le traitement de décodage et pour repérer correctement les données ici spécifiées.

2.5.2 Décodeur modèle du système pour le flux de programme

La sémantique du flux de programme et les contraintes imposées à ces sémantèmes nécessitent des définitions exactes des événements d'arrivée d'octet et de décodage, avec les instants précis de ces événements. Les définitions nécessaires sont exposées dans la présente Spécification au moyen d'un décodeur fictif appelé décodeur modèle du système pour le flux de programme (P-STD).

Le décodeur P-STD est un modèle théorique qui sert à définir précisément ces événements et à donner une représentation du processus de décodage pendant la construction des flux de programme. Le décodeur P-STD n'est défini qu'à cette fin. Ni son architecture ni la chronologie décrites n'excluent la possibilité d'une lecture ininterrompue et synchronisée de flux de programme issus de divers décodeurs possédant des architectures différentes ou des ordonnancements chronologiques différents.

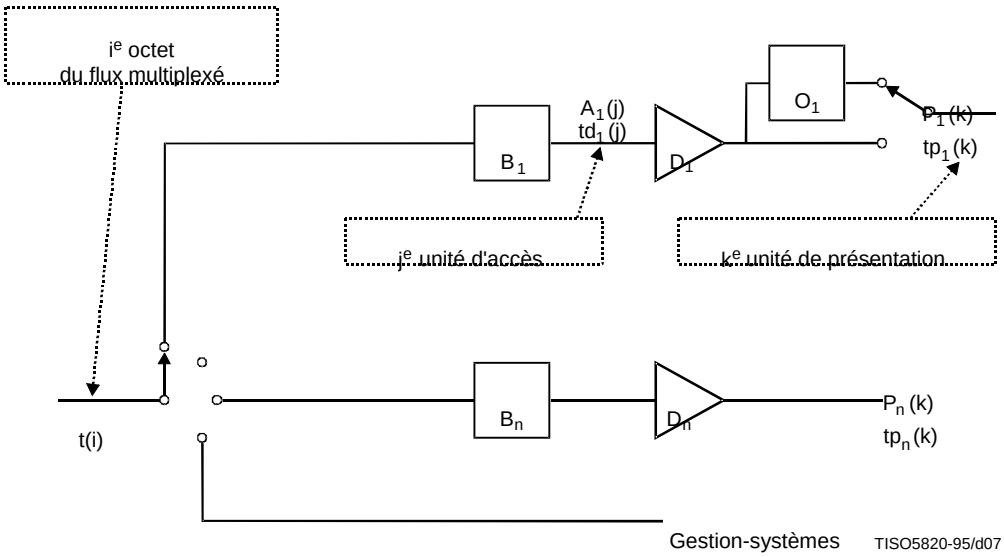


Figure 2-2 – Notation d'un décodeur modèle de système pour un flux de programme

La notation suivante (partiellement illustrée dans la Figure 2-2) est utilisée pour décrire le décodeur modèle du système pour le flux de programme:

i, i′	indices renvoyant aux octets du flux de programme. Le premier octet a l'indice 0.
j	indice renvoyant à des unités d'accès contenues dans les flux élémentaires.
k, k′, k″	indices renvoyant à des unités de présentation contenues dans les flux élémentaires.
n	indice renvoyant aux flux élémentaires.
t(i)	indique en secondes l'instant auquel le i^e octet du flux de programme entre dans le décodeur modèle du système. La valeur t(0) est une constante arbitrairement choisie.
SCR(i)	temps codé dans le champ de référence SCR, mesuré en unités d'horloge système de 27 MHz, où i est l'indice désignant l'octet final du champ de la base de référence temporelle du système.
$A_n(j)$	j^e unité d'accès dans le flux élémentaire n. Les $A_n(j)$ sont indexées dans l'ordre de décodage.
$td_n(j)$	instant, mesuré en secondes, du décodage dans le décodeur modèle du système de la j^e unité d'accès contenue dans le flux élémentaire n.
$P_n(k)$	k^e unité de présentation dans le flux élémentaire n; les $P_n(k)$ sont indexées dans l'ordre de présentation.
$tp_n(k)$	instant, mesuré en secondes, de présentation au décodeur STD, de la k^e unité de présentation contenue dans le flux élémentaire n.
t	temps, mesuré en secondes.
$F_n(t)$	remplissage, mesuré en octets de la mémoire d'entrée du décodeur STD au temps t pour le flux élémentaire n.
B_n	mémoire principale du décodeur STD pour le flux élémentaire n.
BS_n	capacité, en octets, de la mémoire principale du décodeur STD pour le flux élémentaire n.
D_n	décodeur pour le flux élémentaire n.
O_n	mémoire de réordonnancement du flux élémentaire n de données vidéo.

2.5.2.1 Fréquence de l'horloge système

Les informations de rythme indiquées dans le décodeur P-STD sont acheminées par plusieurs champs de données, définis dans la présente Spécification (voir 2.5.3.3 et 2.4.3.6). Ces informations sont codées dans les champs de référence SCR sous forme de valeur échantillonnée d'horloge système.

La valeur de fréquence d'horloge système est mesurée en hertz et doit répondre aux contraintes suivantes:

– 27 000 000 – 810 <= system_clock_frequency <= 27 000 000 + 810;

– dérive de la fréquence d'horloge système: <= 75×10^{-3} Hz/s.

La notation "system_clock_frequency" est utilisée plusieurs fois dans la présente Recommandation | Norme internationale pour désigner la fréquence d'une horloge répondant à ces prescriptions. Par commodité de notation, les équations mettant en jeu une référence SCR, un pointeur PTS ou un pointeur DTS fournissent des valeurs dont la précision est un multiple entier (en secondes) du rapport (300×2^{33}/system_clock_frequency). Ce rapport est dû au codage sur 33 bits des informations de calage de la référence SCR sur 1/300 de la fréquence d'horloge du système, plus 9 bits pour le reste des informations et au codage sur 33 bits de la fréquence d'horloge du système, divisée par 300 pour les pointeurs PTS et DTS.

2.5.2.2 Entrée dans le décodeur modèle du système pour le flux de programme

Les données issues du flux de programme pénètrent dans le décodeur P-STD. Le i^e octet entre au temps t(i). L'instant d'entrée de cet octet dans le décodeur P-STD peut être déduit du flux d'entrée par décodage des champs de référence temporelle du système (SCR) à l'entrée, et du champ program_mux_rate codés dans l'en-tête du bloc de paquets. La référence SCR est codée, selon l'équation 2-18, en deux parties: d'une part, en unités correspondant à 1/300 fois la fréquence d'horloge du système, formant la base de référence temporelle du système (équation 2-19) et, d'autre part, en unités de périodes correspondant à la fréquence d'horloge du système, formant l'extension de fréquence d'horloge système (équation 2-20). Dans ce qui suit, les valeurs codées dans ces champs seront désignées par SCR_base(i) et SCR_ext(i). La valeur codée dans le champ SCR(i) indique le moment t(i) auquel l'indice i renvoie à l'octet contenant le dernier bit du champ system_clock_reference_base.

Plus précisément:

$$SCR(i) = SCR_base(i) \times 300 + SCR_ext(i) \tag{2-18}$$

où:

$$SCR_base(i) = ((system_clock_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-19}$$

$$SCR_ext(i) = ((system_clock_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-20}$$

Le temps d'arrivée à l'entrée du décodeur, t(i), indiqué dans l'équation 2-21, doit être reconstruit pour tous les autres octets à partir de la référence SCR(i) et du débit auquel les données arrivent, ce débit d'arrivée étant, dans chaque bloc de paquets, la valeur du champ program_mux_rate figurant dans l'en-tête de ce bloc de paquets.

$$t(i) = \frac{SCR(i')}{system_clock_frequency} + \frac{i - i'}{program_mux_rate \times 50} \tag{2-21}$$

où:

i′ est l'indice de l'octet contenant le dernier bit du champ de base system_clock_reference contenu dans l'en-tête du bloc de paquets

i est l'indice de tout octet du bloc de paquets, y compris son en-tête

SCR(i′) est le temps codé dans les champs de base et d'extension de type system_clock_reference, en unités d'horloge système

program_mux_rate est un champ défini au 2.5.3.3.

Après acheminement du dernier octet d'un bloc de paquets, il peut s'écouler un moment pendant lequel aucun octet n'est acheminé à l'entrée du décodeur P-STD.

2.5.2.3 Mise en mémoire tampon

Les données de paquet PES issues d'un flux élémentaire n sont injectées dans la mémoire tampon d'entrée pour le flux n, B_n. Le transfert de l'octet i entre l'entrée du décodeur modèle du système et la mémoire B_n est instantané; en d'autres termes, l'octet i entre à l'instant t(i) dans la mémoire de capacité BS_n pour le flux n.

Les octets contenus dans l'en-tête d'un bloc de paquets, dans les en-têtes de données systèmes ou dans les en-têtes de paquets PES du flux de programme, tels que les champs SCR, DTS, PTS et packet_length, ne sont acheminés vers aucune des mémoires tampons mais peuvent être utilisés pour piloter le système.

Les capacités de mémoire d'entrée BS_1 à BS_n sont indiquées dans le paramètre de capacité mémoire du décodeur P-STD, selon la syntaxe des équations 2-16 et 2-17.

A l'instant du décodage, $td_n(j)$, toutes les données concernant l'unité d'accès restée le plus longtemps dans la mémoire d'entrée, $A_n(j)$, et tous les octets de bourrage qui les précèdent immédiatement et qui se trouvent dans la mémoire d'entrée au moment $td_n(j)$, sont retirés instantanément à l'instant $td_n(j)$. La valeur de cet instant de décodage $td_n(j)$ est spécifiée dans les champs DTS ou PTS. Les informations contenues dans le flux élémentaire permettent de calculer les instants de décodage $td_n(j + 1)$, $td_n(j + 2)$, ... des unités d'accès sans champs codés, suivant immédiatement l'unité d'accès j. Voir l'Annexe C de la Rec. UIT-T H.262 | ISO/CEI 13818-2, ISO/CEI 13818-3, ISO/CEI 11172-2 ou ISO/CEI 11172-3. Voir également les indications du 2.7.5. Lorsque l'unité d'accès est extraite de la mémoire, elle est instantanément décodée pour devenir une unité de présentation.

La construction du flux de programme, ainsi que le choix de l'instant t(i), doit permettre d'éviter conjointement le surremplissage et le sous-remplissage des mémoires d'entrée de capacité BS_1 à BS_n dans le décodeur modèle de système du programme. Autrement dit:

$$0 \leq F_n(t) \leq BS_n$$

quels que soient t et n

et:

$$F_n(t) = 0$$

immédiatement avant t = t(0).

$F_n(t)$ étant le remplissage instantané de la mémoire B_n du décodeur P-STD.

Cette condition est toujours vérifiée sauf dans le cas suivant: un sous-remplissage de la mémoire d'entrée B_n du décodeur P-STD est possible en cas de mise à la valeur '1' du fanion low_delay dans l'en-tête de séquence vidéo (voir 2.4.2.6) ou lorsque le mode d'enrichissement est activé (voir 2.4.3.8).

Pour tous les flux de programme, le délai dû à la mise en mémoire d'entrée du décodeur modèle de système doit demeurer inférieur ou égal à 1 seconde sauf pour les données vidéo d'images fixes et les flux ISO/CEI 14496. Le délai de mise en mémoire d'entrée est égal au temps écoulé entre l'introduction d'un octet dans la mémoire d'entrée et le décodage de cet octet.

Plus précisément: en l'absence de données vidéo d'images fixes, le délai est assujetti à la relation suivante:

$$td_n(j) - t(i) <= 1s$$

en présence de données vidéo d'images fixes, le délai doit vérifier la relation suivante:

$$td_n(j) - t(i) <= 60\ s$$

et en présence de flux ISO/CEI 14496, le délai doit vérifier la relation suivante:

$$td_n(j) - t(i) <= 10\ s$$

pour tous les octets contenus dans l'unité d'accès j.

Dans le cas des flux de programme, tant que tous les octets d'un bloc déterminé ne sont pas entrés dans le décodeur P-STD, aucun octet du bloc suivant ne doit y entrer.

Lorsque le fanion **low_delay** est mis à '1' dans l'extension de séquence vidéo (voir 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2), la mémoire du vérificateur VBV peut être sous-remplie. Dans ce cas et si la mémoire B_n du flux élémentaire pour le décodeur P-STD est examinée à l'instant spécifié par $td_n(j)$, les données pour l'unité d'accès peuvent ne pas être toutes présentes dans la mémoire B_n. Si cela se produit, la mémoire doit être réexaminée à intervalles de deux périodes de trame, jusqu'à ce que la mémoire contienne les données pour l'unité d'accès complète. A cet instant, toute l'unité d'accès et ses données systèmes éventuellement associées doivent être extraites de la mémoire B_n.

Il est admis qu'un sous-remplissage du tampon du vérificateur VBV se produise en permanence. Le décodeur P-STD doit supprimer du tampon B_n les données d'unité d'accès au plus tôt possible sans enfreindre la règle de l'alinéa précédent, ainsi que toutes valeurs de pointeur DTS ou PTS codées dans le flux binaire. Le décodeur peut ne pas être en

mesure de rétablir les temps corrects de décodage et d'affichage indiqués par les pointeurs DTS et PTS avant que l'état de sous-remplissage du tampon VBV disparaisse et qu'un autre pointeur PTS ou DTS soit trouvé dans le flux binaire.

2.5.2.4 Flux de paquets PES

Il est possible de construire un flux de données sous la forme d'une suite continue de paquets PES contenant chacun des données du même flux élémentaire et de même identificateur stream_id. Un tel flux est appelé flux PES. Le modèle de décodeur PES-STD pour un flux PES est identique au décodeur pour flux de programme (P-STD), sauf que la référence temporelle de flux élémentaire (ESCR) est utilisée à la place de la référence SCR et que ES_rate se substitue à program_mux_rate. Le démultiplexeur n'envoie les données qu'à une seule mémoire de flux élémentaire.

Les capacités de la mémoire BS_n sont définies comme suit pour le décodeur PES-STD:

– *pour les données vidéo de la Rec. UIT-T H.262 | ISO/CEI 13818-2:*

$$BS_n = VBV_{max}[\text{profil, niveau}] + BS_{oh}$$

BS_{oh} = (1/750) secondes × R_{max}[profil, niveau], avec VBV_{max}[profil, niveau] et R_{max}[profil, niveau] étant respectivement la capacité maximale de la mémoire VBV et le débit maximal de données vidéo selon le profil, le niveau et la couche définis respectivement dans les Tableaux 8-14 et 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2. La mémoire BS_{oh} est attribuée à l'en-tête de paquet PES;

– *pour les données vidéo de ISO/CEI 11172-2:*

$$BS_n = VBV_{max} + BS_{oh}$$

BS_{oh} = (1/750) secondes × R_{max}, où R_{max} et vbv_max se réfèrent respectivement au débit maximal et à la capacité maximale de la mémoire VBV pour un flux paramétrique binaire contraint de ISO/CEI 11172-2;

– *pour les données audio de ISO/CEI 11172-3 ou ISO/CEI 13818-3:*

$$BS_n = 2848 \text{ octets}$$

– *pour les données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10*:

$$BS_n = 1200 \times \text{MaxCPB}[\text{niveau}] + BS_{oh}$$

Où MaxCPB[niveau] est défini dans le Tableau A.1 (Limites de niveau) de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour chaque niveau.

2.5.2.5 Décodage et présentation

Dans le décodeur modèle du système pour le flux de programme, le décodage et la présentation suivent les mêmes définitions que pour le décodeur T-STD (voir 2.4.2.4 et 2.4.2.5).

2.5.2.6 Extensions du décodeur P-STD pour l'acheminement de données ISO/CEI 14496

Le modèle du décodeur P-STD est étendu pour le décodage des données ISO/CEI 14496 acheminées dans un flux de programme. Pour le décodage de flux élémentaires ISO/CEI 14496 distincts dans le décodeur P-STD, voir 2.11.2. Le 2.11.3 définit les extensions et les paramètres P-STD pour le décodage des scènes et des flux ISO/CEI 14496 associés.

2.5.2.7 Extensions du décodeur P-STD pour l'acheminement de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10

Pour le décodage dans le décodeur P-STD de flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 acheminés dans un flux de programme, voir 2.14.3.2.

2.5.3 Spécification de la syntaxe du flux de programme

La syntaxe suivante décrit un flux d'octets.

2.5.3.1 Flux de programme

Voir Tableau 2-37.

Tableau 2-37 – Flux de programme

Syntaxe	Nombre de bits	Mnémonique
MPEG2_program_stream() {		
do {		
pack()		
} while (nextbits() == pack_start_code)		
MPEG_program_end_code	**32**	**bslbf**
}		

2.5.3.2 Définition sémantique des champs dans un flux de programme

MPEG_program_end_code (code de fin de programme à codage MPEG) – Ce champ contient la chaîne binaire '0000 0000 0000 0000 0000 0001 1011 1001' (0x000001B9). Il termine le flux de programme.

2.5.3.3 Couche bloc de paquets dans un flux de programme

Voir Tableaux 2-38 et 2-39.

Tableau 2-38 – Bloc de paquets du flux de programme

Syntaxe	Nombre de bits	Mnémonique
pack() {		
pack_header()		
while (nextbits() == packet_start_code_prefix) {		
PES_packet()		
}		
}		

Tableau 2-39 – En-tête de bloc de paquets de flux de programme

Syntaxe	Nombre de bits	Mnémonique
pack_header() {		
pack_start_code	**32**	**bslbf**
'01'	**2**	**bslbf**
system_clock_reference_base [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_base [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_base [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_extension	**9**	**uimsbf**
marker_bit	**1**	**bslbf**
program_mux_rate	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
marker_bit	**1**	**bslbf**
reserved	**5**	**bslbf**
pack_stuffing_length	**3**	**uimsbf**
for (i = 0; i < pack_stuffing_length; i++) {		
stuffing_byte	**8**	**bslbf**
}		
if (nextbits() == system_header_start_code) {		
system_header ()		
}		
}		

2.5.3.4 Définition sémantique des champs contenus dans un bloc de paquets pour flux de programme

pack_start_code (code de début de bloc de paquets) – Ce champ contient la chaîne binaire '0000 0000 0000 0000 0000 0001 1011 1010' (0x000001BA). Il désigne le début d'un bloc de paquets.

system_clock_reference_base; system_clock_reference_extension (base de référence SCR; extension de référence SCR) – La référence temporelle du système (SCR) est un champ codé sur 42 éléments binaires en deux parties: l'une appelée base de référence temporelle du système, est codée sur 33 éléments binaires, et sa valeur est donnée par SCR_base(i) suivant l'équation 2-19; la deuxième partie, dite extension de référence temporelle du système est codée sur 9 éléments binaires, et sa valeur est indiquée par SCR_ext(i) suivant l'équation 2-20. La valeur codée dans le champ SCR indique l'instant d'arrivée prévu de l'octet contenant le dernier bit du champ SCR_base, à l'entrée du décodeur P-STD.

La fréquence des exigences de codage applicables au champ SCR est indiquée au 2.7.1.

marker_bit (bit marqueur) – Fanion codé sur 1 bit qui a la valeur '1'.

program_mux_rate (débit du multiplexeur de flux de programme) – Codé sur 22 éléments binaires, ce champ contient un entier qui spécifie le débit auquel le décodeur P-STD reçoit le bloc de paquets contenant le flux de programme. La valeur du champ program_mux_rate est mesurée en unités de 50 octets par seconde. La valeur 0 est interdite. La valeur du champ program_mux_rate sert à définir l'instant d'arrivée des octets à l'entrée du décodeur P-STD décrit au 2.5.2. La valeur codée dans le champ de débit du multiplexeur de programmes peut varier d'un bloc de paquets à un autre, dans un même flux de programme multiplexé et conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

pack_stuffing_length (longueur de bourrage d'un bloc de paquets) – Codé sur 3 éléments binaires, l'entier de ce champ spécifie le nombre d'octets de bourrage qui suivent ce champ.

stuffing_byte (octet de bourrage) – Codé sur 8 bits, ce champ a une valeur de '1111 1111' qui peut être insérée par le codeur afin, par exemple, de satisfaire aux exigences du canal. Ce champ est ignoré par le décodeur. Dans chaque en-tête de bloc de paquets, pas plus de 7 octets de bourrage doivent être présents.

2.5.3.5 En-tête de données systèmes

Voir Tableau 2-40.

Tableau 2-40 – En-tête système de flux de programme

Syntaxe	Nombre de bits	Mnémonique
system_header () {		
system_header_start_code	**32**	**bslbf**
header_length	**16**	**uimsbf**
marker_bit	**1**	**bslbf**
rate_bound	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
audio_bound	**6**	**uimsbf**
fixed_flag	**1**	**bslbf**
CSPS_flag	**1**	**bslbf**
system_audio_lock_flag	**1**	**bslbf**
system_video_lock_flag	**1**	**bslbf**
marker_bit	**1**	**bslbf**
video_bound	**5**	**uimsbf**
packet_rate_restriction_flag	**1**	**bslbf**
reserved_bits	**7**	**bslbf**
while (nextbits () == '1') {		
stream_id	**8**	**uimsbf**
'11'	**2**	**bslbf**
P-STD_buffer_bound_scale	**1**	**bslbf**
P-STD_buffer_size_bound	**13**	**uimsbf**
}		
}		

2.5.3.6 Définition sémantique des champs contenus dans l'en-tête système

system_header_start_code (code de début d'en-tête de données systèmes) – Ce champ contient la chaîne binaire '0000 0000 0000 0000 0000 0001 1011 1011' (0x000001BB). Il identifie le début d'un en-tête de données systèmes.

header_length (longueur d'en-tête) – Codé sur 16 éléments binaires, ce champ indique la longueur, en octets, de l'en-tête système suivant le champ header_length. On notera que de futures extensions de la présente Spécification pourront s'appliquer à cet en-tête.

rate_bound (limite de débit) – Codé sur 22 éléments binaires, ce champ est une valeur d'entier supérieure ou égale à la valeur maximale du champ program_mux_rate codé dans un bloc de paquets du flux de programme. Ce champ peut être utilisé par un décodeur pour déterminer s'il est en mesure de décoder l'ensemble du flux.

audio_bound (limite de flux audio) – Codé sur 6 éléments binaires, ce champ indique une valeur entière comprise entre 0 et 32 et qui est fixée à une valeur supérieure ou égale au nombre maximal de flux audio ISO/CEI 13818-3 et ISO/CEI 11172-3 multiplexés dans le flux de programme, pour lesquels les processus de décodage sont simultanément actifs. Dans le cadre du présent paragraphe, le processus de décodage d'un flux audio selon ISO/CEI 13818-3 ou ISO/CEI 11172-3 est actif si la mémoire tampon du décodeur STD n'est pas vide ou si l'unité de présentation est en cours de présentation dans le modèle P-STD.

fixed_flag (fanion de débit constant) – Lorsque la valeur de ce fanion est mise à '1', elle indique que le débit est constant. Si elle est mise à '0', elle indique que le débit est variable. En cas de fonctionnement à débit fixe, la valeur codée dans tous les champs de référence temporelle de système contenus dans le flux multiplexé selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 doit vérifier la relation linéaire suivante:

$$SCR_base(i) = ((c1 \times i + c2)\ DIV\ 300)\ \%\ 2^{33} \tag{2-22}$$

$$SCR_ext(i) = ((c1 \times i + c2)\ DIV\ 300)\ \%\ 300 \tag{2-23}$$

où:

c1 constante réelle valable pour tout i

c2 constante réelle valable pour tout i

i indice, dans le flux multiplexé selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, de l'octet contenant le dernier bit d'un champ quelconque de référence temporelle du système.

CSPS_flag (fanion de flux paramétrique contraint du système) – Si la valeur de ce fanion est mise à '1', le flux de programme répond aux contraintes définies au 2.7.9.

system_audio_lock_flag (fanion de verrouillage du rapport entre fréquence d'échantillonnage audio et fréquence système) – Ce fanion codé sur 1 bit indique s'il existe une relation spécifiée de rapport constant entre la fréquence des échantillons audio et la fréquence d'horloge système dans le décodeur modèle du système. Le paragraphe 2.5.2.1 définit la fréquence d'horloge de base du système et l'ISO/CEI 13818-3 spécifie la fréquence des échantillons audio. Le fanion de verrouillage du rapport entre fréquence d'échantillonnage audio et fréquence système ne peut être mis qu'à la valeur '1' si, pour toutes les unités de présentation de tous les flux élémentaires de données audio du flux de programme, le rapport entre la fréquence système et la fréquence réelle d'échantillonnage audio, SCASR, est constant et égal à la valeur indiquée dans le tableau suivant, à la fréquence nominale d'échantillonnage qui est indiquée dans le flux de données audio.

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_P\text{-}STD} \tag{2-24}$$

La notation $\frac{X}{Y}$ indique une division de nombres réels.

Fréquence nominale d'échantillonnage audio (kHz)	16	32	22,05	44,1	24	48
SCASR	27 000 000 / 16 000	27 000 000 / 32 000	27 000 000 / 22 050	27 000 000 / 44 100	27 000 000 / 24 000	27 000 000 / 48 000

system_video_lock_flag (fanion de verrouillage du rapport entre base de temps vidéo et fréquence système) – Ce fanion codé sur 1 bit indique s'il existe une relation spécifiée de rapport constant entre la base de temps vidéo et la fréquence d'horloge système dans le décodeur modèle du système. Ce fanion ne peut être mis à la valeur '1' que si, pour toutes les unités de présentation de tous les flux élémentaires de données vidéo d'un flux de programme Rec. UIT-T H.222.0 | ISO/CEI 13818-1, le rapport entre la fréquence d'horloge système et la fréquence de la base de temps vidéo effective est constant.

Dans le cas de flux vidéo ISO/CEI 11172-2 et Rec. UIT-T H.262 | ISO/CEI 13818-2, si le fanion system_video_lock_flag est mis à '1', le rapport entre la fréquence d'horloge système et la fréquence d'image réelle, SCFR, sera constant et égal à la valeur indiquée dans le tableau suivant pour la fréquence d'image nominale indiquée dans le flux vidéo.

Dans le cas de flux vidéo ISO/CEI 14496-2, si le fanion system_video_lock_flag est mis à '1', la base de temps du flux vidéo ISO/CEI 14496-2, définie par la résolution vop_time_increment_resolution, sera verrouillée sur l'horloge STC et sera exactement égale à N fois la fréquence d'horloge système divisée par K, N et K étant des entiers qui ont une valeur fixe dans chaque séquence d'objet visuel, et K étant supérieur ou égal à N.

Dans le cas des flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10, la fréquence de la base de temps AVC est définie par l'échelle de temps du paramètre AVC. Si le fanion system_video_lock_flag est mis à '1' pour un flux vidéo AVC, la fréquence de la base de temps AVC sera verrouillée sur l'horloge STC et sera exactement égale à N fois la fréquence d'horloge système divisée par K, N et K étant des entiers qui ont une valeur fixe dans chaque séquence vidéo AVC, et K étant supérieur ou égal à N.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_P\text{-}STD} \tag{2-25}$$

Fréquence nominale de trame (Hz)	23,976	24	25	29,97	30	50	59,94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

Les valeurs du rapport SCFR sont exactes. La fréquence de trame réelle peut différer légèrement de la fréquence nominale, si celle-ci est de 23,976, 29,97 ou de 59,94 trames par seconde.

video_bound (limite de flux vidéo) – Entier de 5 bits de valeur comprise entre 0 et 16 et mis à une valeur supérieure ou égale au nombre maximal de flux vidéo dans le flux de programme, pour lesquels les processus de décodage sont simultanément actifs. Dans le cadre de ce paragraphe, le processus de décodage d'un flux vidéo est actif si l'un des tampons du décodeur P-STD n'est pas vide ou si une unité de présentation est en cours de présentation dans le décodeur P-STD.

packet_rate_restriction_flag (fanion de restriction de débit de paquet) – Le fanion de restriction de débit de paquet est codé sur un bit. Si le fanion CSPS est mis à la valeur '1', le fanion de restriction de débit de paquet indique la contrainte applicable au débit de paquet, suivant les indications du 2.7.9. Si le fanion CSPS est mis à la valeur '0', le fanion de restriction de débit de paquet n'a pas de signification définie.

reserved_bits (bits réservés) – Ce champ de 7 bits est réservé pour usage futur par l'ISO/CEI. Sauf spécification contraire par l'UIT-T | ISO/CEI, sa valeur sera '111 1111'.

stream_id (identificateur de flux) – Codé sur 8 éléments binaires, ce champ indique le codage et le numéro de flux élémentaire du flux auquel se rapportent les champs suivants P-STD_buffer_bound_scale et P-STD_buffer_size_bound.

Si ce champ a une valeur égale à '1011 1000', les champs P-STD_buffer_bound_scale et P-STD_buffer_size_bound, qui suivent le champ stream_id, se rapportent à tous les flux de données audio contenus dans le flux de programme.

Si ce champ a une valeur égale à '1011 1001', les champs P-STD_buffer_bound_scale et P-STD_buffer_size_bound, qui suivent le champ stream_id, se rapportent à tous les flux de données vidéo contenus dans le flux de programme.

Si ce champ prend une quelconque autre valeur que les deux précédentes, cette valeur d'octet sera supérieure ou égale à '1011 1100' et devra être interprétée comme se rapportant au codage de flux et au numéro de flux de programme indiqués dans le Tableau 2-22.

Chaque flux élémentaire présent dans le flux de programme doit avoir ses champs P-STD_buffer_bound_scale et P-STD_buffer_size_bound spécifiés exactement une seule fois par ce mécanisme dans chaque en-tête de données systèmes.

P-STD_buffer_bound_scale (échelle de la mémoire limite du décodeur P-STD) – Ce fanion indique le facteur de normalisation utilisé pour interpréter le champ P-STD_buffer_size_bound suivant. Si le précédent identificateur de flux a indiqué qu'il s'agissait d'un flux audio, le champ P-STD_buffer-bound_scale doit avoir la valeur '0'. Si le précédent identificateur de flux a indiqué qu'il s'agissait d'un flux vidéo, le champ P-STD_buffer_bound_scale doit avoir la valeur '1'. Pour tous les autres types de flux, la valeur de ce champ peut être '1' ou '0'.

P-STD_buffer_size_bound (limite de capacité mémoire du décodeur P-STD) – Codé sur 13 éléments binaires, ce champ contient un entier non signé dont la valeur est supérieure ou égale à la capacité maximale de la mémoire tampon d'entrée du décodeur P-STD, BS_n, pour tous les paquets du flux n contenus dans le flux de programme. Si le champ P-STD_buffer_bound_scale a la valeur '0', le champ P-STD_buffer_size_bound mesure la limite de capacité mémoire

en unités de 128 octets. Si le champ P-STD_buffer_bound_scale a la valeur '1', le champ P-STD_buffer_size_bound mesure la limite de capacité mémoire en unités de 1024 octets. On a donc:

$$si(P-STD_buffer_bound_scale == 0)$$
$$BS_n \leq P-STD_buffer_size_bound \times 128$$

sinon:

$$BS_n \leq P-STD_buffer_size_bound \times 1024$$

2.5.3.7 Couche paquet du flux de programme

La couche paquet du flux de programme est définie par 2.4.3.6 concernant la couche paquet du flux PES.

2.5.4 Mappage du flux de programme

Le mappage du flux de programme (PSM, *program stream map*) donne une description des flux élémentaires contenus dans le flux de transport, avec leurs relations réciproques. Cette structure, lorsqu'elle est acheminée dans un flux de transport, ne doit pas être modifiée. Le mappage de flux de programme (PSM) est présent sous forme de paquet PES lorsque la valeur d'identification du flux est 0xBC (voir Tableau 2-22).

NOTE – Cette syntaxe diffère de la syntaxe des paquets PES présentée au 2.4.3.6.

On trouvera au 2.6 une définition des champs descripteurs correspondants.

2.5.4.1 Syntaxe de la structure program_stream_map

Voir Tableau 2-41.

Tableau 2-41 – Mappage du flux de programme

Syntaxe	Nombre de bits	Mnémonique
program_stream_map() {		
packet_start_code_prefix	**24**	**bslbf**
map_stream_id	**8**	**uimsbf**
program_stream_map_length	**16**	**uimsbf**
current_next_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
program_stream_map_version	**5**	**uimsbf**
reserved	**7**	**bslbf**
marker_bit	**1**	**bslbf**
program_stream_info_length	**16**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
elementary_stream_map_length	**16**	**uimsbf**
for (i = 0; i < N1; i++) {		
stream_type	**8**	**uimsbf**
elementary_stream_id	**8**	**uimsbf**
elementary_stream_info_length	**16**	**uimsbf**
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	**32**	**rpchof**
}		

2.5.4.2 Définition sémantique des champs contenus dans la structure de mappage de flux de programme

packet_start_code_prefix (préfixe codé de début de paquet) – Ce champ, codé sur 24 éléments binaires, constitue, de concert avec la structure map_stream_id qui le suit, un code de début de paquet qui indique le point de départ d'un paquet. Ce préfixe est la chaîne binaire '0000 0000 0000 0000 0000 0001' (0x000001 en hexadécimal).

map_stream_id (identificateur du flux à mapper) – Codée sur 8 éléments binaires, la valeur de ce champ est toujours 0xBC.

program_stream_map_length (longueur du champ program_stream_map) – Codé sur 16 éléments binaires, ce champ indique le nombre total d'octets contenus dans la structure program_stream_map qui suit immédiatement ce champ. La valeur maximale de ce champ est 1018 octets (0x3FA).

current_next_indicator (indicateur d'applicabilité de la structure suivante) – Lorsqu'il est mis à '1', ce fanion codé sur 1 bit indique que la structure de mappage du flux de programme envoyée est actuellement applicable. Lorsqu'il est mis à '0', il indique que la structure envoyée n'est pas encore applicable et doit être la prochaine à devenir valide.

program_stream_map_version (version du mappage de flux de programme) – Codé sur 5 éléments binaires, ce champ indique le numéro de version du mappage complet du flux de programme. Ce numéro de version doit être augmenté de 1 modulo 32 chaque fois que la définition du mappage de flux de programme change. Lorsque le champ current_next_indicator est mis à '1', la version de mappage de flux de programme doit être celle qui est actuellement applicable. Lorsqu'il est mis à '0', cette version est la prochaine à appliquer.

program_stream_info_length (longueur des descripteurs de flux de programme) – Codé sur 16 éléments binaires, ce champ spécifie le nombre d'octets dont se composent les descripteurs suivant immédiatement ce champ.

marker_bit (bit marqueur) – Fanion codé sur 1 bit qui a la valeur '1'.

elementary_stream_map_length (longueur des informations de mappage de flux élémentaire) – Codé sur 16 éléments binaires, ce champ spécifie la longueur totale (en octets) de toutes les informations de flux élémentaire contenues dans le mappage du flux de programme. Il comprend les champs stream_type, elementary_stream_id et elementary_stream_info_length.

stream_type (type de flux) – Codé sur 8 éléments binaires, ce champ spécifie le type de flux élémentaire conformément au Tableau 2-34. Le champ de type de flux doit identifier uniquement les flux élémentaires contenus dans les paquets PES. La valeur 0x05 est interdite.

elementary_stream_id (identificateur de flux élémentaire) – Codé sur 8 éléments binaires, ce champ spécifie la valeur de l'identificateur de flux élémentaire contenu dans les en-têtes des paquets PES acheminant ce flux élémentaire.

elementary_stream_info_length (longueur des descripteurs de flux élémentaire) – Codé sur 16 éléments binaires, ce champ spécifie le nombre d'octets dont se composent les descripteurs suivant immédiatement ce champ.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 éléments binaires, ce champ contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe A après traitement de toute la structure de mappage de programme pour le flux de transport.

2.5.5 Répertoire du flux de programme

Le répertoire d'un flux de programme complet se compose de toutes les données de répertoire acheminées par les paquets PSD identifiés par le champ directory_stream_id. La syntaxe pour des paquets PSD contenant le champ program_stream_directory est définie au Tableau 2-42.

NOTE 1 – Cette syntaxe diffère de la syntaxe des paquets PES présentée au 2.4.3.6.

Des entrées de répertoire peuvent être requises afin de repérer des images à codage I dans un flux de données vidéo tel que défini dans la Rec. UIT-T H.262 | ISO/CEI 13818-2 et dans ISO/CEI 11172-2. Si une image à codage I est signalée par une entrée de répertoire et est précédée d'un en-tête de séquence sans en-têtes d'image intermédiaires, cette entrée de répertoire doit renvoyer au premier octet de l'en-tête de séquence. Si une image à codage I est signalée par une entrée de répertoire et est précédée d'un en-tête de groupe d'images sans en-têtes d'image intermédiaires et sans en-tête de séquence immédiatement précédent, cette entrée de répertoire doit renvoyer au premier octet de l'en-tête de groupe d'images. Toute autre image indiquée par une entrée de répertoire doit être signalée par le premier octet de l'en-tête d'image.

NOTE 2 – Il est recommandé que les images I qui suivent immédiatement un en-tête de séquence soient référencées dans des structures de répertoire de manière que celui-ci contienne une entrée à chaque point où le décodeur peut être complètement réinitialisé.

Des entrées d'annuaire seront éventuellement nécessaires pour faire référence à une ou plusieurs images IDR associées à un message SEI de point de rétablissement dans un flux vidéo AVC. Chacune de ces entrées de répertoire désignera le premier octet d'une unité d'accès AVC.

Les références de répertoire à des flux audio conformes à ISO/CEI 13818-3 et ISO/CEI 11172-3 doivent être les mots de synchronisation des trames audio.

NOTE 3 – Il est recommandé que la durée séparant les unités d'accès référencées ne dépasse pas une demi-seconde.

Les unités d'accès doivent être énumérées dans le paquet PSD dans l'ordre de leur apparition dans le flux binaire.

2.5.5.1 Syntaxe du paquet de flux PES pour le répertoire de flux de programme

Voir Tableau 2-42.

Tableau 2-42 – Syntaxe des paquets de flux PES pour un répertoire de flux de programme

Syntaxe	Nombre de bits	Mnémonique
directory_PES_packet(){		
packet_start_code_prefix	**24**	**bslbf**
directory_stream_id	**8**	**uimsbf**
PES_packet_length	**16**	**uimsbf**
number_of_access_units	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[44..30]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[44..30]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
for (i = 0; i < number_of_access_units; i++) {		
packet_stream_id	**8**	**uimsbf**
PES_header_position_offset_sign	**1**	**tcimsbf**
PES_header_position_offset[43..30]	**14**	**uimsbf**
marker_bit	**1**	**bslbf**
PES_header_position_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
PES_header_position_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
reference_offset	**16**	**uimsbf**
marker_bit	**1**	**bslbf**
reserved	**3**	**bslbf**
PTS[32..30]	**3**	**uimsbf**
marker_bit	**1**	**bslbf**
PTS[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
PTS[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
bytes_to_read[22..8]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
bytes_to_read[7..0]	**8**	**uimsbf**
marker_bit	**1**	**bslbf**
intra_coded_indicator	**1**	**bslbf**
coding_parameters_indicator	**2**	**bslbf**
reserved	**4**	**bslbf**
}		
}		

2.5.5.2 Définition sémantique des champs contenus dans un répertoire de flux de programme

packet_start_code_prefix (préfixe codé de début de paquet) – Ce champ, codé sur 24 éléments binaires, constitue, de concert avec la structure stream_id qui le suit, un code de début de paquet qui indique le point de départ d'un paquet. Ce préfixe est la chaîne binaire '0000 0000 0000 0000 0000 0001' (0x000001 en hexadécimal).

directory_stream_id – Champ codé sur 8 bits dont la valeur est '1111 1111' (0xFF).

PES_packet_length (longueur de paquet PES) – Ce champ, codé sur 16 éléments binaires, spécifie le nombre total d'octets contenus dans le paquet PES qui suit le dernier octet de ce champ (voir Tableau 2-22).

number_of_access_units (nombre d'unités d'accès) – Codé sur 15 éléments binaires, ce champ donne le nombre d'unités d'accès qui sont référencées dans un paquet PES de répertoire.

prev_directory_offset (adresse du précédent paquet de répertoire) – Codé sur 45 éléments binaires, ce champ contient un entier non signé qui désigne le décalage du registre d'adresse correspondant au premier octet du code de début du précédent paquet de flux PSD contenant des informations sur le répertoire du flux de programme. Ce décalage d'adresse se rapporte au premier octet du code de début du paquet PSD qui contient ce champ (prev_directory_offset). La valeur '0' est réservée pour indiquer qu'il n'y a pas encore d'informations de répertoire sur le flux de programme.

next_directory_offset (adresse du prochain paquet de répertoire) – Codé sur 45 éléments binaires, ce champ contient un entier non signé qui désigne le décalage du registre d'adresse correspondant au premier octet du code de début du prochain paquet de flux PSD contenant des informations sur le répertoire du flux de programme. Ce décalage d'adresse se rapporte au premier octet du code de début du paquet PSD qui contient ce champ (next_directory_offset). La valeur '0' est réservée pour indiquer qu'il n'y a plus d'informations de répertoire sur le flux de programme.

packet_stream_id (identificateur du flux en paquets) – Codé sur 8 éléments binaires, ce champ est l'identificateur du flux élémentaire qui contient l'unité d'accès indiquée par une entrée de répertoire.

PES_header_position_offset_sign (signe du décalage de position du premier octet d'un paquet PES) – Ce fanion codé sur 1 bit donne le signe arithmétique (positif pour la valeur '0' et négatif pour la valeur '1') du champ PES_header_position_offset dont la description suit immédiatement**.**

PES_header_position_offset (décalage de position du premier octet d'un paquet PES) – Codé sur 44 éléments binaires, ce champ contient un entier non signé qui indique le registre d'adresse correspondant au premier octet du paquet PES contenant l'unité d'accès référencée. Ce décalage d'adresse se rapporte au premier octet du code de début du paquet PES qui contient ce champ. La valeur '0' est réservée pour indique qu'aucune unité d'accès n'est référencée.

reference_offset (décalage de référence) – Codé sur 16 éléments binaires, ce champ contient un entier non signé qui indique la position du premier octet de l'unité d'accès référencée, mesurée en octets par rapport au premier octet du paquet de flux PES contenant le premier octet de l'unité d'accès référencée.

PTS (presentation_time_stamp) (horodateur PTS) – Codé sur 33 éléments binaires, ce champ est l'horodateur de l'unité d'accès référencée. La définition sémantique du codage de ce champ est décrite au 2.4.3.6.

bytes_to_read (octets à lire) – Codé sur 23 éléments binaires, ce champ contient un entier non signé indiquant le nombre d'octets qui viennent à la suite de l'octet indiqué par le champ reference_offset et qui sont nécessaires pour décoder complètement l'unité d'accès. Ce nombre comprend tous les octets éventuellement multiplexés dans la couche système, y compris ceux qui contiennent des informations issues d'autres flux.

intra_coded_indicator (indicateur de codage non prédictif) – Ce fanion codé sur 1 bit, s'il est mis à '1', indique que l'unité d'accès référencée n'est pas codée en mode prédictif. Cette indication est indépendante des autres paramètres de codage qui pourraient être nécessaires afin de décoder l'unité d'accès. Par exemple, pour les images bitrames à codage intratrame, ce fanion sera mis à la valeur '1', tandis que pour les images à codage 'P' ou 'B', il sera mis à la valeur '0'. La valeur de ce champ est réservée pour tous les paquets PES contenant des données ne provenant pas d'un flux de données vidéo conforme à la Rec. UIT-T H.262 | ISO/CEI 13818-2 (voir Tableau 2-43).

Tableau 2-43 – Indicateur de codage non prédictif

Valeur	Signification
0	Codage prédictif
1	Codage non prédictif

coding_parameters_indicator (indicateur de paramètres de codage) – Codé sur 2 éléments binaires, ce champ sert à indiquer l'emplacement des paramètres de codage qui sont nécessaires pour décoder les unités d'accès référencées. Ce champ pourra par exemple servir à déterminer l'emplacement de matrices de quantification pour images vidéo.

Tableau 2-44 – Indicateur de paramètres de codage

Valeur	Signification
00	Tous les paramètres de codage sont réglés à leurs valeurs par défaut
01	Tous les paramètres de codage sont réglés dans cette unité d'accès avec au moins l'un d'entre eux non réglé sur une valeur par défaut
10	Certains paramètres de codage sont réglés dans cette unité d'accès
11	Aucun paramètre de codage n'est codé dans cette unité d'accès

2.6 Descripteurs de programme et d'élément de programme

Les descripteurs de programme et d'élément de programme sont des structures qui peuvent être utilisées afin d'élargir les définitions correspondantes. Tous les descripteurs de flux ont un format qui commence par une valeur d'étiquetage codée sur 8 bits. Cette valeur d'étiquetage est suivie par un champ de 8 bits donnant la longueur du descripteur, puis des champs de données.

2.6.1 Définition sémantique des champs des descripteurs de programme et d'élément de programme

Les définitions sémantiques suivantes s'appliquent aux descripteurs définis dans 2.6.2 à 2.6.34.

descriptor_tag (étiquette de descripteur) – Codé sur 1 octet, ce champ identifie chaque descripteur.

Le Tableau 2-45 indique les valeurs d'étiquetage des descripteurs conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, réservées selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 et mises à la disposition des utilisateurs. Une lettre "X" dans la colonne de flux de transport ou de programme indique que le descripteur correspondant est applicable au flux indiqué, selon le cas. On notera que la signification des champs contenus dans un descripteur peut dépendre du flux auquel celui-ci s'applique. Cela est spécifié dans les définitions sémantiques de descripteur reproduites ci-après.

descriptor_length (longueur de descripteur) – Codé sur 1 octet, ce champ spécifie le nombre d'octets du descripteur qui le suit immédiatement.

Tableau 2-45 – Descripteurs de programme et d'élément de programme

descriptor_tag	TS	PS	Identification
0	n/a	n/a	Réservé
1	n/a	n/a	Réservé
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	system_clock_descriptor
12	X	X	multiplex_buffer_utilization_descriptor
13	X	X	copyright_descriptor
14	X		maximum_bitrate_descriptor
15	X	X	private_data_indicator_descriptor
16	X	X	smoothing_buffer_descriptor
17	X		STD_descriptor
18	X	X	IBP_descriptor
19-26	X		Défini dans l'ISO/CEI 13818-6
27	X	X	MPEG-4_video_descriptor

Tableau 2-45 – Descripteurs de programme et d'élément de programme

descriptor_tag	TS	PS	Identification
28	X	X	MPEG-4_audio_descriptor
29	X	X	IOD_descriptor
30	X		SL_descriptor
31	X	X	FMC_descriptor
32	X	X	external_ES_ID_descriptor
33	X	X	MuxCode_descriptor
34	X	X	FmxBufferSize_descriptor
35	X		multiplexBuffer_descriptor
36	X	X	content_labeling_descriptor
37	X	X	metadata_pointer_descriptor
38	X	X	metadata_descriptor
39	X	X	metadata_STD_descriptor
40	X	X	AVC video descriptor
41	X	X	IPMP_descriptor (défini dans ISO/CEI 13818-11, MPEG-2 IPMP)
42	X	X	AVC timing and HRD descriptor
43	X	X	MPEG-2_AAC_audio_descriptor
44	X	X	FlexMuxTiming_descriptor
45-63	n/a	n/a	Réservés à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1
64-255	n/a	n/a	Réservés à l'utilisateur privé

n/a: sans objet

2.6.2 Descripteur de flux vidéo

Le descripteur de flux vidéo donne des informations de base qui identifient les paramètres de codage d'un flux élémentaire de données vidéo, tel que décrit dans la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou dans ISO/CEI 11172-2 (voir Tableau 2-46).

Tableau 2-46 – Descripteur de flux vidéo

Syntaxe	Nombre de bits	Mnémonique
video_stream_descriptor(){		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
multiple_frame_rate_flag	**1**	**bslbf**
frame_rate_code	**4**	**uimsbf**
MPEG_1_only_flag	**1**	**bslbf**
constrained_parameter_flag	**1**	**bslbf**
still_picture_flag	**1**	**bslbf**
if (MPEG_1_only_flag == '0'){		
profile_and_level_indication	**8**	**uimsbf**
chroma_format	**2**	**uimsbf**
frame_rate_extension_flag	**1**	**bslbf**
reserved	**5**	**bslbf**
}		
}		

2.6.3 Définition sémantique des champs dans un descripteur de flux vidéo

multiple_frame_rate_flag (fanion de fréquences de trame multiples) – S'il est mis à '1', ce fanion codé sur 1 bit indique que plusieurs fréquences de trame peuvent cohabiter dans le flux de données vidéo. S'il est mis à '0', une seule fréquence de trame est présente.

frame_rate_code (code de fréquence de trame) – Codé sur 4 éléments binaires, ce champ est conforme à la définition donnée au 6.3.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, sauf que si ce champ est mis à la valeur '1', l'indication d'une fréquence de trame particulière permet également la présence d'autres fréquences de trame dans le flux vidéo, comme spécifié dans le Tableau 2-47.

Tableau 2-47 – Code de fréquence d'image

Fréquence codée	Fréquences comprises dans le codage
23,976	
24,0	23,976
25,0	
29,97	23,976
30,0	23,976 24,0 29,97
50,0	25,0
59,94	23,976 29,97
60,0	23,976 24,0 29,97 30,0 59,94

MPEG_1_only_flag (fanion de format MPEG_1_uniquement) – S'il est mis à '1', ce fanion codé sur 1 bit indique que le flux vidéo contient uniquement des données conformes à ISO/CEI 11172-2. S'il est mis à '0', le flux vidéo peut aussi bien contenir des données conformes à la Rec. UIT-T H.262 | ISO/CEI 13818-2 que des données de flux paramétriques contraints conformes à l'ISO/CEI 11172-2.

constrained_parameter_flag (fanion de flux paramétrique contraint) – Lorsqu'il est mis à la valeur '1', ce fanion codé sur 1 bit indique que le flux vidéo ne doit pas contenir de données vidéo de flux non contraint conformes à l'ISO/CEI 11172-2. S'il est mis à '0' le flux vidéo peut contenir aussi bien des données vidéo de flux paramétrique contraint que des données de flux non contraint conformes à l'ISO/CEI 11172-2. Lorsque le fanion MPEG_1_only_flag est mis à '0' le fanion constrained_parameter_flag doit être mis à la valeur '1'.

still_picture_flag (fanion d'image figée) – S'il est mis à '1', ce fanion codé sur 1 bit indique que le flux vidéo ne contient que des images figées. S'il est mis à '0', le flux vidéo peut contenir des données d'images cinématographiques ou figées.

profile_and_level_indication (indication de profil et de niveau) – Champ de 8 bits, codé comme les champs profile_and_level_indication du flux vidéo de la Rec. UIT-T H.262 | ISO/CEI 13818-2. La valeur de ce champ indique un profil et un niveau égal ou supérieur à tout profil ou niveau de l'une quelconque des séquences du flux vidéo associé. Aux fins du présent paragraphe, un flux de paramètre contraint ISO/CEI 11172-2 est considéré comme étant un flux de profil principal à faible niveau (MP @ LL).

chroma_format (format de chrominance) – Ce champ de 2 bits est codé de la même façon que les champs chroma_format du flux vidéo de la Rec. UIT-T H.262 | ISO/CEI 13818-2. La valeur de ce champ doit être égale ou supérieure à la valeur du champ chroma_format de toute séquence vidéo du flux vidéo associé. Pour les besoins de ce paragraphe, un flux vidéo ISO/CEI 11172-2 aura un champ chroma_format de valeur '01', indiquant 4:2:0.

frame_rate_extension_flag (fanion d'extension de fréquence de trame) – S'il est mis à '1', ce fanion codé sur 1 bit indique que l'un des champs frame_rate_extension_n ou frame_rate_extension_d, ou les deux, contenus dans un flux vidéo selon la Rec. UIT-T H.262 | ISO/CEI 13818-2, ont une valeur différente de zéro. Pour les besoins du présent paragraphe, un flux vidéo ISO/CEI 11172-2 est contraint d'avoir les 2 champs mis à zéro.

2.6.4 Descripteur de flux audio

Le descripteur de flux de données audio donne des informations de base qui permettent d'identifier la version de codage d'un flux élémentaire de données audio tel que décrit dans l'ISO/CEI 13818-3 ou dans l'ISO/CEI 11172-3 (voir Tableau 2-48).

Tableau 2-48 – Descripteur de flux de données audio

Syntaxe	Nombre de bits	Mnémonique
audio_stream_descriptor(){		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
free_format_flag	**1**	**bslbf**
ID	**1**	**bslbf**
layer	**2**	**bslbf**
variable_rate_audio_indicator	**1**	**bslbf**
reserved	**3**	**bslbf**
}		

2.6.5 Définition sémantique des champs dans un descripteur de flux audio

free_format_flag (fanion de qualification du débit audio) – S'il est mis à '1', ce fanion codé sur 1 bit indique que les données du flux audio ont un indice de débit mis à '0000'. S'il est mis à '0', l'indice de débit est différent de '0000' (voir 2.4.2.3 de l'ISO/CEI 13818-3) dans toutes les trames audio du flux.

ID (identificateur) – Lorsque ce champ, codé sur un bit, est mis à '1', les champs ID de toutes les trames audio du flux audio sont mis à '1' (voir 2.4.2.3 de l'ISO/CEI 13818-3).

layer (couche) – Ce champ de 2 bits est codé de la même façon que le champ layer (couche) des flux audio de l'ISO/CEI 13818-3 ou ISO/CEI 11172-3 (voir 2.4.2.3 de l'ISO/CEI 13818-3). La couche indiquée dans ce champ doit être supérieure ou égale à la couche la plus haute spécifiée dans les trames audio du flux audio.

variable_rate_audio_indicator – Ce fanion sur 1 bit, lorsqu'il prend la valeur '0', indique que la valeur codée du champ débit ne change pas dans des trames audio consécutives qui doivent être présentées sans discontinuité.

2.6.6 Descripteur de hiérarchie

Le descripteur de hiérarchie donne des informations permettant d'identifier les éléments de programme qui contiennent des composantes de flux vidéo, audio ou privés à codage hiérarchique (voir Tableau 2-49).

Tableau 2-49 – Descripteur de hiérarchie

Syntaxe	Nombre de bits	Mnémonique
hierarchy_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**4**	**bslbf**
hierarchy_type	**4**	**uimsbf**
reserved	**2**	**bslbf**
hierarchy_layer_index	**6**	**uimsbf**
reserved	**2**	**bslbf**
hierarchy_embedded_layer_index	**6**	**uimsbf**
reserved	**2**	**bslbf**
hierarchy_channel	**6**	**uimsbf**
}		

2.6.7 Définition sémantique des champs dans un descripteur de hiérarchie

hierarchy_type (type de hiérarchie) – La relation hiérarchique entre la couche hiérarchique associée et sa couche hiérarchique encastrée est définie par le Tableau 2-50.

hierarchy_layer_index (indice de couche hiérarchique) – Codé sur 6 éléments binaires, ce champ définit un indice unique renvoyant, dans une table de couches à codage hiérarchique, à l'élément de programme associé. Ces indices doivent être uniques à l'intérieur d'une même définition de programme.

hierarchy_embedded_layer_index (indice de couche hiérarchique incorporée) – Codé sur 6 éléments binaires, ce champ définit l'indice de la table de couches hiérarchiques de l'élément de programme qu'il faut consulter avant de décoder le flux élémentaire associé à ce descripteur de hiérarchie. La valeur de ce champ n'est pas définie si le champ hierarchy_type a la valeur 15 (couche de base).

hierarchy_channel (canal hiérarchique) – Codé sur 6 éléments binaires, ce champ indique le numéro de canal destiné à l'élément de programme associé, dans une série ordonnée de canaux de transmission. Le canal de transmission le plus robuste est défini par la plus faible valeur de codage de ce champ, compte tenu de la définition globale de la hiérarchie des canaux de transmission.

NOTE – Un même canal hiérarchique peut être assigné simultanément à plusieurs éléments de programme.

Tableau 2-50 – Valeurs du descripteur de hiérarchie

Valeur	Description
0	Valeur réservée
1	échelonnabilité spatiale
2	échelonnabilité SNR
3	échelonnabilité temporelle
4	subdivision des données
5	flux binaire d'extension
6	flux privé
7	profil multiformat
8-14	Valeur réservée
15	Couche de base

2.6.8 Descripteur d'enregistrement

Le descripteur d'enregistrement donne une méthode permettant d'identifier de manière univoque les formats de données privées (voir Tableau 2-51).

Tableau 2-51 – Descripteur d'enregistrement

Syntaxe	Nombre de bits	Mnémonique
registration_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
format_identifier	**32**	**uimsbf**
for (i = 0; i < N; i++){		
additional_identification_info	**8**	**bslbf**
}		
}		

2.6.9 Définition sémantique des champs dans un descripteur d'enregistrement

format_identifier (identificateur de format) – Codé sur 32 éléments binaires, ce champ contient une valeur issue d'un organisme d'enregistrement désigné par l'ISO/CEI JTC 1/SC 29.

additional_identification_info (informations supplémentaires d'identification) – Les octets de ce champ, s'il est présent, sont définis par le champ format_identifier et, une fois définis, ne doivent pas changer.

2.6.10 Descripteur d'alignement de flux de données

Le descripteur d'alignement de flux de données définit le type d'alignement qui est présent dans le flux élémentaire associé. Si l'indicateur d'alignement des données est mis à la valeur '1' dans l'en-tête de paquet PES, et si le descripteur en question est présent, il est obligatoire de respecter l'alignement ainsi spécifié (voir Tableau 2-52).

Tableau 2-52 – Descripteur d'alignement du flux de données

Syntaxe	Nombre de bits	Mnémonique
data_stream_alignment_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
alignment_type	**8**	**uimsbf**
}		

2.6.11 Définition sémantique des champs dans un descripteur d'alignement de flux de données

alignment_type (type d'alignement) – Le Tableau 2-53 décrit le type d'alignement de données vidéo ISO/CEI 11172-2 ou Rec. UIT-T H.262 | ISO/CEI 13818-2, ou celui de flux visuels ISO/CEI 14496-2 lorsque l'indicateur d'alignement de données de l'en-tête de paquet PES a la valeur '1'. Pour ces flux vidéo, le premier octet de données de paquet PES qui suit l'en-tête PES est le premier octet d'un code de déclenchement du type indiqué au Tableau 2-53. Au début d'une séquence vidéo, l'alignement doit être observé au niveau du code de déclenchement du premier en-tête de séquence.

NOTE – La spécification du type d'alignement '01' du Tableau 2-53 n'exclut pas de commencer l'alignement au niveau d'un en-tête GOP ou SEQ.

L'unité d'accès est définie au 2.1.1.

Tableau 2-53 – Valeurs d'alignement du flux de données vidéo

Type d'alignement	Description
00	Valeur réservée
01	Tranche ou unité d'accès vidéo
02	Unité d'accès vidéo
03	GOP ou SEQ
04	SEQ
05-FF	Valeur réservée

Le Tableau 2-54 décrit les types d'alignement pour les données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 lorsque l'indicateur d'alignement de données de l'en-tête de paquet PES a la valeur '1'. Dans ce cas, le premier octet de données de paquet PES qui suit l'en-tête PES est le premier octet d'une unité d'accès AVC ou d'une tranche AVC, comme indiqué par la valeur du type d'alignement.

Tableau 2-54 – Valeurs d'alignement de flux vidéo AVC

Type d'alignement	Description
00	Réservé
01	Tranche AVC ou unité d'accès AVC
02	Unité d'accès AVC
03-FF	Réservé

Le Tableau 2-55 décrit les types d'alignement audio qui correspondent à la valeur '1' du champ data_alignment_indicator dans l'en-tête de paquet PES. Dans ce cas, le premier octet de données de paquet PES suivant l'en-tête PES est le premier octet d'un mot de synchronisation audio.

Tableau 2-55 – Valeurs d'alignement du flux de données audio

Type d'alignement	Description
00	Valeur réservée
01	Mot de synchronisation
02-FF	Valeur réservée

2.6.12 Descripteur de grille de fond réceptrice

Il est possible d'avoir un ou plusieurs flux vidéo qui, une fois décodés, ne sont pas censés occuper toute la zone d'affichage (par exemple sur un moniteur). La combinaison des champs target_background_grid_descriptor et video_window_descriptors permet l'affichage de ces fenêtres vidéo aux emplacements souhaités correspondants. Le descripteur de grille de fond réceptrice sert à décrire une grille de pixels unitaires qui est projetée sur la zone d'affichage. Le descripteur de fenêtre vidéo est ensuite utilisé pour décrire, en ce qui concerne le flux associé, l'emplacement d'affichage, sur cette grille, du pixel supérieur gauche de la fenêtre ou du rectangle d'affichage de l'unité de présentation vidéo. Ce procédé est représenté à la Figure 2-3.

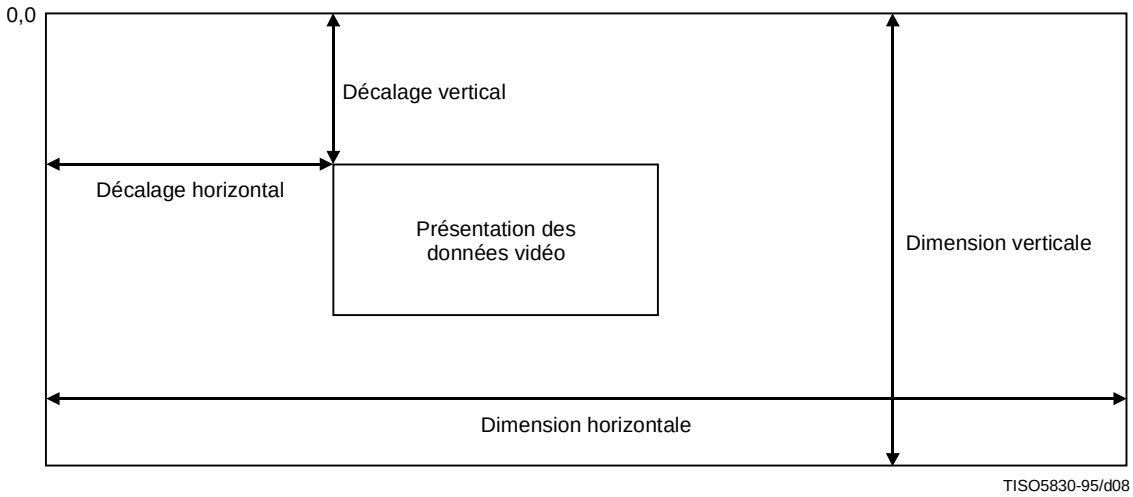


Figure 2-3 – Zone d'affichage par descripteur de la grille de fond réceptrice

2.6.13 Définition sémantique des champs dans un descripteur de grille de fond réceptrice

horizontal_size (dimension horizontale) – Ce champ indique la dimension horizontale (en pixels) de la grille de fond réceptrice.

vertical_size (dimension verticale) – Ce champ indique la dimension verticale (en pixels) de la grille de fond réceptrice.

aspect_ratio_information (information de format d'image) – Indications spécifiant le format d'échantillon ou d'affichage de la grille de fond réceptrice. Les données de format d'affichage sont définies dans la Rec. UIT-T H.262 | ISO/CEI 13818-2 (voir Tableau 2-56).

Tableau 2-56 – Descripteur de grille sur fond réceptrice

Syntaxe	Nombre de bits	Mnémonique
target_background_grid_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
horizontal_size	**14**	**uimsbf**
vertical_size	**14**	**uimsbf**
aspect_ratio_information	**4**	**uimsbf**
}		

2.6.14 Descripteur de fenêtre vidéo

Le descripteur de fenêtre vidéo sert à décrire les caractéristiques de fenêtre du flux élémentaire associé. Ses valeurs se rapportent au descripteur de grille de fond réceptrice dans le même flux. Voir également, au 2.6.12, le descripteur de grille de fond réceptrice (voir Tableau 2-57).

Tableau 2-57 – Descripteur de fenêtre vidéo

Syntaxe	Nombre de bits	Mnémonique
video_window_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
horizontal_offset	**14**	**uimsbf**
vertical_offset	**14**	**uimsbf**
window_priority	**4**	**uimsbf**
}		

2.6.15 Définition sémantique des champs dans un descripteur de fenêtre vidéo

horizontal_offset (décalage horizontal) – La valeur de ce champ indique la position horizontale du pixel situé en haut et à gauche de la fenêtre vidéo ou du rectangle d'affichage si celui-ci est indiqué dans l'extension d'affichage, découpé dans la grille de fond réceptrice, comme défini dans le champ target_background_grid_descriptor. Le pixel gauche supérieur de la fenêtre vidéo doit être l'un des pixels de la grille de fond réceptrice (voir Figure 2-3).

vertical_offset (décalage vertical) – La valeur de ce champ indique la position verticale du pixel situé en haut et à gauche de la fenêtre vidéo ou du rectangle d'affichage, si celui-ci est indiqué dans l'extension d'affichage, découpé dans la grille de fond réceptrice, comme défini dans le champ target_background_grid_descriptor. Le pixel gauche supérieur de la fenêtre vidéo doit être l'un des pixels de la grille de fond réceptrice (voir Figure 2-3).

window_priority (priorité de fenêtre) – Ce champ indique la priorité de recouvrement d'une fenêtre. Une valeur 0 indique la plus faible priorité et une valeur 15 la priorité la plus élevée; autrement dit une fenêtre ayant une priorité de 15 sera toujours visible.

2.6.16 Descripteur d'accès conditionnel

Le descripteur d'accès conditionnel sert à spécifier aussi bien des informations de gestion d'accès conditionnel à l'échelle du système que des informations spécifiques d'un flux élémentaire telles que les messages ECM. Cette structure peut être utilisée pour le champ TS_program_map_section (voir 2.4.4.8) comme pour le champ program_stream_map (voir 2.5.3). Si un flux élémentaire quelconque est embrouillé, un descripteur d'accès conditionnel est présent pour le programme contenant ce flux élémentaire. Si une information de gestion d'accès conditionnel à l'échelle du système existe dans un flux de transport, la table d'accès conditionnel contient un descripteur d'accès conditionnel.

Lorsque le descripteur d'accès conditionnel se trouve dans le champ TS_program_map_section (identificateur de table = 0x02), l'identificateur de paquet d'accès conditionnel (CA_PID) pointe sur les paquets contenant des informations d'accès conditionnel relatives au programme, telles que des messages ECM. La présence de ce descripteur sous forme d'informations complémentaires de programme indique qu'il est applicable à l'ensemble du programme. Dans le même cas, sa présence sous forme d'informations complémentaires de flux élémentaire indique qu'il est applicable au flux élémentaire associé. Le cas des données privées est également pris en compte.

Lorsque le descripteur d'accès conditionnel se trouve dans la section d'accès conditionnel (CA_section avec identificateur de table = 0x01), l'identificateur de paquet d'accès conditionnel (CA_PID) pointe sur les paquets contenant des informations de gestion à l'échelle du système et de gestion d'accès conditionnel, telles que des messages EMM.

Le contenu des paquets de flux de transport contenant des informations d'accès conditionnel est d'ordre privé (voir Tableau 2-58).

Tableau 2-58 – Descripteur d'accès conditionnel

Syntaxe	Nombre de bits	Mnémonique
CA_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
CA_system_ID	**16**	**uimsbf**
reserved	**3**	**bslbf**
CA_PID	**13**	**uimsbf**
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**uimsbf**
}		
}		

2.6.17 Définition sémantique des champs dans un descripteur d'accès conditionnel

CA_system_ID (identificateur de système d'accès conditionnel) – Codé sur 16 éléments binaires, ce champ indique le type de système d'accès conditionnel qui est applicable aux flux des messages ECM et EMM associés. Le codage de ce champ est défini sur base privée et n'est pas spécifié par l'UIT-T | ISO/CEI.

CA_PID (identificateur de paquet d'accès conditionnel) – Codé sur 13 éléments binaires, ce champ indique l'identificateur des paquets de transport qui doivent contenir des informations de message ECM ou EMM pour les systèmes d'accès conditionnel spécifiés par l'identificateur CA_system_ID associé. Le contenu (message ECM ou EMM) des paquets visés par l'identificateur CA_PID est déterminé d'après le contexte dans lequel se trouve ce champ, c'est-à-dire d'après la section de conditionnement de programme de flux de transport ou d'après la table de mappage de programme ou la table d'accès conditionnel contenue dans le flux de transport ou indiquée par le champ stream_id du flux de programme.

Dans les flux de transport, la présence d'un PID 0x03 indique qu'il existe une IPMP telle que décrite dans l'ISO/CEI 13818-11 utilisée par des composantes dans le flux de transport. Dans les flux de programme, la présence d'une valeur de stream_ID_extension de 0x00 indique que la IPMP décrite dans l'ISO/CEI 13818-11 est utilisée par des composantes dans le flux de programme. Dans un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1, les composantes pouvaient utiliser l'IPMP telle que décrite dans l'ISO/CEI 13818-11 ainsi que le CA tel que défini dans l'ISO/CEI 13818-1:2006. La compatibilité entre les deux systèmes est décrite dans l'ISO/CEI 13818-11.

2.6.18 Descripteur de code de langue selon l'ISO 639

Le descripteur de langue sert à spécifier la langue utilisée dans le flux élémentaire associé (voir Tableau 2-59).

Tableau 2-59 – Descripteur de code de langue selon l'ISO 639

Syntaxe	Nombre de bits	Mnémonique
ISO_639_language_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
ISO_639_language_code	**24**	**bslbf**
audio_type	**8**	**bslbf**
}		
}		

2.6.19 Définition sémantique des champs dans un descripteur de code de langue selon l'ISO 639

ISO_639_language_code (code de langue selon l'ISO 639) – Ce champ identifie la ou les langues utilisées par l'élément de programme associé. Ce champ est codé sur 3 caractères, comme spécifié dans ISO 639-2. Chaque caractère est codé sur un octet, conformément à l'ISO 8859-1 et inséré à sa place dans ce champ, qui contient donc 24 éléments binaires. En cas de flux audio multilingue, la séquence de champs ISO_639_language_code doit refléter le contenu du flux de données audio.

audio_type (type de données audio) – Codé sur un octet, ce champ spécifie le type de flux selon le Tableau 2-60.

Tableau 2-60 – Valeurs de type de données audio

Valeur	Description
0x00	Non défini
0x01	Effets d'écoute
0x02	Malentendants
0x03	Commentaire pour malvoyants
0x04-0x7F	Réservé à l'utilisateur
0x80-0xFF	Valeur réservée

clean_effects (effets d'écoute) – Ce champ indique que l'élément de programme visé ne contient pas de données linguistiques.

hearing_impaired (flux pour malentendants) – Ce champ indique que l'élément de programme visé est préparé pour les malentendants.

visual_impaired_commentary (commentaire pour malvoyants) – Ce champ indique que l'élément de programme visé est préparé pour les malvoyants.

2.6.20 Descripteur d'horloge système

Ce descripteur achemine des informations sur l'horloge système utilisée pour construire les horodateurs.

Le champ external_clock_reference_indicator doit être mis sur '1' si une référence temporelle externe a été utilisée. Sur option, le décodeur peut utiliser la même référence externe, si celle-ci est disponible.

Si l'horloge système a une dérive inférieure aux 30 millionièmes (ppm) requis, on peut communiquer cette précision d'horloge en la codant dans le champ clock_accuracy. La précision de la fréquence d'horloge est calculée comme suit:

$$clock_accuracy_integer \times 10^{-clock_accuracy_exponent}\ ppm \qquad (2\text{-}26)$$

Si le champ clock_accuracy_integer a la valeur '0', la précision d'horloge système est de 30 millionièmes. Lorsque le champ external_clock_reference_indicator est mis à '1', la précision d'horloge se rapporte à celle de la référence extérieure (voir Tableau 2-61).

Tableau 2-61 – Descripteur d'horloge de base du système

Syntaxe	Nombre de bits	Mnémonique
system_clock_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
external_clock_reference_indicator	**1**	**bslbf**
reserved	**1**	**bslbf**
clock_accuracy_integer	**6**	**uimsbf**
clock_accuracy_exponent	**3**	**uimsbf**
Reserved	**5**	**bslbf**
}		

2.6.21 Définition sémantique des champs dans un descripteur d'horloge système

external_clock_reference_indicator (indicateur de référence temporelle externe) – S'il est mis à '1', ce fanion codé sur 1 bit indique que l'horloge système a été calée sur une fréquence étalon externe à laquelle le décodeur peut accéder.

clock_accuracy_integer (entier de précision d'horloge) – Codé sur 6 éléments binaires, ce champ donne, en association avec le champ clock_accuracy_exponent, la valeur fractionnaire de stabilité en fréquence de l'horloge système exprimée en millionièmes.

clock_accuracy_exponent (exposant de précision d'horloge) – Codé sur 3 éléments binaires, ce champ donne, en association avec le champ clock_accuracy_integer, la valeur fractionnaire de stabilité en fréquence de l'horloge système, exprimée en millionièmes.

2.6.22 Descripteur d'utilisation du tampon de multiplexage

Le descripteur d'utilisation du tampon de multiplexage donne les limites du taux d'occupation du tampon de multiplexage du décodeur STD. Cette information est destinée à des dispositifs tels que les remultiplexeurs, qui peuvent l'utiliser pour les besoins d'une stratégie donnée de remultiplexage (voir Tableau 2-62).

Tableau 2-62 – Descripteur d'utilisation du tampon de multiplexage

Syntaxe	Nombre de bits	Mnémonique
multiplex_buffer_utilization_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
bound_valid_flag	**1**	**bslbf**
LTW_offset_lower_bound	**15**	**uimsbf**
reserved	**1**	**bslbf**
LTW_offset_upper_bound	**15**	**uimsbf**
}		

2.6.23 Définition sémantique des champs dans un descripteur d'utilisation du tampon de multiplexage

bound_valid_flag (fanion de validité de limite) – Mis à la valeur '1' ce fanion indique la validité des champs LTW_offset_lower_bound et LTW_offset_upper_bound des limites inférieure et supérieure de décalage par rapport à la fenêtre sur temps légal.

LTW_offset_lower_bound (limite inférieure de décalage LTW) – Ce champ codé sur 15 éléments binaires est défini uniquement si le fanion bound_valid est mis à la valeur '1'. Lorsque ce champ est défini, sa valeur est exprimée en périodes (27 MHz / 300) de l'horloge du système, suivant la définition concernant LTW_offset (voir 2.4.3.4). La limite inférieure du décalage LTW correspond à la plus faible valeur d'un champ LTW_offset, si ce champ était codé dans chaque paquet du flux ou des flux référencés par ce descripteur. Les champs LTW_offset réels ne sont pas nécessairement codés dans le flux binaire lorsque le descripteur de taux d'utilisation de tampon de multiplexage est présent. Cette limite est valide jusqu'à la prochaine occurrence de ce descripteur.

LTW_offset_upper_bound (limite supérieure de décalage LTW) – Ce champ codé sur 15 éléments binaires est défini uniquement si le fanion bound_valid est mis à la valeur '1'. Lorsque ce champ est défini, sa valeur est exprimée en périodes (27 MHz / 300) de l'horloge du système, suivant la définition concernant LTW_offset (voir 2.4.3.4). La limite supérieure du décalage LTW correspond à la valeur la plus élevée d'un champ LTW_offset, si ce champ était codé dans chaque paquet du flux ou des flux référencés par ce descripteur. Les champs LTW_offset réels ne sont pas nécessairement codés dans le flux binaire lorsque le descripteur de taux d'utilisation de tampon de multiplexage est présent. Cette limite est valide jusqu'à la prochaine occurrence de ce descripteur.

2.6.24 Descripteur de droit d'auteur

Le descripteur de droit d'auteur fournit une méthode pour permettre l'identification d'œuvres audiovisuelles. Cette structure s'applique à des programmes ou à des flux élémentaires contenus dans des programmes (voir Tableau 2-63).

Tableau 2-63 – Descripteur de droit d'auteur

Syntaxe	Nombre de bits	Mnémonique
copyright_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
copyright_identifier	**32**	**uimsbf**
for (i = 0; i < N; i++){		
additional_copyright_info	**8**	**bslbf**
}		
}		

2.6.25 Définition sémantique des champs dans un descripteur de droit d'auteur

copyright_identifier (identificateur de droit d'auteur) – Codée sur 32 éléments binaires, la valeur de ce champ indique l'organisme d'enregistrement.

additional_copyright_info (information supplémentaire sur le droit d'auteur) – Les valeurs des octets de ce champ, s'ils existent, sont définies par le destinataire désigné par l'identificateur de droit d'auteur et ne doivent plus changer ensuite.

2.6.26 Descripteur de débit maximal

Voir Tableau 2-64.

Tableau 2-64 – Descripteur de débit maximal

Syntaxe	Nombre de bits	Mnémonique
maximum_bitrate_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**2**	**bslbf**
maximum_bitrate	**22**	**uimsbf**
}		

2.6.27 Définition sémantique des champs dans un descripteur de débit maximal

maximum_bitrate (débit maximal) – Codé sur 22 éléments binaires, ce champ contient un entier positif qui indique la limite supérieure du débit maximal, y compris la servitude binaire de transport, qui sera constatée dans le programme ou l'élément de programme. La valeur codée dans le champ maximum_bitrate est exprimée en unités de 50 octets par seconde. Le descripteur de débit maximal est donné dans la table de mappage de programme (PMT, *program map table*). Sa présence en tant qu'information complémentaire de programme indique que ce débit est applicable à l'ensemble du programme. Sa présence en tant qu'information de flux élémentaire indique que le débit est applicable à l'élément de programme associé.

2.6.28 Descripteur d'indicateur de données privées

Voir Tableau 2-65.

Tableau 2-65 – Descripteur d'indication de données privées

Syntaxe	Nombre de bits	Mnémonique
private_data_indicator_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
private_data_indicator	**32**	**uimsbf**
}		

2.6.29 Définition sémantique des champs dans un descripteur d'indicateur de données privées

private_data_indicator (indicateur de données privées) – La valeur de ce champ est privée et ne fera pas l'objet d'une définition de l'UIT-T | ISO/CEI.

2.6.30 Descripteur de tampon de lissage

Ce descripteur optionnel contient des données concernant la capacité d'une mémoire de lissage SB_n, associée à ce descripteur, et concernant le débit débordement associé à cette mémoire, pour l'élément ou les éléments de programme correspondants.

Dans le cas de flux de transport, les octets des paquets de transport de l'élément ou des éléments de programme associés présents dans le flux de transport sont introduits dans une mémoire tampon SB_n dont la capacité est donnée par la valeur de sb_size à l'instant défini par la relation 2-4.

Dans le cas de flux de programme, les octets de tous les paquets PES des flux élémentaires associés sont introduits dans une mémoire tampon SB_n dont la capacité est indiquée par la valeur de sb_size, à l'instant défini par la relation 2-21.

Lorsque cette mémoire contient des données, l'extraction des octets qui s'y trouvent s'effectue selon un débit indiqué par la valeur de sb_leak_rate. La mémoire SB_n ne doit jamais déborder. Tout au long de la durée ininterrompue d'existence d'un programme, la valeur des éléments du descripteur de la mémoire de lissage des différents éléments de programme que comporte le programme en question doit rester inchangée.

Le descripteur de mémoire de lissage n'a une signification définie que s'il figure dans la table de mappage de programme ou dans la table de mappage de flux de programme.

Si, dans le cas d'un flux de transport, le descripteur est présent dans la structure ES info dans la table de mappage de programme, tous les paquets de flux de transport de même identificateur PID que l'élément de programme considéré sont introduits dans la mémoire de lissage.

Si, dans le cas d'un flux de transport, le descripteur figure parmi les informations de programme, les paquets de flux de transport suivants sont introduits dans la mémoire de lissage:

- tous les paquets de flux de transport dont les identificateurs PID sont répertoriés en tant qu'identificateurs élémentaires dans les informations de programme complémentaires; ainsi que
- tous les paquets de flux de transport de même identificateur PID que la table de mappage de programme de cette section;
- tous les paquets de flux de transport ayant le PCR_PID du programme.

Tous les octets introduits dans la mémoire associée en sortent également.

A un moment donné quelconque, il ne doit y avoir plus d'un descripteur se rapportant à un élément de programme particulier et il ne doit pas y avoir plus d'un descripteur se rapportant au programme dans son ensemble.

Tableau 2-66 – Descripteur de mémoire de lissage

Syntaxe	Nombre de bits	Mnémonique
smoothing_buffer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**2**	**bslbf**
sb_leak_rate	**22**	**uimsbf**
reserved	**2**	**bslbf**
sb_size	**22**	**uimsbf**
}		

2.6.31 Définition sémantique des champs dans un descripteur de mémoire de lissage

sb_leak_rate (débit débordement de mémoire de lissage) – Codé sur 22 éléments binaires, ce champ est un nombre entier positif égal au débit débordement de la mémoire de lissage SB_n correspondant au flux élémentaire associé ou à d'autres données, exprimé en unités de 400 bit/s.

sb_size (capacité de mémoire de lissage) – Codé sur 22 éléments binaires, le champ sb_size indique la valeur d'un nombre entier positif égal à la capacité de la mémoire de lissage du tampon de multiplexage SB_n correspondant au flux élémentaire associé ou à d'autres données, exprimé en nombre d'octets (voir Tableau 2-66).

2.6.32 Descripteur STD

Ce descripteur facultatif s'applique uniquement au décodeur T-STD et aux flux élémentaires vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2; les modalités de son utilisation sont spécifiées au 2.4.2. Ce descripteur ne s'applique pas aux flux de programme (voir Tableau 2-67).

Tableau 2-67 – Descripteur de modèle STD

Syntaxe	Nombre de bits	Mnémonique
STD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**7**	**bslbf**
leak_valid_flag	**1**	**bslbf**
}		

2.6.33 Définition sémantique des champs du descripteur STD

leak_valid_flag (fanion de validité de méthode du débordement) – Le fanion leak_valid est codé sur un bit. Lorsqu'il est mis à la valeur '1', le transfert de données entre la mémoire MB_n et la mémoire EB_n à l'intérieur du décodeur T-STD s'effectue selon la méthode du débordement décrite au 2.4.2.3. Lorsqu'il est mis à la valeur '0' et si les champs vbv_delay présents dans le flux vidéo associé ne sont mis à la valeur 0xFFFF, le transfert de données entre la mémoire MB_n et la mémoire EB_n s'effectue selon la méthode du délai de vérificateur VBV décrite au 2.4.2.3.

2.6.34 Descripteur IBP

Ce descripteur facultatif fournit des informations concernant certaines caractéristiques de la séquence de types de trames d'un flux vidéo ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 14496-2 (voir Tableau 2-68).

Tableau 2-68 – Descripteur IBP

Syntaxe	Nombre de bits	Mnémonique
ibp_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
closed_gop_flag	**1**	**uimsbf**
identical_gop_flag	**1**	**uimsbf**
max_gop-length	**14**	**uimsbf**
}		

2.6.35 Définition sémantique des champs du descripteur IBP

closed_gop_flag – Mis à la valeur '1', ce fanion codé sur 1 bit indique qu'un en-tête de groupe d'en-têtes d'images est codé avant chaque image de type I et que ce même fanion est mis à la valeur '1' dans tous les en-têtes de groupes d'images de la séquence vidéo.

identical_gop_flag – Mis à la valeur '1', ce fanion codé sur 1 bit indique le nombre de trames de type P et de trames de type B compris entre des trames de type I, et que les types de codage d'images ainsi que la séquence de types d'image entre les images à codage intratrame, sont identiques dans toute la séquence, sauf éventuellement pour les images situées jusqu'à la deuxième image à codage intratrame.

max_gop_length – La valeur de ce champ codé sur 14 éléments binaires correspond à un nombre entier arithmétique indiquant le nombre maximal d'images codées compris entre deux images consécutives à codage intratrame de la séquence. La valeur zéro ne doit pas être utilisée.

2.6.36 Descripteur MPEG-4 vidéo

En ce qui concerne les flux distincts ISO/CEI 14496-2 directement acheminés dans des paquets PES, tels qu'ils sont définis au 2.11.2, le descripteur MPEG-4 vidéo fournit des renseignements de base pour l'identification des paramètres de codage de ces flux élémentaires vidéo. Le descripteur MPEG-4 vidéo ne s'applique pas aux flux ISO/CEI 14496-2 encapsulés dans des paquets SL et FlexMux, tels que définis dans au 2.11.3.

Tableau 2-69 – Descripteur MPEG-4 vidéo

Syntaxe	Nombre de bits	Mnémonique
MPEG-4_video_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-4_visual_profile_and_level	**8**	**uimsbf**
}		

2.6.37 Définition sémantique des champs dans le descripteur MPEG-4 vidéo

MPEG-4_video_profile_and_level (profil et niveau du flux vidéo MPEG-4) – Codé sur 8 bits, ce champ identifie le profil et le niveau du flux vidéo ISO/CEI 14496-2. Ce champ doit être codé avec la même valeur que le champ profile_and_level_indication contenu dans l'en-tête de séquence d'objet vidéo dans le flux ISO/CEI 14496-2 associé.

2.6.38 Descripteur MPEG-4 audio

En ce qui concerne les flux distincts ISO/CEI 14496-3 directement acheminés dans des paquets PES, tels que définis au 2.11.2, le descripteur MPEG-4_audio_descriptor fournit des renseignements de base pour l'identification des paramètres de codage de ces flux élémentaires audio. Le descripteur MPEG-4_audio ne s'applique pas aux flux ISO/CEI 14496-3 encapsulés dans des paquets SL et FlexMux, tels que définis au 2.11.3.

Tableau 2-70 – Descripteur MPEG-4 audio

Syntaxe	Nombre de bits	Mnémonique
MPEG-4_audio_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-4_audio_profile_and_level	**8**	**uimsbf**
}		

2.6.39 Définition sémantique des champs du descripteur MPEG-4 audio

MPEG-4_audio_profile_and_level (profil et niveau du flux audio MPEG-4) – Codé sur 8 bits, ce champ identifie le profil et le niveau du flux audio ISO/CEI 14496-3 (voir Tableau 2-71).

Tableau 2-71 – Valeurs d'affectation MPEG-4_audio_profile_and_level

Valeur	Description
0x00-0x0F	Réservé
0x10	Profil principal, niveau 1
0x11	Profil principal, niveau 2
0x12	Profil principal, niveau 3
0x13	Profil principal, niveau 4
0x14-0x17	Réservé
0x18	Profil échelonnable, niveau 1
0x19	Profil échelonnable, niveau 2
0x1A	Profil échelonnable, niveau 3
0x1B	Profil échelonnable, niveau 4
0x1C-0x1F	Réservé
0x20	Profil conversation, niveau 1
0x21	Profil conversation, niveau 2
0x22-0x27	Réservé
0x28	Profil synthèse, niveau 1
0x29	Profil synthèse, niveau 2
0x2A	Profil synthèse, niveau 3
0x2B-0x2F	Réservé

Tableau 2-71 – Valeurs d'affectation MPEG-4_audio_profile_and_level

Valeur	Description
0x30	Profil audio de haute qualité, niveau 1
0x31	Profil audio de haute qualité, niveau 2
0x32	Profil audio de haute qualité, niveau 3
0x33	Profil audio de haute qualité, niveau 4
0x34	Profil audio de haute qualité, niveau 5
0x35	Profil audio de haute qualité, niveau 6
0x36	Profil audio de haute qualité, niveau 7
0x37	Profil audio de haute qualité, niveau 8
0x38	Profil audio à faible délai, niveau 1
0x39	Profil audio à faible délai, niveau 2
0x3A	Profil audio à faible délai, niveau 3
0x3B	Profil audio à faible délai, niveau 4
0x3C	Profil audio à faible délai, niveau 5
0x3D	Profil audio à faible délai, niveau 6
0x3E	Profil audio à faible délai, niveau 7
0x3F	Profil audio à faible délai, niveau 8
0x40	Profil audio naturel, niveau 1
0x41	Profil audio naturel, niveau 2
0x42	Profil audio naturel, niveau 3
0x43	Profil audio naturel, niveau 4
0x44-0x47	Réservé
0x48	Profil réseautique mobile audio, niveau 1
0x49	Profil réseautique mobile audio, niveau 2
0x4A	Profil réseautique mobile audio, niveau 3
0x4B	Profil réseautique mobile audio, niveau 4
0x4C	Profil réseautique mobile audio, niveau 5
0x4D	Profil réseautique mobile audio, niveau 6
0x4E-0x4F	Réservé
0x50	Profil AAC, niveau 1
0x51	Profil AAC, niveau 2
0x52	Profil AAC, niveau 4
0x53	Profil AAC, niveau 5
0x54-0x57	Réservé
0x58	Profil AAC à haute efficacité, niveau 2
0x59	Profil AAC à haute efficacité, niveau 3
0x5A	Profil AAC à haute efficacité, niveau 4
0x5B	Profil AAC à haute efficacité, niveau 5
0x5C-0xFF	Réservé

2.6.40 Descripteur IOD

Le descripteur IOD encapsule la structure InitialObjectDescriptor. Un descripteur d'objet initial permet d'accéder à un ensemble de flux ISO/CEI 14496 en identifiant les valeurs ES_ID des flux de description de scène et de descripteur d'objet ISO/CEI 14496-1. Ces flux contiennent tous les deux des informations supplémentaires sur les flux ISO/CEI 14496 qui font partie de la scène. Voir l'Annexe R pour une description de la procédure d'accès au contenu. Le descripteur InitialObjectDescriptor est spécifié au 8.6.3 de l'ISO/CEI 14496-1.

Dans un flux de transport, le descripteur IOD doit être inséré dans la boucle de descripteur qui suit immédiatement le champ program_info_length dans la table PMT. Lorsqu'un conditionnement de flux de programme est présent dans un flux de programme, le descripteur IOD est inséré dans la boucle de descripteur qui suit immédiatement le champ program_stream_info_length dans le conditionnement du flux de programme. Plusieurs descripteurs IOD peuvent être associés à un programme.

NOTE – La présente Spécification n'indique pas la manière dont l'étiquette IOD_label peut être utilisée par des informations de service de niveau supérieur pour la sélection d'une seule présentation ISO/CEI 14496 identifiée par des descripteurs IOD multiples.

Tableau 2-72 – Descripteur IOD

Syntaxe	Nombre de bits	Mnémonique
IOD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
Scope_of_IOD_label	**8**	**uimsbf**
IOD_label	**8**	**uimsbf**
InitialObjectDescriptor ()	**8**	**uimsbf**
}		

2.6.41 Définition sémantique des champs du descripteur IOD

Scope_of_IOD_label (domaine d'application de l'étiquette d'IOD) – Codé sur 8 bits, ce champ spécifie le domaine d'application de l'étiquette d'IOD. La valeur 0x10 indique que l'IOD_label est unique à l'intérieur du flux de programme ou à l'intérieur du programme spécifique dans un flux de transport dans lequel le descripteur IOD est inséré. La valeur 0x11 indique que l'IOD_label est unique dans le flux de transport dans lequel le descripteur IOD est inséré. Les autres valeurs de ce champ sont réservées.

IOD_label (étiquette du descripteur IOD) – Codé sur 8 bits, ce champ spécifie l'étiquette du descripteur IOD.

InitialObjectDescriptor () (descripteur d'objet initial) – Cette structure est définie au 8.6.3.1 de l'ISO/CEI 14496-1.

2.6.42 Descripteur SL

Le descripteur SL doit être utilisé lorsqu'un seul flux de paquets SL selon l'ISO/CEI 14496-1 est encapsulé dans des paquets PES. Le descripteur SL associe l'identificateur ES_ID de ce champ de paquets SL à un identificateur elementary_PID pour un flux de transport ou à un identificateur elementary_stream_id pour un flux de programme. A l'intérieur d'un flux de transport, le descripteur SL correspondant au flux élémentaire doit être inséré dans la boucle de descripteur qui suit immédiatement le champ ES_info_length dans la table PMT. Si un conditionnement du flux de programme est présent dans un flux de programme, le descripteur SL doit être inséré dans la boucle qui suit immédiatement le champ elementary_stream_info_length dans le conditionnement de programme.

NOTE – Des flux de paquets SL peuvent être utilisés dans un flux de programme. Cependant, il n'existe qu'un identificateur stream_id pour les flux de paquets SL selon l'ISO/CEI 14496-1. Afin d'associer divers flux de ce type contenus dans un flux de programme à une scène ISO/CEI 14496-1, le flux FlexMux doit être utilisé et signalé correctement par un descripteur FMC. Cette restriction ne s'applique pas à un flux de transport où le descripteur SL assure un mappage non ambigu entre une valeur d'identificateur ES_ID selon l'ISO/CEI 14496-1 et une valeur d'identificateur elementary_PID selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Tableau 2-73 – Descripteur SL

Syntaxe	Nombre de bits	Mnémonique
SL_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
ES_ID	**16**	**uimsbf**
}		

2.6.43 Définition sémantique des champs du descripteur SL

ES_ID (identificateur de flux élémentaire) – Codé sur 16 bits, ce champ indique l'identificateur d'un flux de paquets SL selon l'ISO/CEI 14496-1.

2.6.44 Descripteur FMC

Le descripteur FMC indique que l'outil FlexMux ISO/CEI 14496-1 a été utilisé pour le multiplexage des flux de paquets SL selon l'ISO/CEI 14496-1 dans un flux FlexMux avant leur encapsulation dans des paquets PES ou dans des sections ISO/CEI 14496. Le descripteur FMC associe des voies FlexMux aux valeurs ES_ID des flux de paquets SL contenus dans le flux FlexMux.

Un descripteur FMC est nécessaire pour chaque élément de programme désigné par une valeur d'identificateur elementary_PID dans un flux de transport et pour chaque identificateur de flux élémentaire dans un flux de programme qui achemine un flux FlexMux. Dans un flux de programme, le descripteur FMC correspondant au flux élémentaire doit être inséré dans la boucle de descripteur qui suit immédiatement le champ ES_info_length dans la table PMT. Si un conditionnement de programme est présent dans un flux de programme, le descripteur FMC doit être inséré dans la boucle qui suit immédiatement le champ elementary_stream_info_length dans le conditionnement du flux de programme.

Pour chaque flux de paquets SL contenu dans un flux FlexMux, la voie FlexMux doit être identifiée par une seule entrée dans le descripteur FMC.

Tableau 2-74 – Descripteur FMC

Syntaxe	Nombre de bits	Mnémonique
FMC_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i=0; i < descriptor_length; i += 3) {		
ES_ID	**16**	**uimsbf**
FlexMuxChannel	**8**	**uimsbf**
}		
}		

2.6.45 Définition sémantique des champs du descripteur FMC

ES_ID (identificateur de flux élémentaire) – Codé sur 16 bits, ce champ indique l'identificateur d'un flux de paquets SL ISO/CEI 14496-1.

FlexMuxChannel (voie FlexMux) – Codé sur 8 bits, ce champ indique le numéro de la voie FlexMux utilisée pour ce flux de paquets SL.

2.6.46 Descripteur External_ES_ID

Le descripteur External_ES_ID attribue un identificateur ES_ID, tel que défini dans l'ISO/CEI 14496-1, à un élément de programme auquel aucune valeur d'ES_ID n'a été attribuée par d'autres moyens. Cet identificateur ES_ID permet de faire référence à l'élément non ISO/CEI 14496 dans la description de scène ou, par exemple, d'associer l'élément non ISO/CEI 14496 à un flux IPMP.

Dans un flux de transport, l'attribution d'un identificateur ES_ID doit être effectuée par l'insertion d'un descripteur External_ES_ID correspondant au flux élémentaire dans la boucle de descripteur qui suit immédiatement le champ ES_info_length dans la table PMT. Si un conditionnement de programme est présent dans un flux de programme, le descripteur ES_ID doit être inséré dans la boucle qui suit immédiatement le champ elementary_stream_info_length dans le conditionnement du flux de programme.

Tableau 2-75 – Descripteur External_ES_ID

Syntaxe	Nombre de bits	Mnémonique
External_ES_ID_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
External_ES_ID	**16**	**uimsbf**
}		

2.6.47 Définition sémantique des champs du descripteur External_ES_ID

External_ES_ID (identificateur de flux élémentaire externe) – Codé sur 16 bits, ce champ indique un identificateur ES_ID tel que défini dans l'ISO/CEI 14496-1, attribué à un élément de programme.

2.6.48 Descripteur Muxcode

Le descripteur Muxcode assure l'acheminement des structures MuxCodeTableEntry définies au 11.2.4.3 de l'ISO/CEI 14496-1. Les entrées MuxCodeTableEntries configurent le mode MuxCode du champ FlexMux.

Un ou plusieurs descripteurs Muxcode peuvent être associés à chaque identificateur elementary_PID ou elementary_stream_id, acheminant chacun un flux FlexMux ISO/CEI 14496-1 qui utilise le mode Muxcode. Dans un flux de transport, le descripteur Muxcode correspondant au flux élémentaire doit être inséré dans la boucle de descripteur qui suit immédiatement le champ ES_info_length dans la table PMT. Si un conditionnement de programme est présent dans un flux de programme, le descripteur Muxcode doit être inséré dans la boucle qui suit immédiatement le champ elementary_stream_info_length dans le conditionnement du flux de programme.

Les entrées MuxCodeTableEntries peuvent être mises à jour avec les nouvelles versions. Dans le cas de ces versions mises à jour, le numéro de version de chaque table PMT ou la version program_stream_map_version de chaque conditionnement du flux de programme, acheminant le descripteur Muxcode dans leur boucle de descripteur, sera incrémenté de la valeur 1 puis ramené à la valeur modulo 32.

Tableau 2-76 – Descripteur Muxcode

Syntaxe	Nombre de bits	Mnémonique
Muxcode_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
MuxCodeTableEntry ()		
}		
}		

2.6.49 Définition sémantique des champs du descripteur Muxcode

MuxCodeTableEntry () – Cette structure est définie au 11.2.4.3 de l'ISO/CEI 14496-1.

2.6.50 Descripteur FmxBufferSize

Le descripteur FmxBufferSize achemine la capacité de mémoire tampon (FB) FlexMux pour chaque flux de paquets SL multiplexé dans un flux FlexMux.

Un descripteur FmxBufferSize doit être associé à chaque identificateur elementary_PID ou elementary_stream_id, acheminant un flux FlexMux ISO/CEI 14496-1. Dans un flux de transport, le descripteur FmxBufferSize correspondant au flux élémentaire doit être inséré dans la boucle de descripteur qui suit immédiatement le champ ES_info_length dans la table PMT. Si un conditionnement de programme est présent dans un flux de programme, le descripteur FmxBufferSize doit être inséré dans la boucle qui suit immédiatement le champ elementary_stream_info_length dans le conditionnement du flux de programme.

Tableau 2-77 – Descripteur FmxBufferSize

Syntaxe	Nombre de bits	Mnémonique
FmxBufferSize_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
DefaultFlexMuxBufferDescriptor()		
for (i=0; i<descriptor_length; i += 4) {		
FlexMuxBufferDescriptor()		
}		
}		

2.6.51 Définition sémantique des champs du descripteur FmxBufferSize

FlexMuxBufferDescriptor() – Ce descripteur indique la capacité de mémoire tampon FlexMux d'un flux de paquets SL acheminé dans le flux FlexMux. Il est défini au 11.2 de l'ISO/CEI 14496-1.

DefaultFlexMuxBufferDescriptor() – Ce descripteur indique la capacité de mémoire tampon par défaut FlexMux de ce flux FlexMux. Il est défini au 11.2 de l'ISO/CEI 14496-1.

2.6.52 Descripteur MultiplexBuffer

Le descripteur MultiplexBuffer achemine la capacité de mémoire tampon de multiplexage MB_n, ainsi que le débit de débordement Rx_n avec lequel les données sont transférées de la mémoire tampon de transport TB_n à la mémoire tampon MB_n pour un élément de programme UIT-T H.222.0 | ISO/CEI 13818-1 spécifique référencé par une valeur d'identificateur elementary_PID.

Un descripteur MultiplexBuffer doit être associé à chaque identificateur elementary_PID contenant un flux FlexMux ISO/CEI 14496 ou un flux de paquets SL, y compris ceux qui contiennent des sections ISO/CEI 14496. Voir 2.11.3.9 pour la définition de ces mémoires tampons et débits dans le modèle T-STD pour le décodage du contenu ISO/CEI 14496.

Le descripteur MultiplexBuffer doit être acheminé dans la boucle du descripteur suivant immédiatement le champ ES_info_length dans la table PMT.

Tableau 2-78 – Descripteur MultiplexBuffer

Syntaxe	Nombre de bits	Mnémonique
MultiplexBuffer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MB_buffer_size	**24**	**uimsbf**
TB_leak_rate	**24**	**uimsbf**
}		

2.6.53 Définition sémantique des champs du descripteur MultiplexBuffer

MB_buffer_size (capacité de la mémoire tampon de multiplexage) **–** Codé sur 24 bits, ce champ indique la capacité en octets de la mémoire tampon MB_n du flux élémentaire n qui est associé au descripteur.

TB_leak_rate (débit de débordement de la mémoire tampon de transport) **–** Codé sur 24 bits, ce champ indique le débit, en unités de 400 bits/s, auquel les données sont transférées de la mémoire tampon de transport TB_n à la mémoire de multiplexage MB_n, pour le flux élémentaire n qui est associé au descripteur.

2.6.54 Descripteur FlexMuxTiming

Voir Tableau 2-79.

Tableau 2-79 – Descripteur FlexMuxTiming

Syntaxe	Nombre de bits	Mnémonique
FlexMuxTiming_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
FCR_ES_ID	**16**	**uimsbf**
FCRResolution	**32**	**uimsbf**
FCRLength	**8**	**uimsbf**
FmxRateLength	**8**	**uimsbf**
}		

2.6.55 Définition sémantique des champs du descripteur FlexMuxTiming

FCR_ES_ID – Identificateur ES_ID associé à ce flux de références d'horloge.

FCRResolution – Résolution de la base de temps de l'objet en cycles par seconde.

FCRLength – Longueur du champ fmxClockReference dans les paquets FlexMux où index = 238. Une longueur de zéro indique qu'aucun paquet FlexMux où index = 238 n'est présent dans le flux FlexMux. La valeur de ce champ peut être comprise entre zéro et 64.

FmxRateLength – Longueur du champ fmxRate dans les paquets FlexMux où index = 238. La valeur de ce champ peut varier entre 1 et 32.

2.6.56 Descripteur d'étiquetage de contenu

Le descripteur d'étiquetage de contenu attribue une étiquette au contenu; l'étiquette peut être utilisée par les métadonnées pour référencer le contenu associé. Cette étiquette (content_reference_id_record) (enregistrement d'identificateur de référence de contenu) est propre au format d'application des métadonnées. Le descripteur d'étiquetage de contenu est associé à un segment de contenu. Pour les besoins qui nous occupent dans le présent paragraphe, on entend par segment de contenu une fraction de temps d'un programme, un flux élémentaire (audio ou vidéo, par exemple) ou un ensemble quelconque de programmes ou de flux élémentaires. Le descripteur peut être inclus dans la table de mappage de programme (PMT, *program map table*) dans la boucle de descripteur pour le programme ou un flux élémentaire, mais il peut aussi figurer dans des tableaux non définis dans la présente Spécification, tels que des tableaux visant à décrire des segments de programmes ou des flux élémentaires. Le descripteur d'étiquetage de contenu donne également des informations sur le type de base de temps qui est utilisé pour le contenu ainsi que sur le décalage entre cette base de temps et la base de temps pour les métadonnées. Lorsque le principe de durée de lecture normale (NPT, *normal play time*) de la commande DSM-CC, telle que spécifiée dans l'ISO/CEI 13818-6, est utilisé comme base de temps pour le contenu, l'identificateur (ID) de la base de temps NPT est fourni. Le descripteur permet d'acheminer des données privées. Voir le Tableau 2-80.

Tableau 2-80 – Descripteur d'étiquetage de contenu

Syntaxe	Nombre de bits	Mnémonique
Content_labeling_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	**32**	**uimsbf**
}		
content_reference_id_record_flag	**1**	**bslbf**
content_time_base_indicator	**4**	**uimsbf**
reserved	**3**	**bslbf**
if (content_reference_id_record_flag == '1'){		
content_reference_id_record_length	**8**	**uimsbf**
for (i=0; i<content_reference_id_record_length;i++){		
content_reference_id_byte	**8**	**bslbf**
}		
}		
if (content_time_base_indicator== 1|2){		
reserved	**7**	**bslbf**
content_time_base_value	**33**	**uimsbf**
reserved	**7**	**bslbf**
metadata_time_base_value	**33**	**uimsbf**
}		
if (content_time_base_indicator== 2){		
reserved	**1**	**bslbf**
contentId	**7**	**uimsbf**
}		
if (content_time_base_indicator==3|4|5|6|7){		
time_base_association_data_length	**8**	**uimsbf**
for (i=0; i< time_base_association_data_length;i++){		
reserved	**8**	**bslbf**
}		
}		
for (i=0; i<N;i++){		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.57 Définition sémantique des champs du descripteur d'étiquetage de contenu

metadata_application_format (format d'application des métadonnées) – Codé sur 16 bits, comme indiqué dans le Tableau 2-81, ce champ indique l'application qui doit définir l'usage, la syntaxe et la sémantique du champ content_reference_id record et de tout autre champ défini à titre privé dans ce descripteur. Voir aussi 2.12.1. La valeur 0xFFFF indique que le format est signalé par la valeur acheminée dans le champ metadata_application_format_identifier field.

Tableau 2-81 – Metadata_application_format

Valeur	Description
0x0000-0x000F	Valeur réservée
0x0010	ISO 15706 (ISAN) codé sous sa forme binaire (voir les Notes 1 et 3)
0x0011	ISO 15706-2 (V-ISAN) codé sous sa forme binaire (voir les Notes 2 et 3)
0x0012-0x00FF	Valeur réservée
0x0100-0xFFFE	Valeur définie par l'utilisateur
0xFFFF	Valeur définie par le champ metadata_application_format_identifier

NOTE 1 – Pour le numéro ISAN, le champ content_reference_id_byte (octet d'identificateur de référence de contenu) est mis sur la position de codage binaire et le champ content_reference_id_record_length (longueur d'enregistrement d'identificateur de référence de contenu) est mis sur 0x08.

NOTE 2 – Pour le numéro V-ISAN, le champ content_reference_id_byte (octet d'identificateur de référence de contenu) est mis sur la position de codage binaire et le champ content_reference_id_record_length (longueur d'enregistrement d'identificateur de référence de contenu) est mis sur 0x0C.

NOTE 3 – Pour assurer l'interopérabilité des applications de métadonnées qui utilisent les valeurs 0x0010 et 0x0011 pour le champ metadata_application_format (format d'application des métadonnées), il est recommandé de mettre le fanion d'enregistrement d'identificateur de référence de contenu (content_reference_id_flag) sur '1' et de mettre le champ content_time_base_indicator (indicateur de base de temps pour le contenu) sur '00'.

metadata_application_format_identifier (identificateur de format d'application de métadonnées) – Champ de 32 éléments binaires, dont le codage est entièrement équivalent à celui du champ format_identifier du descripteur d'enregistrement (registration_descriptor) défini au 2.6.8.

NOTE – L'organisme d'enregistrement désigné pour le champ format_identifier est la SMPTE.

content_reference_id_record_flag (fanion d'enregistrement d'identificateur de référence de contenu) – Fanion d'un élément binaire qui indique la présence d'un enregistrement content_reference_id_record_ dans ce descripteur.

content_time_base_indicator (indicateur de base de temps pour le contenu) – Champ de 4 éléments binaires qui indique la base de temps utilisée pour le contenu. Si le descripteur est associé à un programme, la base de temps pour le contenu s'applique à tous les flux qui font partie de ce programme. Une valeur de 1 indique l'utilisation de l'horloge de base du système (STC, *system time clock*), alors qu'une valeur de '2' indique l'utilisation de la durée de lecture normale (NPT) définie dans l'ISO/CEI 13818-6. Les valeurs de 8 à 15 indiquent l'utilisation d'une base de temps pour le contenu définie à titre privé. Une valeur de codage '0' indique qu'aucune base de temps pour le contenu n'est définie dans ce descripteur. Si aucune base de temps pour le contenu n'est définie pour un programme ou un flux, alors le mappage de références temporelles dans les métadonnées sur le contenu n'est pas défini dans la présente Spécification.

Tableau 2-82 – Valeurs du champ Content_time_base_indicator

Valeur	Description
0	Aucune base de temps pour le contenu n'est définie dans ce descripteur
1	Utilisation de l'horloge STC
2	Utilisation de la durée de lecture normale (NPT)
3-7	Valeurs réservées
8-15	Utilisation d'une base de temps pour le contenu définie à titre privé

content_reference_id_record_length (longueur d'enregistrement d'identificateur de référence de contenu) – Codé sur 8 bits, ce champ indique le nombre d'octets content_reference_id_bytes suivant immédiatement ce champ. La valeur '0' ne doit pas être utilisée pour le codage de ce champ.

content_reference_id_byte (octet d'identificateur de référence de contenu) – Partie d'une chaîne d'un ou de plusieurs octets contigus qui attribue une ou plusieurs identifications (étiquettes) de référence au contenu auquel ce descripteur est associé. Le format de la chaîne d'octets est défini par le corps indiqué par la valeur codée dans le champ metadata_application_format.

content_time_base_value (valeur de base de temps pour le contenu) – Codé sur 33 bits, ce champ indique une valeur en unités de 90 kHz pour la base de temps pour le contenu indiquée par le champ content_time_base_indicator.

metadata_time_base_value (valeur de base de temps pour les métadonnées) – Codé en unités de 90 kHz sur 33 bits, ce champ reprend la valeur de la base de temps pour les métadonnées à l'instant auquel la base de temps indiquée par le

champ content_time_base_indicator atteint la valeur codée dans le champ content_time_base_value. A noter que la base de temps pour les métadonnées peut utiliser n'importe quelle échelle de temps, mais que sa valeur doit être codée en unités de 90 kHz. Par exemple, en cas d'utilisation d'un code temporel de type SMPTE, le nombre d'heures, de minutes, de secondes et de trames est exprimé en autant d'unités de 90 kHz correspondantes.

contentId (identificateur de contenu) – Codé sur 7 bits, ce champ indique la valeur du champ content_Id du descripteur de référence pour la base de temps NPT appliquée.

time_base_association_data_length (longueur des données d'association de base de temps) – Codé sur 8 bits, ce champ indique le nombre d'octets réservés qui le suivent immédiatement. Les octets réservés peuvent être utilisés pour acheminer des données d'association de base de temps pour des bases de temps qui seront définies ultérieurement.

private_data_byte (octet de données privées) – Codé sur 8 bits, ce champ représente des données dont le format est défini à titre privé. Ces octets peuvent être utilisés pour fournir des informations complémentaires jugées nécessaires. L'utilisation de ces octets est définie par le format d'application des métadonnées.

2.6.58 Descripteur de pointeur de métadonnées

Le descripteur de pointeur de métadonnées signale un service de métadonnées isolé et associe ce service à un contenu audiovisuel dans un flux défini dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Les métadonnées sont associées au contenu dans le contexte du descripteur. Le contexte est défini par l'emplacement du descripteur. Dans un flux de transport, le descripteur peut être situé dans la table de mappage de programme (PMT, *program map table*) dans la boucle de descripteur pour le programme ou pour un flux élémentaire, mais il peut aussi figurer dans des tableaux non définis dans la présente Spécification, tels que les tableaux décrivant des bouquets de services de diffusion. Les métadonnées peuvent figurer dans un flux UIT-T H.222.0 | ISO/CEI 13818-1, mais les mêmes métadonnées peuvent aussi figurer ailleurs en d'autres emplacements, par exemple sur l'Internet.

Le descripteur peut contenir l'information de position des métadonnées qui ne sont pas acheminées dans un flux UIT-T H.222.0 | ISO/CEI 13818-1; le codage de l'information de position est propre au format d'application des métadonnées. Le descripteur permet d'acheminer des données privées.

Pour des métadonnées acheminées dans un flux UIT-T H.222.0 | ISO/CEI 13818-1, le descripteur indique les outils utilisés pour acheminer lesdites métadonnées. Si celles-ci sont acheminées dans des paquets PES, dans des sections de métadonnées ou dans des sections de téléchargement synchronisé ISO/CEI 13818-6, le champ metadata_service_id identifie le service de métadonnées dans le flux de métadonnées référencé. Si un carrousel ISO/CEI 13818-6 est utilisé pour acheminer les métadonnées, les données privées peuvent alors fournir des informations pour signaler le service de métadonnées, telles que la valeur appliquée de l'identificateur de module (module_id) aux fins de l'acheminement des métadonnées dans un carrousel de données, ainsi que le nom du fichier des métadonnées en cas d'utilisation du carrousel d'objets.

Les destinataires doivent être conscients que plusieurs services de métadonnées peuvent être signalés par un même flux de programmes ou audiovisuel (tel que défini par le contexte du descripteur). Un descripteur de pointeur de métadonnées isolé doit être utilisé pour signaler chaque service de métadonnées utilisé par le flux de programmes ou audiovisuel. De même, le même service de métadonnées peut être signalé par plusieurs flux de programmes ou audiovisuels, moyennant l'utilisation d'un descripteur de pointeur de métadonnées distinct pour chaque association.

Tableau 2-83 – Descripteur de pointeur de métadonnées

Syntaxe	Nombre de bits	Mnémonique
Metadata_pointer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	**32**	**uimsbf**
}		
metadata_format	**8**	**uimsbf**
if (metadata_format== 0xFF){		
metadata_format_identifier	**32**	**uimsbf**
}		
metadata_service_id	**8**	**uimsbf**
metadata_locator_record_flag	**1**	**bslbf**
MPEG_carriage_flags	**2**	**uimsbf**
reserved	**5**	**bslbf**
if (metadata_locator_record_flag == '1'){		
metadata_locator_record_length	**8**	**uimsbf**
for (i=0; i< metadata_locator_record_length;i++){		
metadata_locator_record_byte	**8**	**bslbf**
}		
}		
if (MPEG_carriage_flags == 0|1|2){		
program_number	**16**	**uimsbf**
}		
if (MPEG_carriage_flags == 1){		
transport_stream_location	**16**	**uimsbf uimsbf**
transport_stream_id	**16**	
}		
for (i=0; i<N;i++){		**bslbf**
private_data_byte	**8**	
}		
}		

2.6.59 Définition sémantique des champs du descripteur de pointeur de métadonnées

metadata_application_format (format d'application des métadonnées) – Codé sur 16 bits, ce champ indique l'application qui doit définir l'usage, la syntaxe et la sémantique de l'enregistrement metadata_locator_record et de tout autre champ de ce descripteur défini à titre privé. Le codage de ce champ est défini dans le Tableau 2-81 au 2.6.57.

metadata_application_format_identifier (identificateur de format d'application de métadonnées) – Le codage de ce champ est défini au 2.6.57.

metadata_format (format de métadonnées) – Codé sur 8 bits, ce champ indique le format et le codage des métadonnées. Le codage de ce champ est défini dans le Tableau 2-84.

Tableau 2-84 – Valeurs du format de métadonnées

Valeur	Description
0x00-0x0F	Valeurs réservées
0x10	ISO/CEI 15938-1 TeM
0x11	ISO/CEI 15938-1 BiM
0x12-0x3E	Valeurs réservées
0x3F	Définie par le format d'application des métadonnées
0x40-0xFE	Usage privé
0xFF	Définie par le champ metadata_format_identifier

Les valeurs 0x10 et 0x11 identifient des données définies dans l'ISO/CEI 15938-1. La valeur 0x3F indique que le format est défini par le corps indiqué par le champ metadata_application_format. Les valeurs de 0x40 à 0xFE peuvent servir à signaler l'utilisation de formats privés. La valeur 0xFF indique que le format est signalé par le champ metadata_format_identifier.

metadata_format_identifier (identificateur de format de métadonnées) – Le codage de ce champ de 32 bits est entièrement équivalent à celui du champ format_identifier du descripteur d'enregistrement (registration_descriptor) défini au 2.6.8.

NOTE – La SMPTE est désignée comme organisme d'enregistrement pour le champ format_identifier.

metadata_service_id (identificateur de service de métadonnées) – Codé sur 8 bits, ce champ désigne le service de métadonnées. Il est utilisé pour extraire un service de métadonnées d'un flux de métadonnées.

metadata_locator_record_flag (fanion d'enregistrement de localisateur de métadonnées) – Lorsqu'il est mis à '1', ce champ codé sur 1 bit indique la présence de métadonnées associées en un autre emplacement que dans un flux UIT-T H.222.0 | ISO/CEI 13818-1 spécifié dans un enregistrement metadata_locator_record.

MPEG_carriage_flags (fanions d'acheminement MPEG) – Codé sur 2 bits, ce champ indique si le flux de métadonnées contenant le service de métadonnées associé est acheminé dans un flux UIT-T H.222.0 | ISO/IEC 13818-1 et, si tel est le cas, si les métadonnées associées sont acheminées dans un flux de transport ou dans un flux de programme. Le codage de ce champ est défini dans le Tableau 2-85.

Tableau 2-85 – MPEG_carriage_flags

Valeur	Description
0	Acheminement dans le même flux de transport que celui dans lequel ce descripteur de pointeur de métadonnées est acheminé.
1	Acheminement dans un flux de transport autre que celui dans lequel ce descripteur de pointeur de métadonnées est acheminé.
2	Acheminement dans un flux de programme. Il peut s'agir ou non du même flux de programme que celui dans lequel ce descripteur de pointeur de métadonnées est acheminé.
3	Autre mode d'acheminement.

metadata_locator_record_length (longueur d'enregistrement de localisateur de métadonnées) – Codé sur 8 bits, ce champ indique le nombre d'octets metadata_locator_record_bytes qui suivent immédiatement. La valeur '0' ne doit pas être utilisée pour le codage de ce champ.

metadata_locator_record_byte (octet d'enregistrement de localisateur de métadonnées) – Partie d'une chaîne d'un ou de plusieurs octets contigus formant l'enregistrement du localisateur des métadonnées. Cet enregistrement indique un ou plusieurs emplacements extérieurs à un flux UIT-T H.222.0 | ISO/CEI 13818-1. Le format de l'enregistrement du localisateur de métadonnées est défini par l'application des métadonnées signalée par le champ metadata_application_format. L'enregistrement peut par exemple contenir des localisateurs URL indiquant où se trouvent les métadonnées et, éventuellement, leur(s) position(s) dans le flux de transport. L'utilisation de la valeur '0', '1' ou '2' pour coder le champ MPEG_carriage_flags et la présence de l'enregistrement du localisateur de métadonnées sont le signe qu'il existe d'autres emplacements pour les mêmes métadonnées.

program_number (numéro de programme) – Codé sur 16 bits, ce champ identifie le numéro (program_number) du programme MPEG-2 figurant dans le flux UIT-T H.222.0 | ISO/CEI 13818-1 dans lequel les métadonnées associées sont acheminées. Si le champ MPEG_carriage_flags a la valeur '0', le flux de transport est celui qui est utilisé au moment considéré; si ce champ a la valeur '1', le flux de transport est celui qui est signalé par les champs transport_stream_location et transport_stream_id.

transport_stream_location (emplacement du flux de transport) – Codé sur 16 bits, ce champ est défini à titre privé. Dans certaines applications, il peut être utilisé, par exemple, pour signaler le champ original_network_id défini par l'ETSI.

transport_stream_id (identificateur de flux de transport) – Codé sur 16 bits, ce champ identifie le flux de transport dans lequel les métadonnées associées sont acheminées.

private_data_byte (octet de données privées) – Codé sur 8 bits, ce champ représente des données, dont le format est défini à titre privé. Les octets de ce champ peuvent être utilisés pour fournir des informations supplémentaires jugées nécessaires.

2.6.60 Descripteur de métadonnées

Le descripteur de métadonnées spécifie les paramètres d'un service de métadonnées acheminées dans un flux de transport (TS) ou dans un flux de programme (PS) MPEG-2. Dans un flux TS MPEG-2, le descripteur est inclus dans la table PMT dans la boucle de descripteur pour le flux élémentaire qui achemine le service de métadonnées. Le descripteur spécifie le format des métadonnées associées, et contient la valeur du champ metadata_service_id destiné à identifier le service de métadonnées auquel le descripteur de métadonnées s'applique. Si nécessaire, le descripteur peut véhiculer des informations destinées à identifier le service de métadonnées d'après un ensemble de métadonnées transmises dans un carrousel DSM-CC. Des données privées propres au format d'application des métadonnées peuvent éventuellement être acheminées.

Le descripteur de métadonnées indique également si la configuration du décodeur est nécessaire et si elle permet d'acheminer les octets de configuration du décodeur; toutefois, dans la pratique, cela n'est utile que si ces octets sont peu nombreux. Si les informations de configuration du décodeur sont trop volumineuses pour être acheminées par le descripteur, elles doivent être incluses dans un service de métadonnées, qu'il s'agisse du service de métadonnées considéré ou d'un autre service de métadonnées du même programme. L'identification du service de métadonnées qui contient la configuration du décodeur est assurée par le descripteur de métadonnées. En cas d'utilisation d'un carrousel DSM-CC pour acheminer l'information de configuration du décodeur, la procédure à suivre pour extraire cette information du carrousel pourra être indiquée.

Tableau 2-86 – Descripteur de métadonnées

Syntaxe	Nombre de bits	Mnémonique
Metadata_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format == 0xFFFF) {		
metadata_application_format_identifier	**32**	**uimsbf**
}		
metadata_format	**8**	**uimsbf**
if (metadata_format== 0xFF){		
metadata_format_identifier	**32**	**uimsbf**
}		
metadata_service_id	**8**	**uimsbf**
decoder_config_flags	**3**	**bslbf**
DSM-CC_flag	**1**	**bslbf**
reserved	**4**	**bslbf**
if (DSM-CC_flag == '1'){		
service_identification_length	**8**	**uimsbf**
for(i=0; i<service_identification_length; i++) {		
service_identification_record_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '001') {		
decoder_config_length	**8**	**uimsbf**
for(i=0; i<decoder_config_length; i++) {		
decoder_config_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '011') {		
dec_config_identification_record_length	**8**	**uimsbf**
for(i=0;i<dec_config_id_record_length;i++) {		
dec_config_identification_record_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '100') {		
decoder_config_metadata_service_id	**8**	**uimsbf**
}		
if (decoder_config_flags == '101'|'110') {		
reserved_data_length	**8**	**uimsbf**
for(i=0;i<reserved_data_length;i++) {		
reserved	**8**	**bslbf**
}		
}		
for (i=0; i<N;i++) {		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.61 Définition sémantique des champs du descripteur de métadonnées

metadata_application_format (format d'application des métadonnées) – Codé sur 16 bits, ce champ indique l'application qui doit définir l'usage, la syntaxe et la sémantique du champ service_identification_record et d'éventuels octets de ce descripteur définis à titre privé. Le codage de ce champ est défini dans le Tableau 2-81.

metadata_application_format_identifier (identificateur du format d'application des métadonnées) – Le codage de ce champ est défini au 2.6.57.

metadata_format (format des métadonnées) – Le codage de champ est défini au 2.6.59.

metadata_format_identifier (identificateur de format de métadonnées) – Le codage de ce champ est défini au 2.6.59.

metadata_service_id (identificateur de service de métadonnées) – Codé sur 8 bits, ce champ identifie le service de métadonnées auquel ce descripteur de métadonnées s'applique.

decoder_config_flags (fanions de configuration de décodeur) – Codé sur 3 bits, ce champ indique si l'information de configuration du décodeur est véhiculée, et dans l'affirmative, de quelle manière.

Tableau 2-87 – decoder_config_flags

Valeur	Description
000	Aucune configuration de décodeur n'est nécessaire.
001	La configuration du décodeur est acheminée dans le champ decoder_config_byte de ce descripteur.
010	La configuration du décodeur est acheminée dans le même service de métadonnées que celui auquel ce descripteur de métadonnées s'applique.
011	La configuration du décodeur est acheminée dans un carrousel DSM-CC. Cette valeur ne doit être utilisée que si le service de métadonnées auquel ce descripteur s'applique utilise le même type de carrousel DSM-CC.
100	La configuration du décodeur est acheminée dans un autre service de métadonnées du même programme, défini dans le champ decoder_config_metadata_service_id de ce descripteur de métadonnées.
101, 110	Valeurs réservées.
111	Valeur définie à titre privé.

DSM-CC_flag (fanion DSM-CC) – Codé sur 1 bit, ce fanion est mis à '1' si le flux auquel ce descripteur est associé est acheminé dans un carrousel de données ou d'objets ISO/CEI 13818-6.

NOTE 1 – Le type de carrousel utilisé (carrousel d'objets ou carrousel de données) est indiqué par la valeur type de flux appliquée à ce flux de métadonnées.

service_identification length (longueur d'identification du service) – Ce champ indique le nombre d'octets service_identification_record_bytes qui suivent immédiatement.

service_identification_record_byte (octet d'enregistrement d'identification de service) – Octet faisant partie d'une chaîne d'un ou de plusieurs octets contigus définissant l'enregistrement service_identification_record. Cet enregistrement contient des données indiquant comment extraire le service de métadonnées d'un carrousel DSM-CC. Le format de l'enregistrement du localisateur des métadonnées est défini par l'application indiquée dans le format d'application des métadonnées. En cas d'utilisation d'un carrousel d'objets DSM-CC, l'enregistrement peut comprendre, par exemple, le seul identificateur d'objet (IOP:IOR() défini aux 11.3.1 et 5.7.2.3 de l'ISO/CEI 13818-6 DSM-CC) correspondant au service de métadonnées. De même, en cas d'utilisation d'un carrousel de données DSM-CC, l'enregistrement peut indiquer, par exemple, les champs transaction_id et module_id du service de métadonnées.

decoder_config_length (longueur de configuration de décodeur) – Ce champ indique le nombre d'octets decoder_config_bytes qui suivent immédiatement.

decoder_config_byte (octet de configuration de décodeur) – Octet comprenant l'information de configuration du décodeur. Une séquence de plusieurs octets de ce type comprend l'information de configuration dont le récepteur a besoin pour décoder ce service. Il est prévu de ne recourir à l'acheminement dans le descripteur de métadonnées que lorsque l'information de configuration est très réduite.

decoder_config_DSM-CC_id (identificateur de commande DSM-CC de configuration de décodeur) – Il s'agit de l'identificateur de téléchargement de l'information de configuration du décodeur lorsque celle-ci est transmise dans un carrousel de données DSM-CC, ou de l'identificateur d'objets de l'information de configuration du décodeur au cas où celle-ci est acheminée dans un carrousel d'objets DSM-CC.

NOTE 2 – Le type de carrousel utilisé (carrousel d'objets ou carrousel de données) est indiqué par la valeur de type de flux appliquée à ce flux de métadonnées.

dec_config_identification_record_length (longueur d'enregistrement d'identification de configuration de décodeur) – Ce champ indique le nombre d'octets dec_config_identification_record_bytes qui suivent immédiatement.

dec_config_identification_record_byte (octet d'enregistrement d'identification de configuration de décodeur) – Cet octet fait partie d'une chaîne d'un ou de plusieurs octets contigus définissant l'enregistrement dec_config_identification_record. Cet enregistrement indique comment extraire la configuration de décodeur voulue d'un carrousel DSM-CC. Le format de l'enregistrement du localisateur de métadonnées est défini par le format d'application des métadonnées. En cas d'utilisation d'un carrousel d'objets DSM-CC, l'enregistrement peut comprendre, par exemple, le seul identificateur d'objet (IOP:IOR() défini aux 11.3.1 et 5.7.2.3 de l'ISO/CEI 13818-6 DSM-CC) correspondant à la configuration du décodeur. De même, en cas d'utilisation d'un carrousel de données DSM-CC, l'enregistrement peut indiquer, par exemple, les champs transaction_id et module_id de la configuration du décodeur.

decoder_config_metadata_service_id (identificateur de service de métadonnées de configuration de décodeur) – Il s'agit de la valeur de l'identificateur metadata_service_id qui est attribuée au service de métadonnées qui contient la configuration du décodeur. Le service de métadonnées indiqué par l'identificateur decoder_config_metadata_service_id et le service de métadonnées qui utilise cette configuration de décodeur doivent être dans le même programme. Ainsi, dans un flux de transport, les descripteurs de métadonnées de ces deux services de métadonnées doivent figurer dans la même table PMT. Le descripteur de métadonnées du service de métadonnées indiqué par l'identificateur decoder_config_metadata_service_id doit avoir un champ decoder_config_flag ayant pour valeur '001', '010' ou '011'.

reserved_data_length (longueur de données réservées) – Ce champ indique le nombre d'octets réservés qui suivent immédiatement.

private_data_byte (octet de données privées) – Codé sur 8 bits, ce champ représente des données dont le format est défini à titre privé. Des octets de ce type peuvent être utilisés pour fournir des informations supplémentaires jugées nécessaires.

2.6.62 Descripteur STD de métadonnées

Ce descripteur définit les paramètres du modèle STD (défini au 2.12.10) pour le traitement du flux de métadonnées auquel ce descripteur est associé.

Tableau 2-88 – Descripteur STD de métadonnées

Syntaxe	Nombre de bits	Mnémonique
Metadata_STD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**2**	**bslbf**
metadata_input_leak_rate	**22**	**uimsbf**
reserved	**2**	**bslbf**
metadata_buffer_size	**22**	**uimsbf**
reserved	**2**	**bslbf**
metadata_output_leak_rate	**22**	**uimsbf**
}		

2.6.63 Définition sémantique des champs du descripteur STD de métadonnées

metadata_input_leak_rate (débit de débordement d'entrée de métadonnées) – Codé sur 22 bits, ce champ indique le débit de débordement du flux de métadonnées associé dans le modèle T-STD, depuis la mémoire tampon TB_n dans la mémoire tampon B_n. Le débit de débordement est exprimé en unités de 400 bits/s. Pour les métadonnées acheminées dans un flux de programme, le codage du champ metadata_input_leak_rate n'est pas précisé, du fait que le débit de débordement dans la mémoire tampon B_n est égal au débit du flux de programme.

metadata_buffer_size (capacité de la mémoire tampon de métadonnées) – Codé sur 22 bits, ce champ indique la capacité de la mémoire tampon B_n dans le modèle STD pour le flux de métadonnées associé. La capacité de la mémoire tampon B_n est exprimée en unités de 1024 octets.

metadata_output_leak_rate (débit de débordement de sortie de métadonnées) – Codé sur 22 bits, ce champ indique pour le service de métadonnées associé le débit de débordement dans le modèle STD, depuis la mémoire tampon B_n dans le décodeur. Le débit de débordement est exprimé en unités de 400 bit/s. Pour les flux de métadonnées transportées en mode synchrone (type de flux 0x15 ou 0x19), les unités d'accès aux métadonnées sont instantanément supprimées de la mémoire tampon B_n suivant les instructions de l'horodateur de présentation (PTS, *presentation timestamps*) et, dans ce cas, le codage du champ metadata_output_leak_rate n'est pas précisé.

2.6.64 Descripteur vidéo AVC

Dans le cas des flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10, le descripteur vidéo AVC donne des informations de base pour identifier les paramètres de codage du flux vidéo AVC associé, par exemple des informations sur les paramètres de profil et de niveau contenus dans l'ensemble SPS d'un flux vidéo AVC.

Le descripteur vidéo AVC signale également la présence d'images fixes AVC et celle d'images 24 heures AVC dans le flux vidéo AVC. Si ce descripteur n'est pas inclus dans la table PMT pour un flux vidéo AVC d'un flux de transport ou dans le mappage PSM, s'il est présent, pour un flux vidéo AVC d'un flux de programme, le flux vidéo AVC ne doit contenir ni image fixe AVC ni image 24 heures AVC (voir Tableau 2-89).

Tableau 2-89 – Descripteur vidéo AVC

Syntaxe	Nbre de bits	Mnémonique
AVC_video_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
profile_idc	**8**	**uimsbf**
constraint_set0_flag	**1**	**bslbf**
constraint_set1_flag	**1**	**bslbf**
constraint_set2_flag	**1**	**bslbf**
AVC_compatible_flags	**5**	**bslbf**
level_idc	**8**	**uimsbf**
AVC_still_present	**1**	**bslbf**
AVC_24_hour_picture_flag	**1**	**bslbf**
reserved	**6**	**bslbf**
}		

2.6.65 Définition sémantique des champs du descripteur vidéo AVC

profile_idc, constraint_set0_flag, **constraint_set1_flag**, **constraint_set2_flag, AVC_compatible_flags et level_idc** – Ces champs, à l'exception de AVC_compatible_flags, seront codés conformément à la sémantique de ces champs définie dans la Rec. UIT-T H.264 | ISO/CEI 14496-10. La sémantique du champ AVC_compatible_flags est exactement la même que la sémantique du ou des champs définis pour les 5 bits compris entre les champs constraint_set2_flag et level_idc de l'ensemble de paramètres de séquence, définie dans la Rec. UIT-T H.264 | ISO/CEI 14496-10. L'ensemble du flux vidéo AVC auquel est associé le descripteur AVC doit être conforme au profil, au niveau et aux contraintes signalés par ces champs.

NOTE – Dans une ou plusieurs séquences du flux vidéo AVC, le niveau peut être inférieur au niveau signalé dans le descripteur vidéo AVC et le profil peut être un sous-ensemble du profil signalé dans le descripteur vidéo AVC. Toutefois, dans l'ensemble du flux vidéo AVC, il ne faut utiliser que les outils inclus dans le profil signalé dans le descripteur vidéo AVC, s'il est présent. Par exemple, si le profil principal est signalé, on peut utiliser le profil de base dans certaines séquences, mais en n'utilisant que les outils contenus dans le profil principal. Si les ensembles de paramètres de séquence d'un flux vidéo AVC signalent des profils différents, et qu'aucune contrainte additionnelle n'est signalée, il se peut que le flux doive être examiné pour déterminer à quel profil, si profil il y a, l'ensemble du flux est conforme. Si un descripteur vidéo AVC doit être associé à un flux vidéo AVC qui n'est pas conforme à un profil unique, le flux vidéo AVC doit être subdivisé en deux sous-flux ou plus afin que les descripteurs vidéo AVC puissent signaler un profil unique pour chaque sous-flux.

AVC_still_present – La valeur '1' de ce champ de 1 bit indique que le flux vidéo AVC peut contenir des images fixes AVC. La valeur '0' indique que le flux vidéo AVC associé ne doit pas contenir d'image fixe AVC.

AVC_24_hour_picture_flag – La valeur '1' de ce champ de 1 bit indique que le flux vidéo AVC associé peut contenir des images 24 heures AVC. La définition de l'image 24 heures AVC est donnée au 2.1.2. La valeur '0' indique que le flux vidéo AVC associé ne doit pas contenir d'image 24 heures AVC.

2.6.66 Descripteur de décodeur HRD et de synchronisation AVC

Le descripteur de décodeur HRD et de synchronisation AVC fournit les paramètres de décodeur HRD et de synchronisation du flux vidéo AVC associé. Pour chaque flux vidéo AVC acheminé dans un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1, le descripteur de décodeur HRD et de synchronisation AVC doit être inclus dans la table PMT ou dans le mappage PSM, si ce mappage PSM est présent dans le flux de programme, à moins que le flux vidéo AVC achemine des paramètres VUI avec le fanion timing_info_present_flag mis à '1':

- pour chaque image IDR; et
- pour chaque image associée à un message SEI de point de rétablissement.

L'absence du descripteur de décodeur HRD et de synchronisation AVC de la table PMT pour un flux vidéo AVC signale que le décodeur T-STD utilise la méthode avec fuite définie au 2.14.3.1 pour le transfert du tampon MB_n au tampon EB_n, mais une telle utilisation peut aussi être signalée par le fanion hrd_management_valid_flag mis à '0' dans le descripteur de décodeur HRD et de synchronisation AVC. Si le débit de transfert dans le tampon EB_n peut être déterminé à partir des paramètres de décodeur HRD contenus dans un flux vidéo AVC et si ce débit de transfert est utilisé dans le décodeur T-STD pour le transfert de MB_n à EB_n, le descripteur de décodeur HRD et de synchronisation AVC avec le fanion hrd_management_valid_flag mis à '1' doit être inclus dans la table PMT pour ce flux vidéo AVC (voir Tableau 2-90).

Tableau 2-90 – Descripteur de décodeur HRD et de synchronisation AVC

Syntaxe	Nbre de bits	Mnémonique
AVC timing and HRD descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
hrd_management_valid_flag	**1**	**bslbf**
reserved	**6**	**bslbf**
picture_and_timing_info_present	**1**	**bslbf**
if (picture_and_timing_info_present) {		
90kHz_flag	**1**	**bslbf**
reserved	**7**	**bslbf**
if (90kHz_flag == '0') {		
N	**32**	**uimsbf**
K	**32**	**uimsbf**
}		
num_units_in_tick	**32**	**uimsbf**
}		
fixed_frame_rate_flag	**1**	**bslbf**
temporal_poc_flag	**1**	**bslbf**
picture_to_display_conversion_flag	**1**	**bslbf**
reserved	**5**	**bslbf**
}		

2.6.67 Définition sémantique des champs du descripteur de décodeur HRD et de synchronisation AVC

hrd_management_valid_flag – Ce champ de 1 bit est uniquement destiné à être utilisé dans les flux de transport.

Lorsque le descripteur de décodeur HRD et de synchronisation AVC est associé à un flux vidéo AVC acheminé dans un flux de transport, on procède comme suit. Si le fanion hrd_management_valid_flag est mis à '1', les messages SEI de période de mise en tampon et de synchronisation d'image, définis dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10, seront présents dans le flux vidéo AVC associé. Les messages SEI de période de mise en tampon contiendront les valeurs codées de initial_cpb_removal_delay et de initial_cpb_removal_delay_offset pour le décodeur NAL-HRD. Si le fanion hrd_management_valid_flag est mis à '1', le transfert de chaque octet de MB_n à EB_n dans le décodeur T-STD sera conforme au programme de remise de cet octet dans le tampon CPB du décodeur NAL-HRD, comme cela est déterminé à partir des valeurs codées de initial_cpb_removal_delay et de initial_cpb_removal_delay_offset pour SchedSelIdx = cpb_cnt_minus1. Lorsque le fanion hrd_management_valid_flag est mis à '0', le décodeur T-STD utilisera la méthode avec fuite définie au 2.14.3.1 pour le transfert de MB_n à EB_n.

Lorsque le descripteur de décodeur HRD et de synchronisation AVC est associé à un flux vidéo AVC acheminé dans un flux de programme, le sens du fanion hrd_management_valid_flag n'est pas défini.

picture_and_timing_info_present – Ce champ de 1 bit mis à '1' indique que le fanion 90kHz_flag et les paramètres permettant d'assurer un mappage précis avec une horloge système à 90 kHz sont contenus dans ce descripteur.

90kHz_flag, N, K – Le fanion 90kHz_flag mis à '1' indique que la fréquence de la base de temps AVC est de 90 kHz. Pour un flux vidéo AVC, la fréquence de la base de temps AVC est définie par le paramètre AVC time_scale parmi les paramètres VUI, définis dans l'Annexe E de la Rec. UIT-T H.264 | ISO/CEI 14496-10. La relation entre time_scale AVC et l'horloge STC sera définie comme suit avec les paramètres N et K de ce descripteur:

$$time_scale = \frac{(N \times system_clock_frequency)}{K}$$

où time_scale désigne la fréquence exacte de la base de temps AVC, K étant supérieur ou égal à N.

Si le fanion 90kHz_flag est mis à '1', N est égal à 1 et K est égal à 300. Si le fanion 90kHz_flag est mis à '0', les valeurs de N et de K sont données par les valeurs codées des champs N et K.

NOTE 1 – Cela permet de mapper les instants exprimés dans l'unité de time_scale en instants exprimés dans l'unité valant 90 kHz, comme cela est requis pour le calcul des horodates PTS et DTS, par exemple dans des décodeurs pour unités d'accès AVC pour lesquels aucune horodate PTS ou DTS n'est codée dans l'en-tête PES.

num_units_in_tick – Codé exactement de la même manière que le champ num_units_in_tick figurant dans les paramètres VUI de l'Annexe E de la Rec. UIT-T H.264 | ISO/CEI 14496-10. L'information donnée par ce champ s'appliquera à l'ensemble du flux vidéo AVC auquel le descripteur de décodeur HRD et de synchronisation AVC est associé.

fixed_frame_rate_flag – Codé exactement de la même manière que le fanion fixed_frame_rate_flag figurant dans les paramètres VUI de l'Annexe E de la Rec. UIT-T H.264 | ISO/CEI 14496-10. Lorsque ce fanion est mis à '1', il indique que la fréquence d'image codée est constante dans le flux vidéo AVC associé. Lorsque ce fanion est mis à '0', aucune information de fréquence d'image du flux vidéo AVC associé n'est donnée dans ce descripteur.

temporal_poc_flag – Lorsque le fanion temporal_poc_flag est mis à '1' et que le fanion fixed_frame_rate_flag est mis à '1', le flux vidéo AVC associé achemine l'information de numéro d'ordre d'image (POC, *PicOrderCnt*), ce numéro correspondant au nombre d'intervalles $\Delta t_{fi,dpb}(n)$, où $\Delta t_{fi,dpb}(n)$ est spécifié dans l'équation E-10 de la Rec. UIT-T H.264 | ISO/CEI 14496-10. Lorsque le fanion temporal_poc_flag est mis à '0', aucune information au sujet d'une relation possible entre l'information de POC dans le flux vidéo AVC et le temps n'est acheminée.

NOTE 2 – Cela réduit le préfixe nécessaire pour signaler la synchronisation de chaque unité d'accès. Des horodates PTS et DTS équivalentes peuvent être calculées pour des unités d'accès pour lesquelles aucune horodate PTS/DTS explicite n'est acheminée. La répétition du champ le plus récemment présenté de la parité (ou image) appropriée est sous-entendue lorsque la différence entre les horodates PTS de l'image considérée et de l'image suivante est supérieure à $2 \times \Delta t_{fi,dpb}$ (ou supérieure à $\Delta t_{fi,dpb}$ quand *frame_mbs_only_flag* est égal à 1).

picture_to_display_conversion_flag – Ce champ de 1 bit mis à '1' indique que le flux vidéo AVC associé peut acheminer des informations d'affichage relatives aux images codées en fournissant le champ pic_struct des messages SEI de synchronisation d'image (voir Annexe D de la Rec. UIT-T H.264 | ISO/CEI 14496-10) et/ou en donnant l'information de numéro d'ordre d'image (POC), ce numéro correspondant au nombre d'intervalles $\Delta t_{fi,dpb}(n)$ (voir aussi la définition sémantique de temporal_poc_flag), afin que l'information de synchronisation relative à une unité d'accès AVC suivante puisse être déterminée à partir de l'image précédente dans l'ordre de décodage ou de présentation.

Lorsque le fanion picture_to_display_conversion_mode_flag est mis à '0', les messages SEI de synchronisation d'image du flux vidéo AVC, s'ils sont présents, ne contiennent pas de champ pic_struct, et dès lors le fanion pic_struct_present_flag sera mis à '0' dans les paramètres VUI du flux vidéo AVC.

2.6.68 Descripteur MPEG-2 AAC audio

En ce qui concerne les flux distincts ISO/CEI 13818-7 directement acheminés dans des paquets PES, le descripteur MPEG-2_AAC_audio défini dans le Tableau 2-91 fournit des renseignements de base pour l'identification des paramètres de codage de ces flux élémentaires audio.

Tableau 2-91 – Descripteur MPEG-2_AAC_audio_descriptor

Syntaxe	Nombre de bits	Mnémonique
MPEG-2_AAC_audio_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-2_AAC_profile	**8**	**uimsbf**
MPEG-2_AAC_channel_configuration	**8**	**uimsbf**
MPEG-2_AAC_additional_information	**8**	**uimsbf**
}		

2.6.69 Définition sémantique des champs du descripteur MPEG-2 AAC audio

MPEG-2_AAC_profile – (profil de codage AAC MPEG-2) – Codé sur 8 bits, ce champ indique le profil de codage AAC conforme à l'indice du Tableau 31 de l'ISO/CEI 13818-7:2006.

MPEG-2_AAC_channel_configuration – (configuration de voie de codage AAC MPEG-2) – Codé sur 8 bits, ce champ indique le nombre et la configuration des voies audio offertes à l'auditeur par le décodeur AAC pour le programme considéré. Les valeurs de 1 à 6 indiquent le nombre et la configuration des voies audio indiqués pour le "nombre d'indices de flux binaire par défaut" dans le Tableau 42 de l'ISO/CEI 13818-7:2006. Toutes les autres valeurs indiquent que le nombre et la configuration des voies audio ne sont pas définis.

MPEG-2_AAC_additional_information – (informations additionnelles de codage AAC MPEG-2) – Codé sur 8 bits, ce champ indique si les données d'extension de largeur de bande définies dans la Norme ISO/CEI 13818-7:2006 sont ou non intégrées dans le flux binaire AAC comme indiqué dans le Tableau 2-92.

Tableau 2-92 – Valeurs du champ MPEG-2_AAC_additional_information

Valeur	Description
0x00	Données AAC conformes à la Norme ISO/CEI 13818-7:2006
0x01	Données AAC avec présence de données d'extension de largeur de bande conformément à la Norme ISO/CEI 13818-7:2006
0x02-0xFF	Réservé

2.7 Restrictions appliquées à la sémantique des flux binaires multiplexés

2.7.1 Fréquence de codage de la référence temporelle du système

Le flux de programme doit être construit de manière que l'intervalle de temps entre les octets contenant le dernier bit des champs de type system_clock_reference_base contenus dans des blocs de paquets successifs soit inférieur ou égal à 0,7 s. Autrement dit:

$$\left| t(i) - t(i') \right| \leq 0{,}7 \text{ s}$$

pour tous les i et pour tous les i′, où i et i′ désignent les indices des octets contenant le dernier bit des champs consécutifs de type system_clock_reference_base.

2.7.2 Fréquence de codage de la référence temporelle de programme

Le flux de transport doit être construit de manière que l'intervalle de temps entre les octets contenant le dernier bit des champs de type program_clock_reference_base des occurrences successives de champs PCR contenus dans des paquets du flux de transport et désignés par le champ PCR_PID pour chaque programme, soit inférieur ou égal à 0,1 s. Autrement dit:

$$\left| t(i) - t(i') \right| \leq 0{,}1 \text{ s}$$

pour tous les i et pour tous les i′, où i et i′ désignent les indices des octets contenant le dernier bit des champs consécutifs de type program_clock_reference_base, contenus dans les paquets de transport désignés par l'identificateur PCR_PID de chaque programme.

Il doit y avoir au moins deux (2) références temporelles de programme correspondant aux valeurs spécifiées du champ PCR_PID à l'intérieur d'un flux de transport, entre des discontinuités consécutives de référence temporelle de programme (voir 2.4.3.4), afin de faciliter le verrouillage de phase et l'extrapolation des instants d'injection des octets.

2.7.3 Fréquence de codage de la référence temporelle d'un flux élémentaire

Les flux de programme et de transport doivent être construits de telle manière que, si le champ de référence temporelle de flux élémentaire est codé dans un quelconque paquet PES contenant des données d'un flux élémentaire déterminé, l'intervalle de temps entre les octets contenant le dernier bit de champs successifs de type ESCR_base soit inférieur ou égal à 0,7 s. Dans les flux de type PES, le codage du champ ESCR est requis avec le même intervalle. On a donc:

$$\left| t(i) - t(i') \right| \leq 0{,}7 \text{ s}$$

pour tous les i et pour tous les i′, où i et i′ désignent les indices des octets contenant le dernier bit des champs consécutifs de type ESCR_base.

NOTE – Le codage des champs de référence d'horloge des flux élémentaires est facultatif; ils n'ont pas à être codés. Cependant, s'ils sont codés, ce paragraphe s'applique.

2.7.4 Fréquence de codage de l'élément presentation timestamp

Le flux de programme et le flux de transport doivent être construits de manière que l'intervalle de temps maximal entre les horodateurs de présentation codés pour chaque flux élémentaire de données vidéo ou audio soit inférieur ou égal à 0,7 s. Autrement dit:

$$\left| tp_n(k) - tp_n(k'') \right| \leq 0{,}7 \text{ s}$$

pour tous les n, k et k″ répondant aux conditions suivantes:

- $P_n(k)$ et $P_n(k'')$ sont les unités de présentation pour lesquelles des horodateurs de présentation sont codés;
- k et k″ sont choisis de telle manière qu'il n'y ait aucune unité de présentation, $P_n(k')$ ayant un horodateur de présentation codé et de manière que $k < k' < k''$;
- aucune discontinuité de décodage n'existe dans le flux élémentaire n entre $P_n(k)$ et $P_n(k'')$.

La contrainte liée à l'observation de 0,7 s n'est pas applicable dans le cas:

- des images fixes définies au 2.1;
- des images fixes AVC;
- des unités d'accès AVC à très faible fréquence d'image, pour lesquelles la différence entre les instants de présentation d'unités d'accès successives est supérieure à 0,7 s. Dans ce cas particulier, les paramètres VUI num_units_in_tick et time_scale seront présents soit dans le flux vidéo AVC, soit dans un descripteur de décodeur HRD et de synchronisation AVC associé au flux vidéo AVC.

NOTE – L'instant de présentation d'une unité d'accès AVC est l'équivalent de l'instant de sortie du tampon DPB $t_{o,dpb}(n)$ défini dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

2.7.5 Codage conditionnel des éléments horodateurs

Pour chaque flux élémentaire d'un flux de programme ou de transport, l'horodateur de présentation (PTS, *presentation timestamp*) doit être codé sur la base de la première unité d'accès.

Une discontinuité de décodage existe au début d'une unité d'accès $A_n(j)$ contenue dans un flux élémentaire si l'instant de décodage $td_n(j)$ de cette unité d'accès a une valeur plus grande que la plus grande valeur admissible compte tenu de la tolérance spécifiée sur la fréquence d'horloge système. Pour les données vidéo, sauf en cas d'activation du mode d'enrichissement ou si le fanion low_delay est mis à '1', cette discontinuité n'est autorisée qu'au début d'une séquence vidéo. Si une discontinuité de décodage existe dans un flux élémentaire de données vidéo ou audio à l'intérieur du flux de transport ou de programme, un horodateur PTS doit être codé sur la base de la première unité d'accès, après chaque discontinuité de décodage, sauf en mode d'enrichissement.

Lorsque le fanion low_delay est mis à '1', il faut coder l'horodateur PTS sur la base de la première unité d'accès à la suite d'un sous-remplissage de la mémoire EB_n ou B_n.

Un horodateur PTS peut être présent dans un en-tête de paquet PES de flux élémentaire vidéo ou audio selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 seulement si le paquet PES contient le premier octet d'un code de déclenchement d'image ou d'une unité d'accès audio.

Un horodateur de décodage (DTS, *decoding timestamp*) doit être contenu dans un en-tête de paquet PES si et seulement si les deux conditions suivantes sont remplies:

- un horodateur PTS est présent dans l'en-tête de paquet PES;
- l'instant du décodage diffère de l'instant de présentation.

Pour chaque image 24 heures AVC, aucune valeur PTS et DTS explicite ne sera codée dans l'en-tête PES. Pour une telle unité d'accès AVC, les décodeurs doivent déduire l'instant de présentation des paramètres contenus dans le flux vidéo AVC. Par conséquent, chaque flux vidéo AVC qui contient une ou plusieurs images 24 heures doit:

- soit acheminer des messages SEI de synchronisation d'image avec des valeurs codées de cpb_removal_delay et de dpb_output_delay;
- soit acheminer des paramètres VUI dont le fanion fixed_frame_rate_flag est mis à '1' et acheminer l'information de numéro d'ordre d'image, ce numéro correspondant au nombre d'intervalles $\Delta t_{fi,dpb}(n)$, où $\Delta t_{fi,dpb}(n)$ est spécifié dans l'équation E-10 de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

  NOTE 1 – Les prescriptions du second alinéa en retrait sont satisfaites si un descripteur de décodeur HRD et de synchronisation AVC est associé au flux vidéo AVC avec le fanion fixed_frame_rate_flag mis à '1' et le fanion temporal_poc_flag mis à '1'.

Pour les unités d'accès AVC d'un flux vidéo AVC acheminé dans un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1, on procède comme suit. Pour chaque unité d'accès AVC qui ne représente pas une image 24 heures AVC, un en-tête PES contenant une valeur PTS codée et, si cela s'applique, une valeur DTS codée sera fourni, à moins que toutes les conditions exprimées dans l'un des quatre alinéas en retrait ci-dessous s'appliquent:

- Les messages SEI suivants sont présents dans la séquence vidéo AVC, comme cela est signalé par les paramètres VUI:

  a) messages SEI de synchronisation d'image donnant les paramètres cpb_removal_delay et dpb_output_delay, et

  b) messages SEI de période de mise en tampon donnant les paramètres initial_cpb_removal_delay et initial_cpb_removal_delay_offset;

  NOTE 2 – Lorsque des messages SEI de synchronisation d'image sont présents dans la séquence vidéo AVC, ces messages sont présents pour chaque unité d'accès AVC, comme requis par la Rec. UIT-T H.264 | ISO/CEI 14496-10. Lorsque des messages SEI de période de mise en tampon sont présents dans la séquence vidéo AVC, ces messages seront présents pour chaque unité d'accès IDR et pour chaque unité d'accès associée à un message SEI de point de rétablissement, comme requis par la Rec. UIT-T H.264 | ISO/CEI 14496-10.
- un descripteur de décodeur HRD et de synchronisation AVC est associé au flux vidéo AVC et, dans ce descripteur, le fanion fixed_frame_rate_flag et le fanion temporal_poc_flag sont mis à '1';
- un descripteur de décodeur HRD et de synchronisation AVC est associé au flux vidéo AVC et, dans ce descripteur, le fanion fixed_frame_rate_flag est mis à '1', le fanion picture_to_display_conversion_flag est mis à '1', le fanion temporal_poc_flag est mis à '0' et des messages SEI de synchronisation d'image contenant le champ pic_struct sont présents dans la séquence vidéo AVC;

  NOTE 3 – Dans ce cas précis, le champ pic_struct sert à déterminer des valeurs PTS subséquentes.
- un descripteur de décodeur HRD et de synchronisation AVC est associé au flux vidéo AVC et, dans ce descripteur, le fanion fixed_frame_rate_flag est mis à '1', le fanion temporal_poc_flag est mis à '0' et le fanion picture_to_display_conversion_flag est mis à '0'.

  NOTE 4 – Dans ce cas, l'information POC contenue dans le flux vidéo AVC sert à déterminer des valeurs PTS subséquentes.

2.7.6 Contraintes de synchronisation applicables au décodage échelonnable

Si une séquence audio est codée au moyen d'un flux binaire d'extension, comme spécifié dans l'ISO/CEI 13818-3, les unités correspondantes de décodage/présentation dans les deux couches doivent avoir des valeurs identiques d'horodate PTS.

Si une séquence vidéo est codée en tant qu'amélioration SNR d'une autre séquence, comme spécifié au 7.8 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les deux séquences doivent comporter deux séries identiques d'instants de présentation.

Si une séquence vidéo est codée sous la forme de deux séquences, comme spécifié au 7.10 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les deux séquences doivent comporter deux séries identiques d'instants de présentation.

Si une séquence vidéo est codée en tant qu'amélioration spatialement échelonnable d'une autre séquence, comme spécifié au 7.7 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les règles suivantes sont applicables:

- Si les deux séquences ont la même fréquence d'image, elles doivent comporter deux séries identiques d'instants de présentation.

NOTE – On notera que cela n'implique pas un même type de codage d'image dans les deux couches.

- Si les deux séquences n'ont pas la même fréquence d'image, leurs séries respectives d'instants de présentation doivent comporter un nombre aussi grand que possible d'instants de présentation communs.
- La prédiction spatiale doit être réalisée à partir de l'une des images suivantes:
  - image simultanée ou décodée le plus récemment de la couche inférieure;
  - image simultanée ou décodée le plus récemment de la couche inférieure à codage prédictif ou intratrame;
  - image suivante de la couche inférieure, par ordre d'ancienneté croissante du décodage, à codage prédictif ou intratrame, à condition que le fanion low_delay de la couche inférieure ne soit pas mis à la valeur '1'.

Si une séquence vidéo est codée en tant qu'amélioration temporellement échelonnable d'une autre séquence, comme spécifié au 7.9 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les images de la couche inférieure désignées ci-dessous peuvent servir de référence. On notera que le temps est mesuré par rapport aux instants de présentation:

- de l'image simultanée ou présentée le plus récemment de la couche inférieure;
- de l'image suivante à présenter de la couche inférieure.

2.7.7 Fréquence de codage de l'élément P-STD_buffer_size dans les en-têtes de paquet d'un flux PES

Dans un flux de programme, les champs P-STD_buffer_scale et P-STD_buffer_size doivent être contenus dans le premier paquet PES de chaque flux élémentaire et aussi chaque fois qu'ils changent de valeur. Ils peuvent également être contenus dans un autre paquet de flux PES.

2.7.8 Codage d'en-tête système dans le flux de programme

Dans un flux de programme, l'en-tête de données systèmes peut être présent dans un bloc de paquets quelconque, immédiatement après l'en-tête de ce bloc. L'en-tête de données systèmes doit être présent dans le premier bloc de paquets d'un flux de programme. Les valeurs codées dans tous les en-têtes de données systèmes d'un même flux de programme doivent être identiques.

2.7.9 Flux de programme de type flux paramétrique contraint du système

Un flux de programme est considéré comme un "flux paramétrique contraint du système" (CSPS, *constrained system parameter stream*) s'il est conforme aux limites spécifiées dans le présent paragraphe. Les flux de transport ne sont pas contraints par les limites spécifiées pour le flux CSPS. Un flux CSPS peut être identifié au moyen du champ CSPS_flag qui est défini au 2.5.3.5 pour les en-têtes de données systèmes. Le flux CSPS est un sous-ensemble de tous les flux de programme possibles.

Débit des paquets

Dans le flux CSPS, le débit maximal d'arrivée des paquets à l'entrée du décodeur P-STD est de 300 paquets par seconde si la valeur codée dans le champ rate_bound (voir 2.5.3.6) est inférieure ou égale à 4 500 000 bit/s, si le fanion de restriction de débit de paquets est mis à la valeur '1', et inférieure ou égale à 2 000 000 bit/s, si le fanion de restriction de débit de paquets est mis à la valeur '0'. Pour des débits plus élevés, le débit de paquets CSPS est limité par une relation linéaire avec la valeur codée dans le champ rate_bound.

Plus précisément, pour tous les blocs de paquets p contenus dans le flux de programme, et si le fanion de restriction de débit de paquets (voir 2.5.3.5) est mis à la valeur '1',

$$NP \leq \left(t(i') - t(i)\right) \times 300 \times \max\left[1, \frac{R_{\max}}{4{,}5 \times 10^6}\right] \tag{2-27}$$

et si le fanion de restriction de débit de paquets est mis à la valeur '0',

$$NP \leq \left(t(i') - t(i)\right) \times 300 \times \max\left[1, \frac{R_{\max}}{2{,}5 \times 10^6}\right] \tag{2-28}$$

où:

$$R_{\max} = 8 \times 50 \times rate_bound \quad \text{bit/s} \tag{2-29}$$

NP est le nombre de préfixes packet_start_code_prefix et de codes system_header_start_code entre champs adjacents de type pack_start_code ou entre le champ last_pack_start_code et le champ MPEG_program_end_code qui est défini dans le Tableau 2-37 et dont la sémantique se trouve au 2.5.3.2.

t(i) est le temps, mesuré en secondes, qui est codé dans le champ system_clock_reference du bloc de paquets p.

t(i′) est le temps, mesuré en secondes, qui est codé dans le champ system_clock_reference du bloc de paquets p + 1, suivant immédiatement le bloc de paquets p; ou, si le dernier bloc de paquets se trouve dans le flux de programme, l'instant d'arrivée de l'octet contenant le dernier bit du champ MPEG_program_end_code.

Capacité de la mémoire tampon du décodeur

Dans le cas d'un CSPS, la taille maximale de chaque tampon d'entrée dans le décodeur modèle est limitée. Différentes limites s'appliquent aux flux élémentaires vidéo et audio.

Dans le cas d'un flux élémentaire vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 11172-2 contenu dans un flux CSPS, ce qui suit s'applique:

La taille BS_n est égale à la somme de la taille du vérificateur de tampon vidéo (VBV, *video buffer verifier*) spécifiée dans le flux Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 11172-2 et d'une taille de tampon supplémentaire, BS_{add}. Celle-ci est spécifiée comme suit:

$$BS_{add} \le MAX\,[6 \times 1024,\ R_{vmax} \times 0{,}001] \text{ octets}$$

où R_{vmax} est le débit maximal du flux élémentaire vidéo Rec. UIT-T H.262 | ISO/CEI 13818-2 ou ISO/CEI 11172-2.

Dans le cas d'un flux élémentaire vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 contenu dans un flux CSPS, ce qui suit s'applique:

La taille BS_n est égale à la somme de cpb_size et d'une taille de tampon supplémentaire BS_{add}. Celle-ci est spécifiée comme suit:

$$BS_{add} \le MAX\,[6 \times 1024,\ R_{vmax} \times 0{,}001] \text{ octets}$$

où R_{vmax} est le débit maximal du flux vidéo AVC, et

où cpb_size est la taille CpbSize[cpt_cnt_minus1] du tampon CPB pour le format de flux d'octets signalé dans les paramètres NAL hrd_parameters() dans le flux vidéo AVC. Si les paramètres NAL hrd_parameters() sont absents du flux vidéo AVC, la taille cpb_size sera la taille définie par 1200 × MaxCPB dans l'Annexe A de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour le niveau appliqué.

Dans le cas d'un flux élémentaire de données audio intégré dans un flux CSPS, la relation suivante est applicable:

$$BS_n \le 4096 \text{ octets}$$

Dans le cas d'un flux élémentaire de données audio ADTS ISO/CEI 13818-7 intégré dans un flux CSPS, la relation suivante est applicable pour la prise en charge de 8 canaux:

$$BS_n \le 8976 \text{ octets}$$

2.7.10 Flux de transport

Verrouillage de la fréquence des échantillons dans les flux de transport

Dans le flux de transport, il doit y avoir une relation de rapport constant et spécifié entre la fréquence d'échantillonnage des données audio et la fréquence de l'horloge système, à l'intérieur du décodeur modèle du système; de même, il doit y avoir une relation de rapport constant et spécifié entre la fréquence de trame des données vidéo et la fréquence de l'horloge système. Le paragraphe 2.4.2 définit le champ system_clock_frequency. La fréquence de trame vidéo est spécifiée dans la Rec. UIT-T H.262 | ISO/CEI 13818-2 ou dans ISO/CEI 11172-2. La fréquence d'échantillonnage audio est spécifiée dans l'ISO/CEI 13818-3 ou dans l'ISO/CEI 11172-3. Pour toutes les unités de présentation contenues dans tous les flux élémentaires de données audio d'un flux de transport, le rapport SCASR entre la fréquence d'horloge

système et la fréquence réelle d'échantillonnage audio est constant et égal à la valeur indiquée dans le tableau suivant, à la fréquence d'échantillonnage nominale qui est indiquée dans le flux de données audio.

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_T\text{-}STD} \tag{2-30}$$

La notation $\frac{X}{Y}$ indique une division de nombres réels.

Fréquence nominale d'échantillonnage audio (kHz)	16	32	22,05	44,1	24	48
SCASR	27 000 000 / 16 000	27 000 000 / 32 000	27 000 000 / 22 050	27 000 000 / 44 100	27 000 000 / 24 000	27 000 000 / 48 000

Pour toutes les unités de présentation contenues dans chaque flux vidéo ISO/CEI 11172-2 et Rec. UIT-T H.262 | ISO/CEI 13818-2 du flux de transport, le rapport SCFR entre la fréquence d'horloge système et la fréquence d'image réelle est constant et égal à la valeur indiquée dans le tableau suivant, à la fréquence d'image nominale indiquée dans le flux vidéo.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_T\text{-}STD} \tag{2-31}$$

Fréquence nominale de trame (Hz)	23,976	24	25	29,97	30	50	59,94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

Les valeurs du rapport SCFR sont exactes. La fréquence de trame réelle diffère légèrement de la fréquence nominale si celle-ci est de 23,976, 29,97 ou 59,94 trames par seconde.

Dans le cas d'un flux vidéo ISO/CEI 14496-2 acheminé dans un flux de transport, la base de temps du flux vidéo ISO/CEI 14496-2, définie par vop_time_increment_resolution, sera verrouillée sur l'horloge STC et sera exactement égale à N fois la fréquence system_clock_frequency divisée par K, avec des entiers N et K qui ont une valeur fixe dans chaque séquence d'objet visuel, K étant supérieur ou égal à N.

Dans le cas d'un flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10, la base de temps du flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 sera verrouillée sur la fréquence d'horloge du système. La fréquence de la base de temps AVC est définie par le paramètre AVC time_scale et cette fréquence sera exactement égale à N fois la fréquence system_clock_frequency divisée par K, avec des entiers N et K qui ont une valeur fixe dans chaque séquence vidéo AVC, K étant supérieur ou égal à N. Par exemple, si time_scale est mis à 90 000, la fréquence de la base de temps AVC est exactement égale à la fréquence system_clock_frequency divisée par 300.

2.8 Compatibilité avec l'ISO/CEI 11172

Le flux de programme de la présente Recommandation | Norme internationale est défini de manière à être compatible par anticipation avec l'ISO/CEI 11172-1. Les décodeurs du flux de programme qui sont définis dans la présente Recommandation | Norme internationale doivent également permettre le décodage des données conformes à ISO/CEI 11172-1.

2.9 Enregistrement des identificateurs de droits d'auteur

2.9.1 Généralités

Les parties 1, 2 et 3 de l'ISO/CEI 13818 permettent d'assurer la gestion des droits d'auteur d'œuvres audiovisuelles. Dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, cette gestion est assurée au moyen d'un descripteur de droits d'auteur, alors que la Rec. UIT-T H.262 | ISO/CEI 13818-2 et l'ISO/CEI 13818-3 contiennent des champs d'identification des détenteurs de droits d'auteur, ces champs syntaxiques étant insérés dans la syntaxe des flux élémentaires. La présente Recommandation | Norme internationale présente la méthode permettant d'obtenir et d'enregistrer les identificateurs de droits d'auteur indiqués dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

La Rec. UIT-T H.222.0 | ISO/CEI 13818-1 spécifie un identificateur de droits d'auteur unique, codé sur 32 éléments binaires, qui identifie le code de type d'œuvre (comme ISBN, ISSN, ISRC, etc.) et qui est inséré dans le descripteur de droits d'auteur. L'identificateur de droits d'auteur permet d'identifier un grand nombre d'organismes d'enregistrement de droits d'auteur. Chaque organisme d'enregistrement de droits d'auteur peut spécifier une syntaxe et une sémantique pour identifier les œuvres audiovisuelles ou d'autres travaux protégés par droits d'auteur dans le cadre de cet organisme d'enregistrement particulier. Cette spécification fait un usage approprié du champ additional_copypright_info (information supplémentaire sur les droits d'auteur), de longueur variable et contenant le numéro d'enregistrement des droits d'auteur.

Le paragraphe suivant et les Annexes L, M et N décrivent les avantages et les responsabilités de toutes les parties à l'enregistrement de l'identificateur de droits d'auteur.

2.9.2 Implémentation d'un organisme d'enregistrement

L'ISO/CEI JTC 1 doit lancer un appel à candidatures pour créer une organisation internationale qui fera office d'organisme d'enregistrement pour l'élément de données **copyright_identifier** qui est défini au 2.6.24. L'organisation choisie fera office d'organisme d'enregistrement. Celui-ci doit remplir ses tâches conformément à l'Annexe H des Directives du JTC 1. L'identificateur de droits d'auteur enregistré sera désigné ci-dessous par le terme identificateur enregistré (RID, *registered identifier*).

Après la sélection de l'organisme d'enregistrement, le JTC 1 doit demander la création d'un comité de gestion pour l'enregistrement (RMG, *registration management group*) qui examinera les appels déposés par les organisations dont l'organisme d'enregistrement a refusé la demande d'identificateur RID à utiliser en association avec la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Les Annexes L, M et N donnent des renseignements sur les procédures d'enregistrement d'un unique identificateur de droits d'auteur.

2.10 Enregistrement du format de données privées

Le descripteur d'enregistrement de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est indiqué ci-après afin de permettre aux utilisateurs de la présente Spécification d'acheminer des données sans ambiguïté lorsque leur format n'est pas reconnu par la présente Spécification. Cette disposition permettra aux systèmes conformes à la présente Spécification d'acheminer tous les types de données tout en offrant une méthode d'identification univoque des caractéristiques des données privées sous-jacentes.

2.10.1 Généralités

Le paragraphe suivant et les Annexes O et P décrivent les avantages et responsabilités de toutes les parties à l'enregistrement du format de données privées.

2.10.2 Implémentation d'un organisme d'enregistrement (RA, *registration authority*)

L'ISO/CEI JTC 1/SC 29 doit lancer un appel à candidatures pour créer une organisation internationale qui fera office d'organisme d'enregistrement pour l'élément de données **format_identifier** qui est défini aux 2.6.8 et 2.6.9. L'organisation choisie fera office d'organisme d'enregistrement. Celui-ci doit remplir ses tâches conformément à l'Annexe H des Directives du JTC 1. L'identificateur de format de données privées enregistré sera désigné ci-dessous par le terme identificateur enregistré (RID, *registered identifier*).

Après la sélection de l'organisme d'enregistrement, le JTC 1 doit demander la création d'un comité de gestion pour l'enregistrement (RMG, *registration management group*) qui examinera les appels déposés par les organisations dont l'organisme d'enregistrement a refusé la demande d'identificateur RID à utiliser en association avec la présente Spécification.

Les Annexes O et P donnent des renseignements sur les procédures d'enregistrement d'un unique identificateur de format de données privées.

2.11 Acheminement de données ISO/CEI 14496

2.11.1 Introduction

Un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1 peut acheminer des flux élémentaires individuels ISO/CEI 14496-2 et 14496-3 tout comme des scènes audiovisuelles ISO/CEI 14496-1 avec leurs flux associés. Généralement, les flux ISO/CEI 14496 sont des éléments d'un programme Rec. UIT-T H.222.0 | ISO/CEI 13818-1, comme défini par la table PMT dans le cas d'un flux de transport et par le mappage PSM dans le cas d'un flux de programme.

Pour l'acheminement de données ISO/CEI 14496 dans des flux de transport et des flux de programme, on fait une distinction entre les flux élémentaires individuels et une scène audiovisuelle ISO/CEI 14496-1 avec ses flux associés. Pour l'acheminement de flux élémentaires individuels ISO/CEI 14496-2 et 14496-3, on utilise uniquement les outils système de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, comme défini au 2.11.2. Pour l'acheminement d'une scène audiovisuelle ISO/CEI 14496-1 et de ses flux élémentaires ISO/CEI 14496 associés, contenus dans des flux SL_packetized ISO/CEI 14496-1 ou FlexMux, on utilise à la fois les outils de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 et ceux de l'ISO/CEI 14496-1, comme défini au 2.11.3.

L'acheminement de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 sur des flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est spécifié au 2.14.

2.11.2 Acheminement de flux élémentaires distincts ISO/CEI 14496-2 et 14496-3 dans des paquets PES

2.11.2.1 Introduction

Des flux élémentaires ISO/CEI 14496-2 et 14496-3 distincts peuvent être acheminés dans des paquets PES en tant qu'octets de données de paquet PES. Concernant la mise en paquets PES, aucune contrainte particulière relative à l'alignement de données ne s'applique. Pour ce qui est de la synchronisation, des systèmes d'horodatage de présentation (PTS) et, selon le cas, des systèmes d'horodatage de décodage (DTS) sont codés dans l'en-tête du paquet PES qui achemine les données de flux élémentaire ISO/CEI 14496. En ce qui concerne le codage PTS et DTS, les mêmes contraintes que celles des flux élémentaires ISO/CEI 13818 s'appliquent. Voir le Tableau 2-93 pour avoir un aperçu général sur la manière d'acheminer des flux ISO/CEI 14496 distincts dans un flux conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Tableau 2-93 – Acheminement de flux ISO/CEI 14496 distincts dans un flux selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

ISO/CEI 14496-2 vidéo	Acheminement dans paquets PES	Stream_type = 0x10	Stream_id = '1110 xxxx'
ISO/CEI 14496-3 audio	Acheminement dans paquets PES	Stream_type = 0x11	Stream_id = '110x xxxx'

Le système PTS ou DTS présent dans l'en-tête de paquet PES doit désigner l'objet vidéo qui suit le premier code de début VOP ou le premier code de départ de type d'objet de texture fixe situé au début du paquet PES. Chaque flux vidéo ISO/CEI 14496-2 acheminé par un flux selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 doit contenir les informations nécessaires pour décoder le flux vidéo ISO/CEI 14496-2. Le flux doit donc contenir des en-têtes de séquence d'objet vidéo, des en-têtes d'objet visuel et les en-têtes de couche objet vidéo.

Dans le cas d'un flux élémentaire ISO/CEI 14496-3, avant la mise en paquets PES, les données du flux élémentaire sont encapsulées dans la syntaxe de transport LATM/LOAS AudioSyncStream() définie dans l'ISO/CEI 14496-3. Un pointeur PTS présent dans l'en-tête de paquet PES doit désigner la première trame audio qui suit les premiers mots de synchronisation situés au début de la charge utile du paquet PES.

L'acheminement de flux élémentaires ISO/CEI 14496-2 et ISO/CEI 14496-3 individuels dans des paquets PES doit être identifié par des valeurs stream_id et stream_type appropriées, indiquant l'utilisation de flux vidéo ISO/CEI 14496-2 ou de flux audio ISO/CEI 14496-3. En outre, cet acheminement doit être signalé par le descripteur MPEG-4_video ou MPEG-4_audio. Ces descripteurs doivent être insérés dans la boucle de descripteur de l'entrée de flux élémentaire contenue dans la table de mappage de programme dans le cas d'un flux de transport ou dans le mappage de flux de programme, lorsqu'il est présent, dans le cas d'un flux de programme. La Rec. UIT-T H.222.0 | ISO/CEI 13818-1 ne spécifie pas la présentation des flux élémentaires ISO/CEI 14496-2 et ISO/CEI 14496-3 dans le contexte d'un programme.

2.11.2.2 Extensions du système STD pour des flux élémentaires distincts ISO/CEI 14496

Le modèle T-STD comporte une mémoire tampon de transport TB_n et une mémoire tampon de multiplex B_n avant le décodage de chaque flux élémentaire distinct n ISO/CEI 14496. Il faut noter que la mémoire tampon de multiplex B_n s'applique aussi à la vidéo ISO/CEI 14496-2 comme indiqué à la Figure 2-4, et non pas la méthode à deux mémoires tampons MB_n et EB_n utilisée pour la vidéo ISO/CEI 13818-2 dans le modèle T-STD. Pour les mémoires tampons TB_n et B_n et le débit Rx_n de l'une à l'autre, les contraintes suivantes s'appliquent.

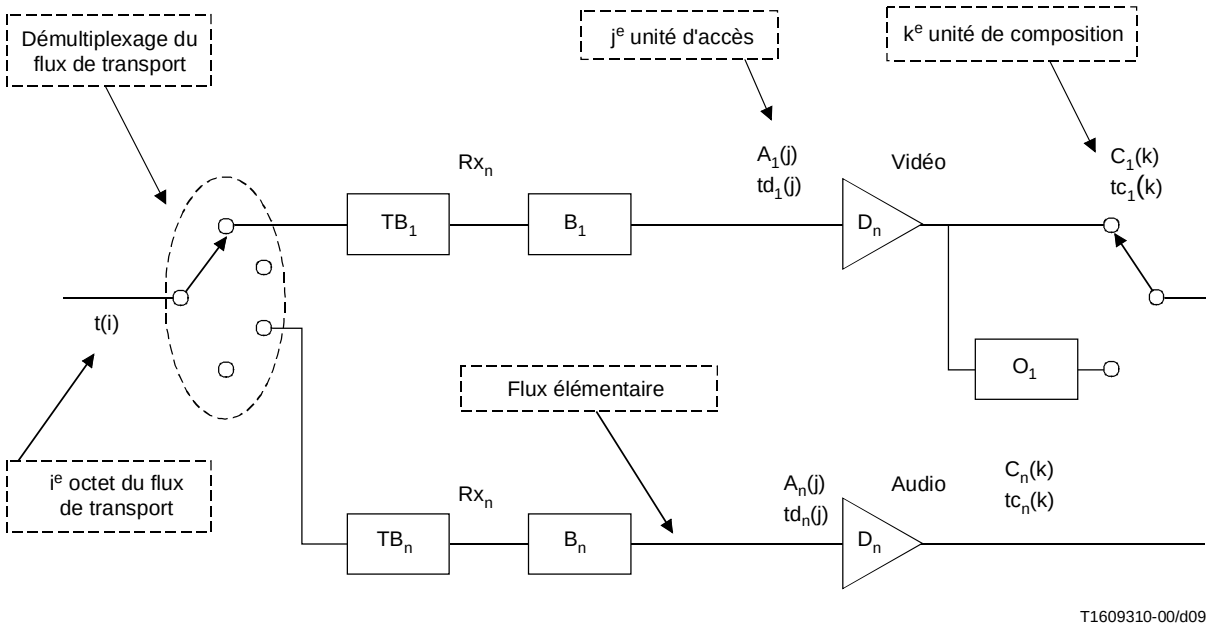


Figure 2-4 – Extensions du modèle T-STD pour les flux élémentaires distincts ISO/CEI 14496

Dans le cas de l'acheminement d'un flux ISO/CEI 14496-2:

capacité BS_n de la mémoire tampon B_n:

$$BS_n = BS_{mux} + BS_{oh} + VBV_{max}[profil, niveau]$$

où:

BS_{oh}, tampon de surdébit de paquet, est défini par:

$BS_{oh} = (1/750)$ secondes $\times \max\{R_{max}[profil, niveau], 2\,000\,000$ bit/s$\}$

et:

BS_{mux}, tampon de multiplexage supplémentaire, par:

$BS_{mux} = 0{,}004$ seconde $\times \max\{R_{max}[profil, niveau], 2\,000\,000$ bit/s$\}$

Débit Rx_n:

$$Rx_n = 1{,}2 \times R_{max}[profil, niveau]$$

où:

$VBV_{max}[profil, niveau]$ et $R_{max}[profil, niveau]$ sont définis dans l'ISO/CEI 14496-2 pour chaque profil et niveau. Dans le cas des profils et niveaux pour lesquels aucune valeur VBV_{max} n'est spécifiée, la capacité de mémoire tampon B_n et le débit Rx_n sont définis par l'utilisateur.

Dans le cas de l'acheminement d'un flux ISO/CEI 14496-3:

capacité BS_n de la mémoire B_n pour les données audio AAC selon l'ISO/CEI 14496-3,

$$\text{sinon } BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ octets}$$

Dans ce cas, les tailles de la mémoire tampon de décodage de l'unité d'accès BS_{dec} et de la mémoire tampon du préfixe de paquet BS_{oh} sont soumises à la contrainte suivante:

$$BS_{dec} + BS_{oh} \leq 2848 \text{ octets}$$

Une partie (736 octets) de la mémoire tampon de 3584 octets est affectée au multiplexage. Le reste, 2848 octets, est partagé entre les mémoires tampons BS_{dec}, BS_{oh} et la mémoire tampon de multiplexage additionnel.

Débit Rx_n pour les données audio AAC selon l'ISO/CEI 14496-3 identique à celui défini pour les données audio ADTS selon l'ISO/CEI 13818-7 au 2.4.2.3,

$$\text{autre } Rx_n = 2\,000\,000 \text{ bit/s}$$

Le modèle P-STD comporte une mémoire tampon de multiplex B_n avant le décodage de chaque flux élémentaire distinct n ISO/CEI 14496. La mémoire de capacité BS_n de la mémoire B_n du modèle P-STD est définie par le champ P-STD_buffer_size contenu dans l'en-tête de paquet PES.

2.11.3 Acheminement de scènes ISO/CEI 14496-1 audiovisuelles et de flux ISO/CEI 14496 associés

2.11.3.1 Introduction

Le présent paragraphe aborde les problèmes d'encapsulation et de signalisation lors de l'acheminement d'une scène audiovisuelle, représentée par les données ISO/CEI 14496, dans un flux de programme ou un flux de transport conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Le contenu ISO/CEI 14496 est constitué du descripteur initialObjectDescriptor et d'un nombre variable de flux tels que des flux de descripteurs d'objets, des flux de description de scène (acheminant soit une commande BIFS soit des unités d'accès BIFS-Anim), des flux IPMP, des flux OCI ou des flux audiovisuels. Chaque flux ISO/CEI 14496 doit être contenu dans un flux de paquets SL et peut, en option, être multiplexé dans un flux FlexMux. Ces flux sont tous les deux définis dans l'ISO/CEI 14496-1. Pour l'acheminement des flux de programme ou des flux de transport selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, ces flux de paquets SL ou ces flux FlexMux doivent contenir les champs codés (OCR, *object clock reference*) et (FCR, *flexmux clock reference*) spécifiés dans le 2.11.3.4 et dans le 2.11.3.5 respectivement. Les flux de paquets SL ou les flux FlexMux sont ensuite encapsulés dans des paquets PES ou dans des sections ISO/CEI 14496 avant la paquetisation et le multiplexage du flux de transport. Les sections ISO/CEI 14496 sont construites selon le format long des sections H.222.0 | ISO/CEI 13818-1.

2.11.3.2 Attribution de valeurs ES_ID

Une scène ISO/CEI 14496-1 acheminée dans un flux selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 peut associer un certain nombre de flux ISO/CEI 14496, ISO/CEI 13818 ou autres au moyen du paramètre ES_ID. La scène et les flux associés peuvent être acheminés dans le même flux selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Toutefois, il peut arriver qu'une scène corresponde à des flux acheminés à un autre endroit, par exemple à un réseau IP. Ces autres possibilités ne sont pas examinées dans la présente Spécification.

L'ISO/CEI 14496-1 définit les règles de détection de nom des identificateurs. Ces règles permettent d'utiliser la même valeur ES_ID pour deux flux différents dans le contenu ISO/CEI 14496. Lorsqu'une ou plusieurs scènes ISO/CEI 14496-1 sont acheminées dans un programme UIT-T H.222.0 | ISO/CEI 13818-1, ces valeurs dupliquées ES_ID ne peuvent pas apparaître dans le programme tel que chaque flux paquet SL ISO/CEI 14496 ou canal FlexMux ISO/CEI 14496-1 ait une valeur unique d'identificateur ES_ID dans le programme..

2.11.3.3 Synchronisation des scènes ISO/CEI 14496 et des flux associés

Lorsqu'un flux ISO/CEI 14496 est véhiculé sur un flux UIT-T H.222.0 | ISO/CEI 13818-1, sa base de temps objet sera verrouillée sur l'horloge STC de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Ainsi,

$$\text{si } X(t) = f_{stc}(t) / f_{object}(t)$$

la valeur de X(t) doit être constante à n'importe quel instant t

où:

$f_{stc}(t)$ désigne la fréquence désirée de l'horloge STC à l'instant t, c'est-à-dire 27 000 000 Hz

$f_{object}(t)$ désigne la fréquence de la base de temps objet à l'instant t.

La base de temps objet des flux ISO/CEI 14496 transportés sur des flux UIT-T H.222.0 | ISO/CEI 13818-1 est acheminée comme suit:

- La base de temps objet d'un flux paquet SL acheminé dans les paquets PES sans utilisation du flux FlexMux doit être acheminée par des champs OCR codés dans l'en-tête des paquets SL du flux considéré (voir 2.11.3.4).
- La base de temps objet d'un flux paquet SL acheminé dans les paquets PES dans un flux FlexMux doit être acheminée par des champs FCR du flux FlexMux (voir 2.11.3.5). En conséquence, tous les flux ISO/CEI 14496 contenus dans le même flux FlexMux utilisent la même base de temps objet commune.

– La base de temps objet d'un flux paquet SL acheminé dans des sections doit être acheminée par un autre flux ISO/CEI 14496 dans le flux transport ou dans le flux de programme indiqué dans le champ OCR_ES_ID dans le descripteur d'ES pour ce champ.

Le codage des champs OCR et des FCR dans les flux paquets SL et les flux FlexMux insérés dans des flux UIT-T H.222.0 | ISO/CEI 13818-1 est soumis aux contraintes suivantes:

– les références OCR et FCR dans chaque flux paquet SL et de chaque flux FlexMux associées à la même scène doivent avoir la même résolution;

– les résolutions des références OCR et FCR, f_{cr}, pour une même scène, doivent être inférieures ou égales à 90 000 Hz;

– le rapport $(f_{stc}(t) / 300) / f_{cr}$, doit être un entier supérieur ou égal à 1. Par conséquent, la résolution des éléments de syntaxe OCR et FCR ne peut être que de 90 000 Hz, 45 000 Hz, 30 000 Hz, 22 500 Hz, 18 000 Hz, etc.

Compte tenu des contraintes ci-dessus et de la contrainte propre à l'ISO/CEI 14496-1 selon laquelle la résolution f_{cr} doit être un nombre entier de Hz par seconde, la valeur de f_{cr} peut être choisie en fonction de la scène.

Les horodates ISO/CEI 14496 codées dans l'en-tête des paquets SL doivent désigner des instants de la base de temps objet du flux acheminé dans le paquet SL. La résolution de chaque horodate doit être inférieure d'un facteur 2^k à la résolution des références OCR ou FCR associées au flux, k étant un entier positif supérieur ou égal à zéro. Pour obtenir le même rapport de résolution, la longueur des champs d'horodatage, TimeStampLength, doit être inférieure de k bits à la longueur du champ OCR ou FCR, OCRLength et FCRLength, respectivement. Ainsi, pour chaque flux, le codage des horodates doit respecter les conditions suivantes:

– TimeStampResolution = (OCRResolution ou FCRResolution respectivement)/2^k, k étant un entier positif supérieur ou égal à zéro. D'après l'ISO/CEI 14496-1 TimeStampResolution doit être égal à un nombre entier de Hz.

– TimeStampLength = OCRLength ou FCRLength respectivement – k.

La relation entre la valeur de l'horloge STC et la valeur correspondante de la base de temps objet d'un flux est établie en associant les champs PTS dans les en-têtes de paquets PES aux OCR ou FCR dans les en-têtes de paquets SL et les paquets de flux FlexMux respectivement, comme spécifié au 2.11.3.6 et au 2.11.3.7.

2.11.3.4 Synchronisation de flux empaquetés SL

Pour acheminer le contenu ISO/CEI 14496 dans un flux conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, on utilise un flux de paquets SL ISO/CEI 14496-1. Pour chaque flux de paquets SL acheminé dans un paquet PES sans utiliser le FlexMux, le champ objectClockReference doit être codé de la façon suivante:

1) un champ objectClockReference (OCR) doit être présent dans l'en-tête du premier paquet SL d'un flux paquet SL;

2) le flux de paquets SL doit être élaboré de telle façon que l'intervalle de temps entre les octets contenant le dernier bit des champs successifs OCR soit inférieur ou égal à 0,7 s. Ainsi:

$$|t(i'') - t(i')| <= 0.7\,\text{s}$$

pour tous les i′ et i″ où i′ et i″ sont les indices des octets contenant le dernier bit des champs OCR consécutifs dans le flux FlexMux.

Si un champ objectClockReference est codé dans un en-tête SL, le champ instantBitrate doit également être codé.

2.11.3.5 Synchronisation de flux FlexMux

Outre les flux de paquets SL, on peut utiliser l'outil FlexMux ISO/CEI 14496-1 pour acheminer le contenu ISO/CEI 14496 dans les flux conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Les paquets FlexMux doivent être composés de paquets SL, tels que spécifiés dans l'ISO/CEI 14496-1. Dans chaque flux FlexMux acheminé dans le flux conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, le champ fmxClockReference doit être codé de la façon suivante:

1) un champ fmxClockReference (FCR) doit être présent dans le premier paquet FlexMux d'un flux FlexMux;

2) le flux FlexMux doit être élaboré de telle façon que l'intervalle de temps entre les octets contenant le dernier bit des champs successifs FCR soit inférieur ou égal à 0,7 s. Ainsi:

$$|t(i'') - t(i')| <= 0.7\,\text{s}$$

pour tous les i′ et i″ où i′ et i″ sont les indices des octets contenant le dernier bit des champs FCR consécutifs dans le flux FlexMux;

3) toutes les horodates ISO/CEI 14496 dans les flux paquets SL acheminés dans un flux FlexMux doivent correspondre à des instants de la base de temps objet acheminée dans les champs FCR du flux FlexMux. Il n'est pas nécessaire que les flux paquets SL acheminés dans des paquets FlexMux acheminent des champs OCR. Si des champs OCR sont présents, ils peuvent être ignorés.

2.11.3.6 Acheminement de flux de paquets SL dans des paquets PES

Un seul flux de paquets SL selon l'ISO/CEI 14496-1 peut être mappé avec un seul flux PES. Un et un seul paquet SL d'un flux de paquets SL doit constituer la charge utile d'un paquet PES. Les paquets PES qui acheminent un flux de paquets SL doivent être identifiés par stream_id = 0xFA dans l'en-tête PES.

Lorsqu'un champ OCR est codé dans l'en-tête de paquet SL, un pointeur PTS doit être codé dans l'en-tête du paquet PES qui achemine cet en-tête de paquet SL. Ce pointeur PTS doit être codé avec la valeur sur 33 bits de la partie à 90 kHz de l'horloge STC qui correspond à la valeur de la base de temps objet à l'instant indiqué par l'OCR.

L'identificateur ES_ID associé au flux de paquets SL doit être signalé par un descripteur SL comme spécifié au 2.6.46.

2.11.3.7 Acheminement de paquets FlexMux dans des paquets PES

Les paquets PES, dont la charge utile est composée de paquets FlexMux, doivent être identifiés par stream_id = 0xFB dans l'en-tête de paquet PES. Un nombre entier de paquets FlexMux doit composer la charge utile d'un paquet PES. Autrement dit, la charge utile d'un paquet PES acheminant un flux FlexMux doit débuter par un en-tête de paquet FlexMux et se terminer par le dernier octet d'un paquet FlexMux.

Si un champ fmxClockReference (FCR) est codé dans un des paquets FlexMux contenus dans un paquet PES, un pointeur PTS doit être codé dans l'en-tête du paquet PES qui contient ce paquet FlexMux. Ce pointeur PTS doit être codé avec la valeur sur 33 bits de la partie à 90 kHz de l'horloge STC qui correspond à la valeur de la base de temps objet du flux FlexMux à l'instant indiqué par la FCR. Si de nombreux paquets FlexMux codés avec des champs FCR sont contenus dans un paquet PES, le pointeur PTS doit correspondre au temps indiqué par la référence FCR dans le premier de ces paquets FlexMux rencontré dans la charge utile du paquet PES.

Les identificateurs ES_ID associés à chaque flux de paquets SL acheminé dans le flux FlexMux doivent être signalés par un descripteur FMC, comme spécifié au 2.6.44.

2.11.3.8 Acheminement de paquets SL et de paquets FlexMux dans des sections

Pour le transport d'un contenu ISO/CEI 14496 dans des sections, des sections ISO/CEI 14496 sont définies. Seuls les flux de descripteur d'objet paquet SL et les flux de description de scène doivent utiliser des sections ISO/CEI 14496. Une seule section ISO/CEI 14496 doit contenir soit un nombre entier de paquets FlexMux acheminant chacun un paquet SL du même flux élémentaire ISO/CEI 14496-1.

Le Tableau 2-94 représente la syntaxe des sections ISO/CEI 14496 définies pour acheminer des flux élémentaires ISO/CEI 14496-1 acheminant des éléments de programme ISO/CEI 14496, qualifiés par l'identificateur table_id comme des descripteurs d'objet ou des données de flux de description de scène. Les données de flux de descripteur d'objet représentent une table de descripteurs d'objets comprenant un certain nombre de descripteurs d'objets. La table de descripteurs d'objets peut être transmise dans plusieurs sections ISO/CEI 14496. Il n'est pas nécessaire d'attendre de recevoir une table complète pour procéder au traitement de sa charge utile. Toutefois, la charge utile des sections doit être traitée dans l'ordre correct, tel qu'indiqué par la valeur du champ numéro de section dans les octets d'en-tête de section ISO/CEI 14496.

Tableau 2-94 – Syntaxe utilisée par les sections pour l'acheminement des flux ISO/CEI 14496

Syntaxe	Nombre de bits	Mnémonique
ISO_IEC_14496_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
ISO_IEC_14496_section_length	**12**	**uimsbf**
table_id_extension	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
if (PMT_has_SL_descriptor(current_PID)) {		
SL_Packet()		
}		
else if (PMT_has_FMC_descriptor(current_PID)) {		
for (i=1; i<N; i++)		
FlexMuxPacket()		
}		
else {		
for (i=1; i<N; i++)		
reserved	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		

table_id (identificateur de table) **–** Ce champ, codé sur 8 bits, doit être mis à la valeur '0x04' ou '0x05' dans le cas d'une section ISO/CEI 14496. La valeur '0x04' indique une section description de scène acheminant un flux de description de scène ISO/CEI 14496-1. La valeur '0x05' indique une section description d'objet acheminant un flux de descripteur d'objet ISO/CEI 14496-1.

section_syntax _indicator (indicateur de syntaxe de section) – Ce champ, codé sur 1 bit, doit être mis à '1'.

private_indicator (indicateur privé) – Ce champ, codé sur 1 bit, ne doit pas être défini par la présente Spécification.

ISO_IEC_14496_section_length (longueur de section ISO/CEI 14496) – Ce champ, codé sur 12 bits, indique le nombre d'octets restants dans la section qui suit immédiatement le champ ISO_IEC_14496_section_length jusqu'à la fin de la section ISO/CEI 14496. La valeur de ce champ ne doit pas dépasser 4093 (0xFFD).

table_id_extension (extension d'identificateur de table) – Ce champ, codé sur 16 bits, ne doit pas être défini par la présente Spécification; son utilisation et sa valeur sont définies par l'utilisateur.

version_number (numéro de version) – Ce champ, codé sur 5 bits, représente le numéro de version de la table de descripteurs d'objet ou de la table de description de scène. Le numéro de version sera incrémenté de la valeur 1 puis ramené à la valeur modulo 32 à chaque modification de la table. La mise à jour de la version est effectuée par l'application.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Ce champ, codé sur 1 bit, doit être mis à 1.

section_number (numéro de section) – Ce champ, codé sur 8 bits, indique le numéro de la section ISO/CEI 14496. Le champ numéro de la première section ISO/CEI 14496 de la table de descripteurs d'objet ou de la table de description de scène doit avoir une valeur égale à 0x00. La valeur du numéro de la section doit être incrémentée de 1 à chaque nouvelle section contenue dans la table.

last_section_number (numéro de la dernière section) – Ce champ, codé sur 8 bits, indique le numéro de la dernière section de la table de descripteurs d'objet ou de la table de description de scène dont cette section fait partie.

PMT_has_SL_descriptor(current_PID) – Pseudo-fonction qui doit être "Vraie" si un descripteur SL est contenu dans la boucle de descripteur contenue dans la table de mappage de programme pour l'élément de programme UIT-T H.222.0 | ISO/CEI 13818-1 qui achemine cette section ISO/CEI 14496.

SL_Packet() – Paquet de couche sync, tel que spécifié au 10.2.2 de l'ISO/CEI 14496-1.

PMT_has_FMC_descriptor(current_PID) – Pseudo-fonction qui doit être "Vraie" si un descripteur FMC est contenu dans la boucle de descripteur contenue dans la table de mappage de programme pour l'élément de programme ISO/CEI 13818-1 acheminant cette section ISO/CEI 14496.

FlexMuxPacket() – Paquet FlexMux spécifié au 11.2.4 de l'ISO/CEI 14496-1.

CRC_32 (contrôle CRC sur 32 bits) – Ce champ, codé sur 32 bits, contient la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe A après traitement de toute la section ISO/CEI 14496.

2.11.3.9 Extensions du modèle T-STD

2.11.3.9.1 Modèle T-STD utilisé pour le contenu 14496

La Figure 2-5 représente les extensions du décodeur modèle du système pour les flux de transport en vue de l'acheminement d'éléments de programme ISO/CEI 14496 encapsulés dans des flux de transport conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

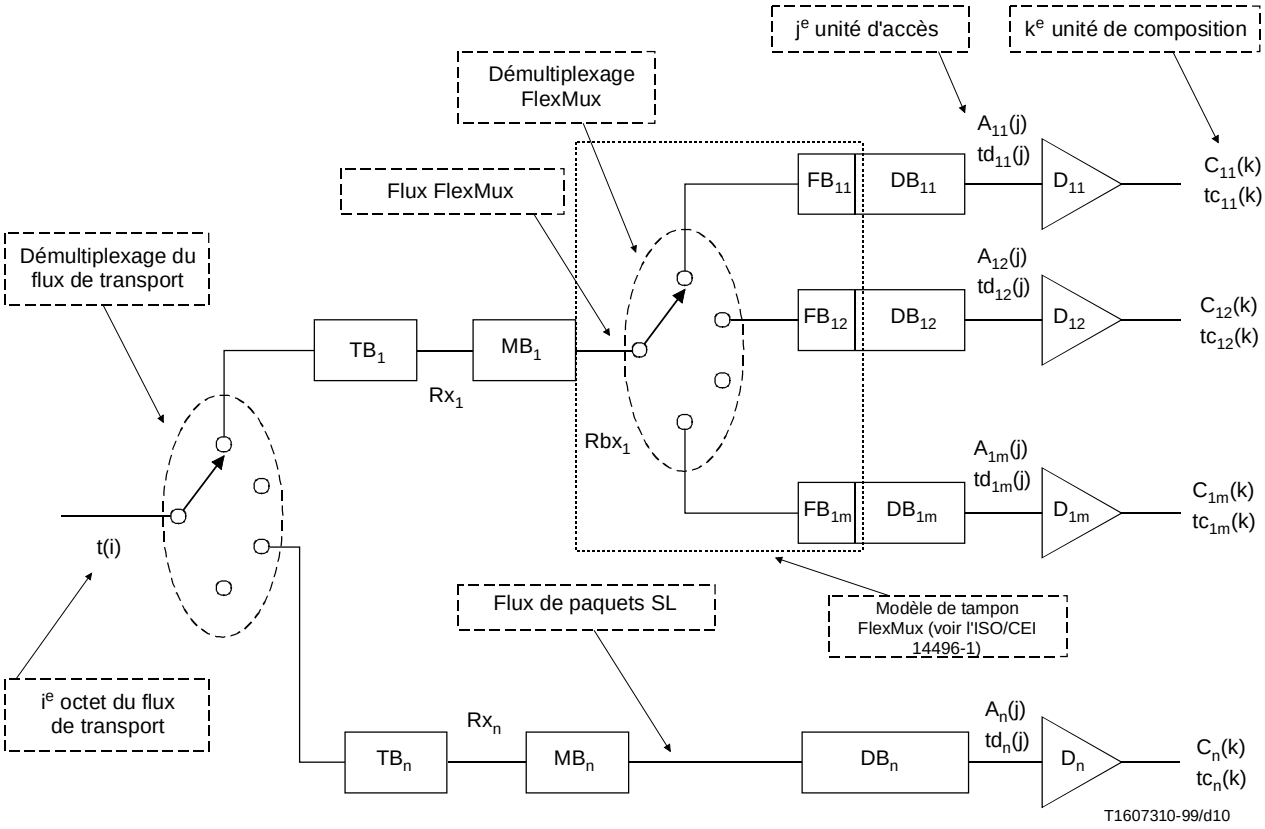


Figure 2-5 – Modèle T-STD utilisé pour le contenu ISO/CEI 14496

La notation suivante est utilisée dans la Figure 2-5 et sa description:

TB_n	mémoire tampon de transport.
MB_n	mémoire de multiplexage destinée au flux FlexMux n ou au flux empaqueté SL n.
FB_{np}	mémoire tampon FlexMux destinée au flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.
DB_{np}	mémoire tampon du décodeur destinée au flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.
DB_n	mémoire tampon du décodeur destinée au flux élémentaire n.
D_{np}	décodeur destiné au flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.

D_n décodeur du flux élémentaire n.

Rx_n débit d'extraction des données de la mémoire TB_n.

Rbx_n débit d'extraction des données de la mémoire MB_n.

$A_{np}(j)$ j^e unité d'accès dans le flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n. Les $A_{np}(j)$ sont indexées dans l'ordre de décodage.

$A_n(j)$ j^e unité d'accès dans le flux élémentaire n. Les $A_n(j)$ sont indexées dans l'ordre de décodage.

$td_{np}(j)$ temps de décodage, mesuré en secondes dans le décodeur modèle du système de la j^e unité d'accès dans le flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.

$Td_n(j)$ temps de décodage, mesuré en secondes, dans le décodeur modèle du système de la j^e unité d'accès dans le flux élémentaire n.

$C_{np}(k)$ k^e unité de composition dans le flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n. $C_{np}(k)$ résulte du décodage de $A_{np}(j)$. Les $C_{np}(k)$ sont indexées dans l'ordre de composition.

$C_n(k)$ k^e unité de composition dans le flux élémentaire n. $C_n(k)$ résulte du décodage de $A_n(j)$. Les $C_n(k)$ sont indexées dans l'ordre de composition.

$tc_{np}(k)$ temps de composition, mesuré en secondes, dans le décodeur modèle du système de l'unité de composition k^e contenue dans le flux élémentaire dans la voie FlexMux p du flux FlexMux n.

$tc_n(k)$ temps de composition, mesuré en secondes, dans le décodeur modèle du système de l'unité de composition k^e contenue dans le flux élémentaire n.

t(i) indique, en secondes, l'instant auquel le i^e octet du flux de transport pénètre dans le décodeur modèle du système.

2.11.3.9.2 Traitement des flux FlexMux

Les paquets de flux de transport, complets et contenant des données issues du flux FlexMux n, sont transmis au tampon de transport TB_n pour le flux FlexMux. La taille du tampon TB_n est fixée à 512 octets. Tous les octets qui entrent dans le tampon TB_n en sont extraits au débit Rx_n, spécifié par le champ TB_leak_rate figurant dans le descripteur MultiplexBuffer associé au flux FlexMux n. Lorsqu'il n'y a plus de données dans le tampon TB_n, le débit Rx_n est égal à zéro. Les paquets TS en double ne sont pas remis au tampon MB_n.

En cas d'acheminement dans des paquets PES, les octets d'en-tête de paquet PES et de données de charge utile sont insérés dans le tampon MB_n; tous les autres octets sortant du tampon TB_n n'entrent pas dans le tampon MB_n mais peuvent être utilisés pour piloter le système. En cas d'acheminement dans des sections conformes à l'ISO/CEI 14496, les octets de l'en-tête de section, de charge utile et de données CRC-32 sont insérés dans le tampon MB_n; tous les autres octets n'entrent pas dans le tampon MB_n mais peuvent être utilisés pour piloter le système. Dans un cas comme dans l'autre, la capacité du tampon MB_n est spécifiée par le champ MB_buffer_size du descripteur MultiplexBuffer.

Les octets des paquets du flux FlexMux dans le tampon MB_n sont tous insérés dans leur tampon FlexMux associé au débit spécifié par le champ fmxRate codé dans le flux FlexMux et conforme au modèle de tampon FlexMux défini au 11.2.9 de l'ISO/CEI 14496-1. Seuls les octets de données de charge utile des paquets FlexMux dans la voie FlexMux p sont insérés dans le tampon FB_{np}. Les octets d'en-tête des paquets FlexMux dans la voie FlexMux p du flux FlexMux n sont ignorés et peuvent être utilisés pour commander le système. Le débit spécifié dans le champ fmxRate doit être applicable à tous les paquets FlexMux contenus dans le flux suivant immédiatement le paquet de voie de référence d'horloge FlexMux jusqu'au prochain paquet de voie de référence d'horloge FlexMux rencontré. En l'absence de données de flux FlexMux dans le tampon MB_n, aucun octet n'est éliminé du tampon MB_n. La base de temps de la scène est synchronisée avec l'horloge STC au moyen d'un champ de pointeur PTS qui est codé dans l'en-tête PES. Les octets issus de l'en-tête PES ou de l'en-tête de section ISO/CEI 14496 précédant immédiatement un en-tête FlexMux sont instantanément extraits et mis de côté pour éventuellement servir au pilotage du système. Les octets issus des champs CRC-32 de la section ISO/CEI 14496, qui suivent immédiatement le dernier paquet du flux FlexMux dans la charge utile de section sont instantanément extraits et mis de côté pour éventuellement servir à vérifier l'intégrité des données. Les octets issus de la voie de référence d'horloge FlexMux sont instantanément extraits et mis à l'écart pour éventuellement servir à caler sur l'horloge STC la base de temps de l'objet ISO/CEI 14496. Lorsqu'il n'y a aucun octet de paquet PES ou de données de charge utile de section (selon le cas) dans le tampon MB_n, aucune donnée n'est extraite du tampon MB_n. Toutes les données qui entrent dans MB_n en sortent. Tous les octets de charge utile de paquet PES du flux n entrent instantanément dans le démultiplexeur, dès qu'ils sortent du tampon MB_n.

2.11.3.9.3 Définition du tampon FlexMux, FB_{np}

Pour chaque voie p d'un flux FlexMux n, la capacité du tampon FB_{np} est définie au moyen du descripteur FmxBufferSize. Les octets de charge utile des paquets FlexMux sont transférés du tampon FB_{np} vers le tampon du

décodeur DB_{np} conformément au modèle de tampon FlexMux défini au 11.2.9 de l'ISO/CEI 14496-1. Seuls les octets de données de charge utile des paquets FlexMux dans la voie FlexMux n sont insérés dans le tampon DB_{np}. Les octets d'en-tête des paquets SL dans la voie FlexMux p du flux FlexMux sont ignorés et peuvent servir à commander le système.

2.11.3.9.4 Traitement des flux de paquets SL

Les paquets de flux de transport, complets et contenant des données issues du flux SL, sont transmis au tampon de transport TB_n pour le flux SL. Tous les octets qui entrent dans le tampon TB_n en sont extraits au débit Rx_n, spécifié par le champ TB_leak_rate figurant dans le descripteur MultiplexBuffer associé au flux SL n. Lorsqu'il n'y a plus de données dans le tampon TB_n, le débit Rx_n est égal à zéro. Les paquets TS en double ne sont pas remis au tampon MB_n.

En cas d'acheminement dans des paquets PES, les octets d'en-tête de paquet PES et de données de charge utile sont insérés dans le tampon MB_n; tous les autres octets sortant du tampon TB_n n'entrent pas dans le tampon MB_n mais peuvent être utilisés pour piloter le système. En cas d'acheminement dans des sections conformes à l'ISO/CEI 14496, les octets de l'en-tête de section, de charge utile et de données CRC-32 sont insérés dans le tampon MB_n; tous les autres octets n'entrent pas dans le tampon MB_n mais peuvent être utilisés pour piloter le système. Dans un cas comme dans l'autre, la capacité du tampon MB_n est spécifiée par le champ MB_buffer_size du descripteur MultiplexBuffer.

Les octets du flux paquet SL dans le tampon MB_n sont tous insérés dans le tampon de décodeur DB_n au débit spécifié par le champ instantBitRate codé dans le flux paquet SL et conforme au modèle de décodeur de système défini au 7.4 de l'ISO/CEI 14496-1. Le débit spécifié par le champ instantBitRate est applicable à tous les octets de données contenus dans le flux de paquets SL qui suivent immédiatement le champ instantBitRate dans l'en-tête de paquet SL jusqu'au prochain champ instantBitRate rencontré. En l'absence de données de flux paquet SL dans le tampon MB_n, aucun octet n'est éliminé du tampon MB_n. Les octets issus de l'en-tête PES ou de l'en-tête de section ISO/CEI 14496 précédant immédiatement un en-tête SL sont instantanément extraits et mis à l'écart pour éventuellement servir au pilotage du système. Les octets issus des champs CRC-32 de la section ISO/CEI 14496, qui suivent immédiatement le dernier paquet du flux SL dans la charge utile de section, sont instantanément extraits et mis à l'écart pour éventuellement servir à vérifier l'intégrité des données. Lorsqu'il n'y a aucun octet de paquet PES ou de données de section (selon le cas) dans le tampon MB_n, aucune donnée n'est extraite du tampon MB_n. Toutes les données qui entrent dans MB_n en sortent. Tous les octets de charge utile de paquet PES du flux n entrent instantanément dans le tampon DB_n, dès qu'ils sortent du tampon MB_n, à l'exception de l'en-tête de paquet SL. Les octets issus de l'en-tête de paquet SL n'entrent pas dans le tampon DB_n et peuvent être utilisés pour piloter le système. La capacité du tampon de décodeur DB_n est indiquée par la valeur bufferSizeDB du descripteur DecoderConfig défini dans l'ISO/CEI 14496-1.

2.11.3.9.5 Gestion des tampons

Les flux de transport doivent être construits de façon que les conditions définies dans le présent paragraphe soient satisfaites.

Le tampon TB_n ne doit pas déborder. Il doit être vide au moins une fois par seconde. Le tampon MB_n ne doit pas déborder. Le tampon FB_{np} ne doit pas déborder. Les tampons DB_{np} et DB_n ne doivent pas déborder ni être en sous-remplissage. Le sous-remplissage du tampon DB_{np} se produit lorsqu'un ou plusieurs octets d'une unité d'accès ne sont pas présents dans le tampon DB_{np} à l'instant du décodage associé à cette unité d'accès. Le sous-remplissage du tampon DB_n se produit lorsqu'un ou plusieurs octets d'une unité d'accès ne sont pas présents dans le tampon DB_n à l'instant du décodage associé à cette unité d'accès.

2.11.3.10 Acheminement dans un flux de transport

2.11.3.10.1 Aperçu général

Un flux de transport peut contenir un ou plusieurs programmes dont chacun est décrit par une table de mappage de programme. Le contenu en format ISO/CEI 14496 peut être acheminé en plus des types de flux déjà définis pour un tel programme. Les éléments du contenu ISO/CEI 14496 peuvent être acheminés dans un ou plusieurs éléments de programme UIT-T H.222.0 | ISO/CEI 13818-1 référencés par une unique valeur d'identificateur PID contenue dans un flux de transport. Un cas particulier peut se présenter car il est possible qu'un programme contenu dans un flux de transport ne soit composé que d'éléments de programme de type ISO/CEI 14496. Le contenu ISO/CEI 14496, associé à un programme et acheminé dans le flux de transport, doit être référencé dans la table de mappage de ce programme. Un descripteur d'objet initial doit être utilisé pour définir une scène ISO/CEI 14496. L'utilisation de ce descripteur est spécifiée au 2.11.3.10.2.

L'acheminement du contenu ISO/CEI 14496 dans un identificateur PID est signalé par une valeur de type de flux de 0x12 ou 0x13 dans la table de mappage de programme qui est associée à cette valeur d'identificateur PID. La valeur 0x12 indique l'acheminement dans des paquets PES. Le champ stream_id contenu dans l'en-tête de paquet PES signale si les paquets PES contiennent un seul paquet SL ou un certain nombre de paquets FlexMux. Une valeur de champ stream_type de 0x13 dans la table de mappage de programme indique que l'élément de programme achemine un flux de

descripteur d'objet ou de commande BIFS contenu dans des sections. Dans ce cas, l'identificateur table_id contenu dans l'en-tête de section indique si un flux de descripteur d'objet est acheminé dans les sections ou dans un flux de commande BIFS. Voir également le Tableau 2-95. La section contient soit un seul paquet SL soit un certain nombre de paquets FlexMux, selon ce qui est indiqué par la présence d'un descripteur SL ou FMC, respectivement, dans la boucle de descripteur de la table de mappage de programme pour l'élément de programme UIT-T H.222.0 | ISO/CEI 13818-1 qui achemine les sections. Lorsque du contenu ISO/CEI 14496 est acheminé, les descripteurs SL et FMC doivent spécifier l'identificateur ES_ID pour chaque flux ISO/CEI 14496 encapsulé. Lorsque l'attribution des valeurs ES_ID change, la table de mappage de programme doit être mise à jour et le numéro de version de cette table doit être augmenté de 1 modulo 32. L'Annexe R donne un exemple de procédure d'accès au contenu pour des éléments de programme ISO/CEI 14496 contenus dans un flux de transport.

Tableau 2-95 – Options définies par l'ISO/CEI pour l'acheminement de scènes ISO/CEI 14496 et de flux associés dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

Flux de descripteur d'objet ISO/CEI 14496-1	Encapsulation dans des paquets SL	Acheminement dans des paquets PES	Stream_type = 0x12	Stream_id = '1111 1010'
		Acheminement dans des sections ISO/CEI 14496	Stream_type = 0x13	Table_id = 0x05
	Encapsulation dans des paquets SL suivie d'un multiplex dans des paquets FlexMux	Acheminement dans des paquets PES	Stream_type = 0x12	Stream_id = '1111 1011'
		Acheminement dans des sections ISO/CEI 14496	Stream_type = 0x13	Table_id = 0x05
Flux de description de scènes ISO/CEI 14496-1	Encapsulation dans des paquets SL	Acheminement dans des paquets PES	Stream_type = 0x12	Stream_id = '1111 1010'
		Acheminement dans des sections ISO/CEI 14496	Stream_type = 0x13	Table_id = 0x04
	Encapsulation dans des paquets SL suivie d'un multiplex dans des paquets FlexMux	Acheminement dans des paquets PES	Stream_type = 0x12	Stream_id = '1111 1011'
		Acheminement dans des sections ISO/CEI 14496	Stream_type = 0x13	Table_id = 0x04
Tous les autres flux ISO/CEI 14496	Encapsulation dans des paquets SL	Acheminement dans des paquets PES	Stream_type = 0x12	Stream_id = '1111 1010'
	Encapsulation dans des paquets SL suivie d'un multiplex dans des paquets FlexMux	Acheminement dans des paquets PES	Stream_type = 0x12	Stream_id = '1111 1011'

2.11.3.10.2 Descripteur d'objet initial

En cas d'acheminement d'une scène ISO/CEI 14496-1, le descripteur d'objet initial ISO/CEI 14496-1 sert de point d'accès initial à tous les flux associés. Le descripteur d'objet initial doit être inséré dans le descripteur IOD qui est situé dans la boucle de descripteur qui suit immédiatement le champ program_info_length de la table PMT du programme auquel la scène est associée. Il contient des descripteurs ES désignant les flux de description de scène et d'objet qui font partie de ce programme. Il peut contenir également des descripteurs ES désignant un ou plusieurs flux IPMP ou OCI associés. L'identification des flux est effectuée au moyen d'identificateurs ES_ID comme spécifié à l'article 8 de l'ISO/CEI 14496-1.

2.11.3.11 Modèle P-STD pour le contenu 14496

La Figure 2-6 montre le modèle de décodeur STD lorsque des systèmes ISO/CEI 14496 sont acheminés dans un flux de programme.

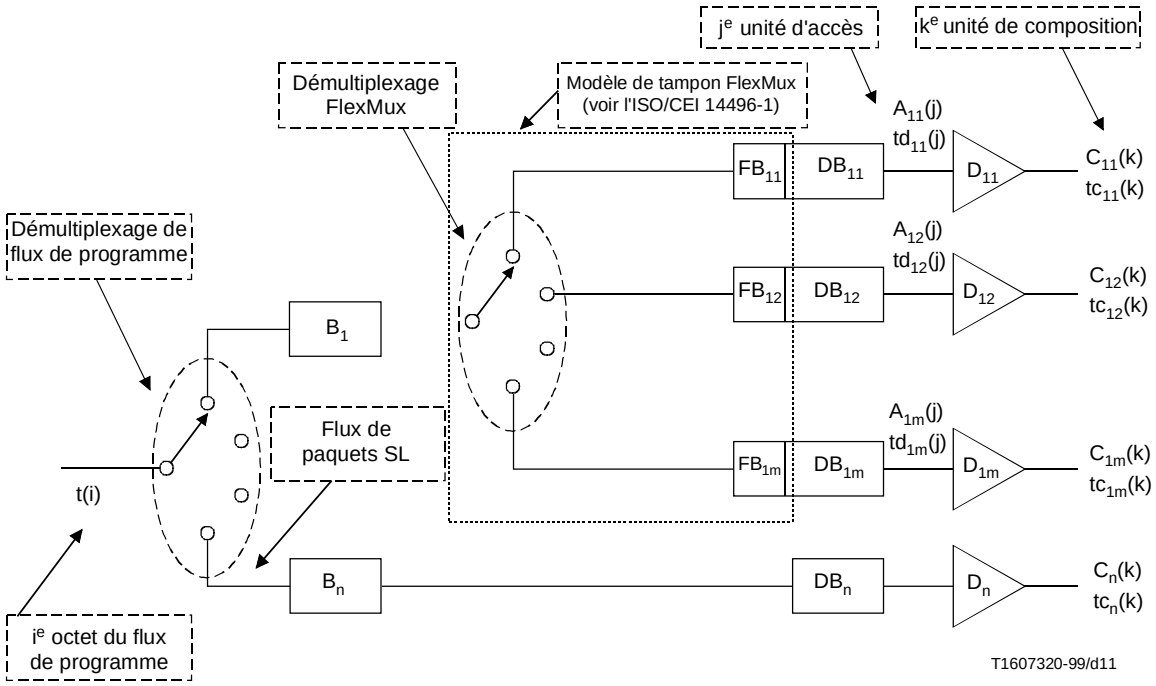


Figure 2-6 – Modèle P-STD pour l'ISO/CEI 14496 dans un flux de programme

La notation suivante est utilisée dans la Figure 2-6 et sa description:

B_n	mémoire tampon d'entrée pour le flux FlexMux n ou le flux paquet SL n.
FB_{np}	mémoire tampon FlexMux destinée au flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.
DB_{np}	mémoire tampon du décodeur destinée au flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.
DB_n	mémoire tampon du décodeur destinée au flux élémentaire n.
D_{np}	décodeur destiné au flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n.
D_n	décodeur du flux élémentaire n.
$A_{np}(j)$	j^e unité d'accès dans le flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n. Les $A_{np}(j)$ sont indexées dans l'ordre de décodage.
$A_n(j)$	j^e unité d'accès dans le flux élémentaire n. Les $A_n(j)$ sont indexées dans l'ordre de décodage.
$td_{np}(j)$	temps de décodage, mesuré en secondes, dans le décodeur modèle du système de la j^e unité d'accès dans le flux élémentaire de la voie FlexMux p du flux FlexMux n.
$td_n(j)$	temps de décodage, mesuré en secondes, dans le décodeur modèle du système de la j^e unité d'accès dans le flux élémentaire n.
$C_{np}(k)$	k^e unité de composition dans le flux élémentaire contenu dans la voie FlexMux p du flux FlexMux n. $C_{np}(k)$ résulte du décodage de $A_{np}(j)$. Les $C_{np}(k)$ sont indexées dans l'ordre de composition.
$C_n(k)$	k^e unité de composition dans le flux élémentaire n. $C_n(k)$ résulte du décodage de $A_n(j)$. Les $C_n(k)$ sont indexées dans l'ordre de composition.
$tc_{np}(k)$	temps de composition, mesuré en secondes, dans le décodeur modèle du système de l'unité de composition k^e contenue dans le flux élémentaire dans la voie FlexMux p du flux FlexMux n.
$tc_n(k)$	temps de composition, mesuré en secondes, dans le décodeur modèle du système de l'unité de composition k^e contenue dans le flux élémentaire n.
t(i)	indique, en secondes, l'instant auquel le i^e octet du flux de transport pénètre dans le décodeur modèle du système.

2.11.3.11.1 Traitement des flux FlexMux

A l'entrée du décodeur STD, chaque octet de la charge utile des paquets PES acheminant un flux FlexMux n est instantanément transféré dans le tampon B_n. L'octet i est introduit dans le tampon B_n à l'instant t(i). Les octets de l'en-tête de paquet PES ne sont pas placés dans le tampon B_n et peuvent servir à la commande du système. La capacité du tampon B_n est spécifiée par le champ P-STD_buffer_size contenu dans l'en-tête du paquet PES qui achemine le flux n.

Les octets des paquets de flux FlexMux du tampon B_n sont tous transférés dans leur tampon FlexMux associé au débit spécifié par le champ fmxRate codé dans le flux FlexMux et conformes au modèle de tampon FlexMux défini au 11.2.9 de l'ISO/CEI 14496-1. Seuls les octets de charge utile paquet FlexMux dans la voie FlexMux p du flux FlexMux n sont placés dans le tampon FB_{np}. Les octets d'en-tête paquet FlexMux dans la voie FlexMux p du flux FlexMux n sont ignorés et peuvent servir à la commande du système. Le débit spécifié par le champ fmxRate doit être applicable à tous les paquets FlexMux du flux, jusqu'au prochain paquet de voie de référence d'horloge FlexMux rencontré. Les octets issus de la voie de référence d'horloge sont instantanément extraits et mis à l'écart pour éventuellement servir à caler sur l'horloge STC la base de temps de l'objet ISO/CEI 14496. Lorsqu'il n'y a aucune donnée de charge utile de paquets PES dans le tampon B_n, aucune donnée n'est extraite de celui-ci. Toutes les données qui entrent dans B_n en sortent. Tous les octets de charge utile de paquets PES du flux n entrent instantanément dans le démultiplexeur FlexMux à leur sortie de B_n.

2.11.3.11.2 Définition du tampon FlexMux, FB_{np}

Pour chaque voie p d'un flux FlexMux n, la taille du tampon FB_{np} est définie en utilisant le descripteur FmxBufferSize si un mappage de flux de programme est présent dans le flux de programme. Les octets de charge utile des paquets FlexMux sont transférés du tampon FB_{np} vers le tampon de décodeur DB_{np} conformément au modèle de tampon FlexMux défini au 11.2.9 de l'ISO/CEI 14496-1. Seuls les octets de charge utile des paquets de la voie FlexMux p du flux FlexMux n sont placés dans le tampon DB_{np}. Les octets d'en-tête des paquets SL de la voie FlexMux p du flux FlexMux sont ignorés et peuvent servir à la commande du système.

2.11.3.11.3 Traitement des flux SL paquet

A l'entrée du décodeur STD, chaque octet de la charge utile des paquets PES acheminant un flux SL paquet n est instantanément transféré dans le tampon B_n. L'octet i est introduit dans le tampon B_n à l'instant t(i). Les octets de l'en-tête de paquet PES ne sont pas placés dans le tampon B_n et peuvent servir à la commande du système. La capacité du tampon B_n est spécifiée par le champ P-STD_buffer_size contenu dans l'en-tête du paquet PES qui achemine le flux n. Les octets des paquets SL du tampon B_n sont transférés dans le tampon de décodeur DB_n au débit spécifié par le champ instantBitRate codé dans le flux paquet SL et conformément au modèle de décodeur système défini au 7.4 de l'ISO/CEI 14496-1. Le débit spécifié par le champ instantBitRate doit être applicable à tous les octets de données contenus dans le flux de paquets SL, jusqu'au prochain champ instantBitRate rencontré. S'il n'y a plus de données de charge utile de paquet PES présentes dans le tampon B_n, aucune donnée n'est extraite de celui-ci. Toutes les données qui entrent dans le tampon B_n en sortent. Tous les octets du flux n entrent instantanément dans le tampon DB_n à leur sortie de B_n, à l'exception de l'en-tête de paquets SL. Les octets issus de l'en-tête de paquets SL n'entrent pas dans le tampon DB_n et peuvent être utilisés pour piloter le système. La capacité du tampon de décodeur DB_n est indiquée par la valeur bufferSizeDB du descripteur DecoderConfig défini dans l'ISO/CEI 14496-1.

2.11.3.11.4 Gestion des tampons

Les flux de programme doivent être construits de façon que le tampon B_n ne soit pas en débordement. Le tampon FB_{np} ne doit pas déborder. Les tampons DB_{np} et DB_n ne doivent pas déborder ni être en sous-remplissage. Le sous-remplissage du tampon DB_{np} se produit lorsqu'un ou plusieurs octets d'une unité d'accès ne sont pas présents dans le tampon DB_{np} à l'instant du décodage associé à cette unité d'accès. Le sous-remplissage du tampon DB_n se produit lorsqu'un ou plusieurs octets d'une unité d'accès ne sont pas présents dans le tampon DB_n à l'instant du décodage associé à cette unité d'accès.

2.11.3.12 Acheminement dans un flux de programme

2.11.3.12.1 Aperçu général

Un flux de programme ne contient qu'un seul programme. Des données ISO/CEI 14496 peuvent être acheminées également dans les types de flux déjà définis pour un tel programme. Un cas particulier peut se présenter car il est possible qu'un flux de programme n'achemine que des données ISO/CEI 14496. Lorsqu'un mappage de flux de programme est présent, le contenu ISO/CEI 14496 ainsi acheminé dans le flux de programme doit être référencé comme suit. L'acheminement de scènes ISO/CEI 14496-1 et de flux ISO/CEI 14496 associés dans des paquets SL et FlexMux est indiqué par l'identificateur de flux approprié et par un descripteur d'objet initial dont l'utilisation est spécifiée au 2.11.3.12.2. Pour chaque flux ISO/CEI 14496 acheminé, le descripteur SL et le descripteur FMC doivent spécifier l'identificateur ES_ID. Lorsque l'attribution des valeurs ES_ID change, le mappage de flux de programme, s'il existe,

doit être mis à jour et la version de ce mappage sera incrémentée de la valeur 1, puis ramenée à sa valeur modulo 32. Il convient de noter que dans un flux de programme, le contenu ISO/CEI 14496 peut être aussi référencé par des moyens privés.

L'Annexe R donne un exemple de procédure d'accès au contenu pour des éléments de programme ISO/CEI 14496 contenus dans un flux de programme.

2.11.3.12.2 Descripteur d'objet initial

En cas d'acheminement d'une scène ISO/CEI 14496-1, le descripteur d'objet initial ISO/CEI 14496 sert de point d'accès initial à tous les flux associés. Lorsqu'un mappage de flux de programme est présent dans le flux de programme, le descripteur d'objet initial doit être inséré dans le descripteur IOD qui est situé dans la boucle de descripteur qui suit immédiatement le champ program_stream_info_length. Il contient des descripteurs ES désignant les flux de description de scène et d'objet qui font partie de ce programme. Il peut contenir également des descripteurs ES désignant un ou plusieurs flux IPMP ou OCI associés. L'identification des flux est effectuée au moyen d'identificateurs ES_ID comme spécifié à l'article 8 de l'ISO/CEI 14496-1. Dans un flux de programme, le descripteur d'objet initial peut être aussi acheminé par des moyens privés.

2.12 Acheminement des métadonnées

2.12.1 Introduction

Un flux UIT-T H.222.0 | ISO/CEI 13818-1 peut acheminer des métadonnées. Le format des métadonnées peut être défini par l'ISO ou par un autre organisme. Les modalités d'acheminement des métadonnées sont définies dans le présent paragraphe; y sont également définis les mécanismes de transport, ainsi que la signalisation relative aux métadonnées, le modèle de base de temps appliqué aux métadonnées et les extensions du modèle STD aux fins du décodage des métadonnées.

On entend par service de métadonnées un ensemble cohérent de métadonnées de même format transmises à un récepteur pour une fin déterminée. Les services de métadonnées sont contenus dans des flux de métadonnées; chaque flux de métadonnées véhicule un ou plusieurs services de métadonnées. La présente Spécification repose sur la notion d'unités d'accès aux métadonnées dans un service de métadonnées. La définition d'une unité d'accès aux métadonnées est propre au format des métadonnées, mais chaque service de métadonnées est supposé représenter une concaténation (ou un ensemble) d'unités d'accès aux métadonnées.

Pour le transport d'un service de métadonnées sur un flux UIT-T H.222.0 | ISO/CEI 13818-1, un seul identificateur de service de métadonnées est attribué au service considéré. Un identificateur de service de métadonnées désigne de manière univoque un service de métadonnées parmi tous les services de métadonnées offerts sur un même flux de transport ou de programme et *non pas seulement* parmi ceux d'un flux de métadonnées. L'identificateur de service de métadonnées est utilisé pour extraire le service de métadonnées et toutes les informations nécessaires pour le décoder.

Le décodage des métadonnées peut nécessiter la mise à disposition des données de configuration du décodeur. Si un service de métadonnées véhiculé dans un flux UIT-T H.222.0 | ISO/CEI 13818-1 nécessite les données de configuration du décodeur aux fins du décodage, celles-ci doivent être acheminées dans le même programme du même flux UIT-T H.222.0 | ISO/CEI 13818-1.

Le 2.12.2 traite de la synchronisation des métadonnées; le 2.12.3 donne un aperçu des outils définis pour le transport des métadonnées sur un flux UIT-T H.222.0 | ISO/CEI 13818-1. Les modalités d'utilisation des outils de transport disponibles sont énoncées aux 2.12.4 à 2.12.8; la signalisation relative aux métadonnées est définie au 2.12.9. Enfin, le modèle STD pour le décodage des métadonnées est défini au 2.12.10.

Etant donné que de nombreuses formes de métadonnées peuvent être acheminées, il est indispensable d'indiquer non seulement le format et le codage précis des métadonnées, mais aussi la signification sémantique que celles-ci véhiculent. Les deux premiers éléments (format et codage) sont indiqués dans le format des métadonnées; la signification sémantique est indiquée dans le format d'application des métadonnées. En d'autres termes, le format des métadonnées véhicule les procédures de décodage des métadonnées, le format d'application des métadonnées véhiculant quant à lui les instructions d'utilisation des métadonnées, qui indiquent pour l'essentiel quelle application utilisent les métadonnées. Cette distinction est importante car elle sépare le codage ou la représentation des métadonnées de leur signification, ce qui permet à une application d'ignorer les moyens par lesquels ses métadonnées sont acheminées.

2.12.2 Modèle de chronogramme pour les métadonnées

Les métadonnées peuvent renvoyer à des codes temporels associés au contenu, par exemple pour indiquer le début d'un segment de contenu. Chaque indication temporelle figurant dans les métadonnées renvoie à un certain chronogramme de contenu de métadonnées propre au format des métadonnées et/ou au format d'application des métadonnées effectivement utilisé. Par exemple, un format d'application de métadonnées peut utiliser des codes temporels UTC, alors

qu'un autre format d'application de métadonnées peut utiliser des codes temporels SMPTE. Pour permettre le transport du contenu à tout moment sur un support quelconque, le chronogramme du contenu des métadonnées est censé, mais pas obligatoirement, ignorer le mode de transport.

Pour le transport du contenu et des métadonnées associées sur des flux UIT-T H.222.0 | ISO/CEI 13818-1, des références temporelles précises au contenu doivent être conservées dans les métadonnées. Il en va de même si les métadonnées sont acheminées par d'autres moyens. A cet effet, le modèle de chronogramme de la Figure 2-7 est pris comme base d'analyse dans la présente Spécification.

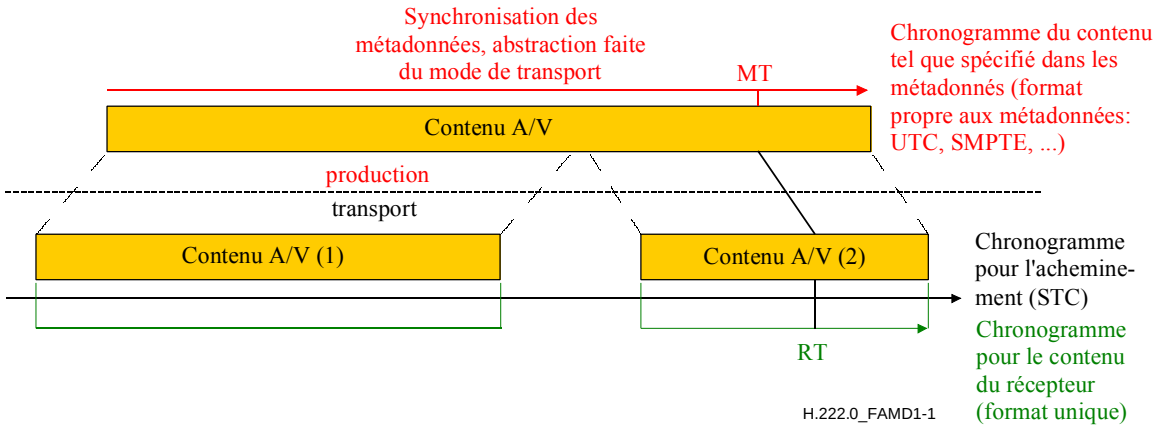


Figure 2-7 – Modèle de synchronisation pour l'acheminement du contenu et des métadonnées

Les métadonnées sont associées au contenu audiovisuel, abstraction faite généralement du mode de transport, au stade de la production ou à quelque autre stade antérieur au transport. En cas de besoin, une information temporelle est insérée dans les métadonnées pour indiquer par exemple certains segments du contenu, au moyen du chronogramme de contenu de métadonnées utilisé dans les métadonnées. Par exemple, des codes temporels UTC ou SMPTE peuvent être utilisés. Le format du chronogramme est indépendant de tout code temporel qui peut ou non être inséré dans le flux audiovisuel. Par exemple, le chronogramme des métadonnées peut utiliser des codes temporels UTC, alors que des codes temporels SMPTE sont insérés dans le flux vidéo.

Chaque flux de métadonnées doit satisfaire aux conditions suivantes:

- le chronogramme du contenu des métadonnées ne doit subir aucune interruption;
- ce chronogramme doit être verrouillé sur l'horloge d'échantillonnage du contenu;
- chaque référence temporelle figurant dans le flux de métadonnées doit se rapporter au même chronogramme de contenu de métadonnées.

Aux fins du transport, une base de temps déterminée – dénommée chronogramme d'acheminement – est associée au contenu. Si le transport est assuré sur un flux UIT-T H.222.0 | ISO/CEI 13818-1, le chronogramme d'acheminement est fourni par l'horloge de base du système (STC, *system time clock*). Le contenu peut être acheminé sous forme d'éléments d'information contigus, mais il est également possible d'en interrompre l'acheminement, par exemple dans le cas d'interruptions d'un programme pour diffuser des flashs d'information; en pareils cas ou dans d'autres, le chronogramme peut subir des interruptions.

En cas d'utilisation dans les métadonnées de références temporelles, celles-ci doivent être associées de manière non équivoque, dans le décodeur de référence du système (STD, *system target decoder*), aux valeurs temporelles du contenu reçu. Pour procéder à cette opération, un chronogramme du contenu du récepteur est nécessaire. L'horloge de base du système (STC) peut faire office d'un tel chronogramme, mais en raison des interruptions qu'elle risque de subir, l'horloge STC ne permet pas nécessairement de procéder à une association temporelle sans équivoque. En conséquence, la notion de durée de lecture normale (NPT, *normal play time*) dérivée de la commande DSM-CC ISO/CEI 13818-6 peut également être utilisée comme chronogramme pour le contenu du récepteur. Dans tout mode de lecture – normal, inversé, ralenti, avance rapide, retour en arrière rapide et image fixe, par exemple – la durée de lecture normale offre une association temporelle sans équivoque, indépendamment des interruptions de l'horloge STC et de l'insertion d'autres contenus. A noter qu'un nouveau descripteur de référence de durée de lecture normale (NPT_reference_descriptor) doit être transmis en cas de réinitialisation de l'horloge STC.

Pour maintenir la précision des références temporelles au contenu à partir des métadonnées, il faut être dûment informé des procédures de mappage d'une valeur temporelle pour les métadonnées (MT, *metadata time*), définie dans le

chronogramme du contenu des métadonnées, sur la valeur temporelle correspondante pour le récepteur (RT, *receiver time*), du chronogramme du contenu du récepteur. Ce mappage est assuré par la mise en place du décalage de temps (en unités de 90 kHz) entre le chronogramme du contenu des métadonnées et le chronogramme du contenu du récepteur. Le décalage est introduit dans le descripteur d'étiquetage de contenu. Il véhicule la valeur de la base de temps pour les métadonnées à l'instant auquel la base de temps du contenu du récepteur atteint une valeur spécifiée. Voir aussi la Figure 2-7.

La base de temps interne aux systèmes de métadonnées peut se rapporter à une image ou une trame audio déterminée, utilisant par exemple des codes temporels SMPTE. Le décalage de temps entre le chronogramme du contenu des métadonnées et le chronogramme du contenu du récepteur est exprimé en unités de 90 kHz; par conséquent, la référence de temps des métadonnées sera convertie en une valeur de 90 kHz dans les récepteurs. Pour tenir compte des erreurs, les récepteurs doivent fonctionner selon le principe qu'en cas de référence à une image ou à une trame audio, le critère de la correspondance la plus étroite doit être retenu. Par exemple, la référence de temps des métadonnées convertie à 90 kHz doit être appariée à l'image ou à la trame dont la valeur de l'horodateur de présentation (PTS) est la plus proche de la valeur convertie.

En cas d'utilisation de la durée de lecture normale (NPT), quel que soit le mode de reproduction et à quelque moment que ce soit, le décalage entre la base de temps pour les métadonnées et la base de temps NPT demeure constant. Tant que l'horloge STC ne subit pas d'interruptions et qu'il n'est pas inséré d'autres contenus, le décalage entre la base de temps pour les métadonnées et la base de temps de l'horloge STC demeure lui aussi constant, mais seulement en mode de lecture normale. Pour des chronogrammes définis à titre privé, le décalage doit aussi être constant, mais éventuellement dans le cadre de contraintes non définies dans la présente Spécification.

En cas d'application du transport synchrone de métadonnées dans des paquets PES ou moyennant l'utilisation du protocole de téléchargement synchronisé DSM-CC, des horodateurs de présentation (PTS) sont attribués aux métadonnées. Ces horodateurs PTS peuvent par exemple indiquer l'instant marquant le début de la durée de validité des métadonnées. Cela suppose de savoir *a priori* comment associer les métadonnées à la base de temps pour l'acheminement. Toutefois, les métadonnées transportées en mode synchrone peuvent également contenir des références temporelles, qu'il convient de mapper du chronogramme du contenu des métadonnées sur le chronogramme du contenu du récepteur en utilisant le décalage spécifié entre les deux chronogrammes. Voir aussi la Figure 2-8.

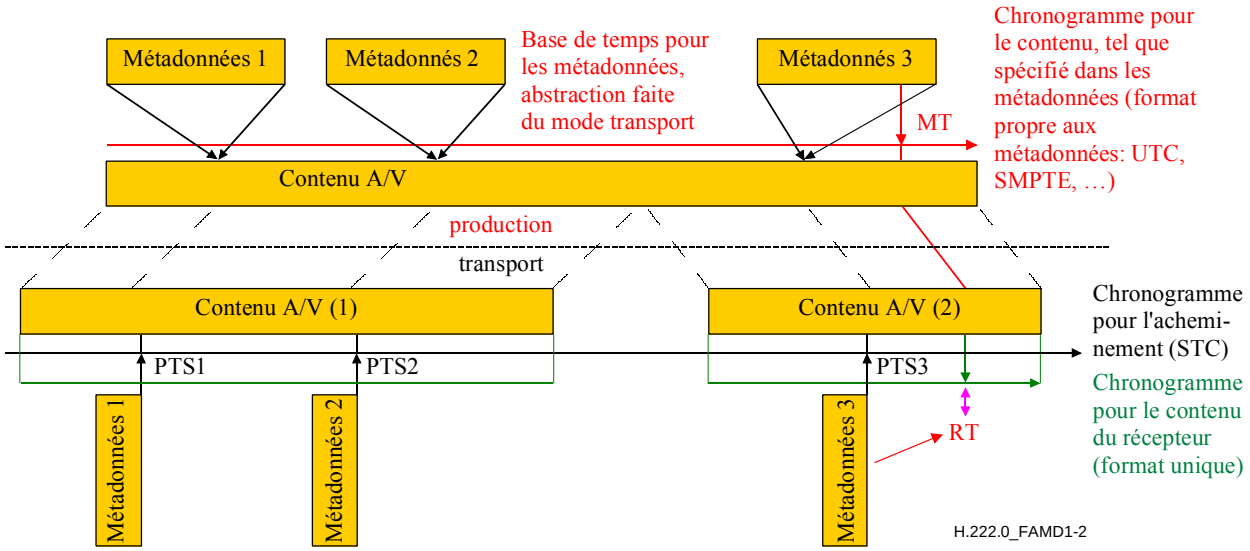


Figure 2-8 – Acheminement de métadonnées dans les paquets PES

2.12.3 Options de transport des métadonnées

Compte tenu de la grande diversité des caractéristiques des métadonnées, divers outils sont définis pour transporter les métadonnées sur un flux UIT-T H.222.0 | ISO/CEI 13818-1.

La présente Spécification définit deux outils d'acheminement en mode synchrone des métadonnées:

- l'acheminement dans des paquets PES,
- l'utilisation du protocole de téléchargement synchronisé DSM-CC.

En outre, la présente Spécification définit trois outils d'acheminement en mode asynchrone des métadonnées:

- l'acheminement dans des sections de métadonnées,
- l'utilisation de carrousels de données DSM-CC,
- l'utilisation de carrousels d'objets DSM-CC.

A noter que certaines des options de transport en mode asynchrone autorisent l'utilisation de carrousels et de structures de fichier. Le choix de l'outil de transport dépend des conditions applicables à l'acheminement des métadonnées ainsi que des caractéristiques des outils, comme indiqué dans les paragraphes suivants.

Les métadonnées peuvent également être acheminées par des moyens privés, tels que des paquets PES ayant pour valeur d'identificateur de flux 0xBD ou 0xBF (private_stream_id_1 ou private_stream_id_2) ou des sections privées. La présente Spécification n'indique pas comment utiliser des moyens privés pour acheminer des métadonnées, mais autorise l'utilisation des descripteurs définis aux 2.6.56 à 2.6.63 aux fins de la signalisation de ces métadonnées.

La référenciation de base des services de métadonnées, assurée au moyen de l'identificateur de service de métadonnées, est la même pour tous les outils. Il existe toutefois des différences d'un outil à l'autre. En cas d'utilisation de paquets PES, de sections de métadonnées ou de sections de téléchargement synchronisé DSM-CC, les données provenant de chaque service de métadonnées sont expressément signalées dans un flux de métadonnées, au moyen du champ metadata_service_id. Cependant, en cas d'utilisation de carrousels DSM-CC, cette signalisation est librement déterminée dans le cadre des applications de métadonnées. A noter que la présente Spécification autorise l'acheminement d'un service de métadonnées dans un carrousel DSM-CC, mais qu'elle ne limite en rien le nombre de services de métadonnées qui peuvent être acheminés dans un tel carrousel.

Lorsqu'elles sont acheminées dans un descripteur de métadonnées, les données de configuration du décodeur de métadonnées sont expressément signalées dans des paquets PES ayant pour valeur de type de flux (stream_type) 0x15 et pour valeur d'identificateur de flux (stream_id) 0xFC, dans des sections de métadonnées ou dans des sections de téléchargement synchronisé DSM-CC. Lorsqu'elles sont acheminées dans un carrousel DSM-CC, les données de configuration du décodeur de métadonnées doivent impérativement être signalées. Toutefois, non définies dans la présente Spécification, les modalités de cette signalisation sont librement déterminées dans le cadre des applications.

2.12.4 Utilisation de paquets PES pour transporter des métadonnées

Les paquets PES constituent un mécanisme de transport des métadonnées en mode synchrone. L'horodateur de présentation (PTS) figurant dans l'en-tête des paquets PES permet d'associer les unités d'accès aux métadonnées à un instant déterminé de l'horloge STC, sans qu'il soit besoin de références temporelles dans les métadonnées. Cela suppose de savoir *a priori* comment associer les métadonnées à la base de temps pour l'acheminement. Des valeurs précises d'identification de flux (stream_id) et de type de flux (stream_type) sont attribuées aux fins de la signalisation des paquets PES qui acheminent les métadonnées (voir 2.12.9).

En cas d'utilisation de paquets PES ayant pour valeur stream_type 0x15 et pour valeur stream_id 0xFC pour le transport des métadonnées, il convient d'utiliser un enrobage d'unité d'accès aux métadonnées pour aligner les paquets PES et les unités d'accès aux métadonnées, à l'aide de cellules metadata_AU_cells. On disposera ainsi d'une indication d'accès aléatoire, dont la signification dépend du format des métadonnées, et d'un compteur de séquence de cellule permettant d'identifier la perte de cellules des unités d'accès aux métadonnées (metadata_AU_cells). Chaque unité d'accès aux métadonnées est acheminée et, s'il y a lieu, fragmentée en une ou plusieurs cellules metadata_AU_cells. Dans chaque paquet PES qui achemine des métadonnées, le premier octet PES_packet_data_byte doit être le premier octet d'une cellule Metadata_AU_cell. Pour chaque unité d'accès aux métadonnées contenue dans le même paquet PES, on recourt à l'horodateur PTS figurant dans l'en-tête PES. L'horodateur PTS indique l'heure à laquelle les unités d'accès aux métadonnées sont décodées instantanément et supprimées de la mémoire tampon B_n du décodeur STD. A noter que la relation entre une unité d'accès aux métadonnées décodée et le contenu audiovisuel n'est pas abordée dans la présente Spécification.

Un paquet PES peut contenir une seule cellule metadata_AU_cell. Cela est utile si une unité d'accès aux métadonnées n'entre dans aucun paquet PES pris isolément, auquel cas la fragmentation de l'unité d'accès aux métadonnées est assurée par la cellule metadata_AU_cell.

Lorsque les métadonnées sont acheminées par des paquets PES dans un flux de programme, et si ce flux fait l'objet d'un mappage, le mappage du flux de programme (PSM, *program stream map*) doit indiquer les paquets PES qui contiennent les métadonnées associées.

2.12.4.1 Enrobage de l'unité d'accès aux métadonnées

Il convient de recourir à l'enrobage des unités d'accès aux métadonnées lorsque ces unités sont acheminées dans des paquets PES ayant pour valeur stream_type 0x15 et pour valeur stream_id 0xFC ou dans des sections de téléchargement synchronisé DSM-CC ayant pour valeur stream_type 0x19. L'enrobage définit une structure constituée de plusieurs cellules Metadata_AU_cells concaténées. Le codage de la taille des métadonnées contenues dans chaque cellule

metadata_AU_cell permet de procéder, en l'absence de tout élément d'information, à une analyse syntaxique des métadonnées dans les récepteurs: l'analyseur synthaxique procède à l'extraction des métadonnées et les communique à un décodeur sans rien connaître *a priori* de leurs caractéristiques. La cellule Metadata_AU_cell doit être alignée sur le transport, c'est-à-dire que le premier octet de la charge utile du paquet PES ou de la section de téléchargement synchronisé DSM-CC doit être le premier octet d'une cellule Metadata_AU_cell.

Si elle n'entre pas intégralement dans une cellule metadata_AU_cell, une unité d'accès aux métadonnées doit être fragmentée en plusieurs cellules metadata_AU_cells. Chacune de ces cellules contient une indication de fragmentation (fragmentation_indication) indiquant qu'elle contient un fragment.

A chaque cellule Metadata_AU_cell contenue dans le même paquet PES ou la même section de téléchargement synchronisé est appliqué l'horodateur PTS tel que codé dans l'en-tête du paquet PES ou de la section de téléchargement synchronisé, selon le cas.

Tableau 2-96 – Enrobage de l'unité d'accès aux métadonnées

Syntaxe	Nombre de bits	Mnémonique

Metadata_AU_wrapper () {
for (i=0; i<N;i++){
Metadata_AU_cell ()
}
}		

Tableau 2-97 – Cellule d'unité d'accès aux métadonnées

Syntaxe	Nombre de bits	Mnémonique
Metadata_AU_cell () {		
metadata_service_id	**8**	**uimsbf**
sequence_number	**8**	**uimsbf**
cell_fragment_indication	**2**	**bslbf**
decoder_config_flag	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
reserved	**4**	**bslbf**
AU_cell_data_length	**16**	**uimsbf**
for (i=0; I<AU_cell_data_length;i++){		
AU_cell_data_byte	**8**	**bslbf**
}		
}		

metadata_service_id (identificateur de service de métadonnées) – Codé sur 8 bits, ce champ indique le service de métadonnées associé à l'unité d'accès aux métadonnées acheminée dans la cellule d'unité d'accès aux métadonnées considérée.

sequence_number (numéro de séquence) – Codé sur 8 bits, ce champ indique le numéro de séquence de la cellule metadata_AU_cell. Ce numéro est augmenté d'une unité pour chacune des cellules metadata_AU_cell successives constituant l'enrobage metadata_AU_wrapper, indépendamment de la valeur codée de l'identificateur metadata_service_id.

cell_fragment_indication (indication de fragmentation de cellule) – Codé sur 2 bits, ce champ véhicule des informations sur l'unité d'accès aux métadonnées acheminée dans la cellule metadata_AU_cell considérée, comme indiqué dans le Tableau 2-98.

Tableau 2-98 – Indication de fragment de cellule

Valeur	Description
11	Cellule unique acheminant une unité d'accès aux métadonnées complète.
10	Première cellule d'une série de plusieurs cellules contenant des données provenant d'une unité d'accès aux métadonnées.
01	Dernière cellule d'une série de plusieurs cellules contenant des données provenant d'une unité d'accès aux métadonnées.
00	Cellule d'une série de plusieurs cellules contenant des données provenant d'une unité d'accès aux métadonnées, à l'exception toutefois de la première et de la dernière.

random_access_indicator (indicateur d'accès aléatoire) – Lorsqu'il est mis à la valeur '1', ce champ codé sur 1 bit indique que les métadonnées acheminées dans la cellule metadata_AU_cell considérée constituent un point d'accès au service de métadonnées où peut être effectué le décodage en l'absence d'informations émanant de la cellule metadata_AU_cell précédente. La signification d'un point d'accès aléatoire est définie par le format des métadonnées.

decoder_config_flag (fanion de configuration de décodeur) – Codé sur 1 bit, ce champ indique la présence de l'information de configuration du décodeur dans l'unité d'accès aux métadonnées acheminée. A noter que ce champ n'empêche pas la présence de métadonnées dans l'unité d'accès voisine des données de configuration du décodeur.

AU_cell_data_length (longueur des données de cellule d'unité d'accès) – Codé sur 16 bits, ce champ indique le nombre d'octets AU_cell_data_bytes qui suivent immédiatement.

AU_cell_data_byte (octet de données de cellule d'unité d'accès) – Codé sur 8 bits, ce champ contient des octets contigus provenant d'une unité d'accès aux métadonnées.

2.12.5 Utilisation du protocole de téléchargement synchronisé DSM-CC pour le transport de métadonnées

Pour le transport en mode synchrone, on peut utiliser, en plus de paquets PES, le protocole de téléchargement synchronisé DSM-CC. En cas d'utilisation de sections de téléchargement synchronisé DSM-CC pour le transport de métadonnées, il convient de recourir à l'enrobage des unités d'accès aux métadonnées défini au 2.12.4.1 pour encapsuler les unités d'accès aux métadonnées. On disposera ainsi d'une indication d'accès aléatoire, dont la signification dépend du format des métadonnées, et d'un compteur de séquence de cellule permettant d'identifier la perte des cellules metadata_AU_cells. Dans chaque section de téléchargement synchronisé DSM-CC qui achemine des métadonnées, le premier octet de la charge utile doit être le premier octet d'une cellule Metadata_AU_cell. Pour chaque unité d'accès aux métadonnées contenue dans la même section de téléchargement synchronisé DSM-CC, on recourt à l'horodateur PTS figurant dans l'en-tête de cette section. Cet horodateur indique le moment auquel les unités d'accès aux métadonnées sont décodées instantanément et supprimées de la mémoire tampon B_n du décodeur STD. A noter que la relation entre une unité d'accès aux métadonnées décodée et le contenu audiovisuel n'est pas abordée dans la présente Spécification. L'attribution d'une valeur précise de type de flux (stream_type) (comme indiqué dans le Tableau 2-34) permet de signaler l'acheminement de métadonnées dans des sections de téléchargement synchronisé DSM-CC.

2.12.6 Utilisation de sections de métadonnées pour le transport de métadonnées

En cas de nécessité, des sections de métadonnées peuvent être utilisées pour le transport en mode asynchrone d'unités d'accès aux métadonnées en l'absence d'un mécanisme d'acheminement par carrousel. La syntaxe et la sémantique des sections de métadonnées sont définies dans le présent paragraphe. Comme indiqué dans le champ section_fragment_indication, chaque section de métadonnées doit acheminer une unité d'accès aux métadonnées complète ou une seule partie d'une unité d'accès aux métadonnées.

Pour le transport de sections de métadonnées, les unités d'accès aux métadonnées sont structurées en une ou plusieurs tables de métadonnées. Chacune de ces tables contient une ou plusieurs unités d'accès aux métadonnées complètes provenant d'un ou de plusieurs services de métadonnées. Théoriquement, le mécanisme de transport des tables de métadonnées est comparable à celui des tables de mappage de programme et des tables d'association de programmes. Chaque table de métadonnées peut être composée de plusieurs sections de métadonnées et contenir des métadonnées provenant de plusieurs services de métadonnées.

Des valeurs précises de type de flux (stream_type) et d'identificateur de table (table_id) sont attribuées aux sections de métadonnées. Celles-ci peuvent également acheminer des données de configuration du décodeur de métadonnées, assorties d'une valeur de description des métadonnées attribuée par le descripteur de configuration de ce décodeur.

Tableau 2-99 – Syntaxe de la section relative au transport de métadonnées

Syntaxe	Nombre de bits	Mnémonique
Metadata_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
decoder_config_flag	**1**	**bslbf**
metadata_section_length	**12**	**uimsbf**
metadata_service_id	**8**	**uimsbf**
reserved	**8**	**bslbf**
section_fragment_indication	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i=1; i<N; i++){		
metadata_byte	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		

table_id (identificateur de table) – Codé sur 8 bits, ce champ doit être mis à la valeur '0x06' pour chaque section de métadonnées.

section_syntax_indicator (indicateur de syntaxe de section) – Codé sur 1 bit, ce champ doit être mis à '1'.

private_indicator (indicateur de données privées) – Codé sur 1 bit, ce champ n'est pas défini dans la présente Spécification.

random_access_indicator (indicateur d'accès aléatoire) – Lorsqu'il est mis à la valeur '1', ce champ codé sur 1 bit indique que les métadonnées acheminées dans la section de métadonnées considérée représentent un point d'accès au service de métadonnées où peut s'effectuer le décodage en l'absence d'informations émanant des sections de métadonnées précédentes. La signification d'un point d'accès aléatoire est définie par le format des métadonnées.

decoder_config_flag (fanion de configuration du décodeur) – Lorsqu'il est mis à la valeur '1', ce champ codé sur 1 bit indique que l'information de configuration du décodeur est présente dans l'unité d'accès aux métadonnées acheminée dans la section de métadonnées considérée.

metadata_section_length (longueur de section de métadonnées) – Codé sur 12 bits, ce champ indique le nombre d'octets restant dans la section qui suit immédiatement le champ metadata_section_length, y compris le contrôle CRC. La valeur de ce champ ne doit pas dépasser 4093 (0xFFD).

metadata_service_id (identificateur de service de métadonnées) – Codé sur 8 bits, ce champ indique le service de métadonnées associé à l'unité d'accès aux métadonnées acheminée dans la section de métadonnées considérée. Chaque table de métadonnées peut contenir des métadonnées émanant de différents services de métadonnées.

section_fragment_indication (indication de fragment de section) – Codé sur 2 bits, ce champ véhicule l'information relative à la fragmentation de l'unité d'accès aux métadonnées acheminée dans la section de métadonnées considérée, correspondant au Tableau 2-100.

Tableau 2-100 – Indication de fragment de section

Valeur	Description
11	Section de métadonnées isolée acheminant une unité d'accès aux métadonnées complète.
10	Première section de métadonnées d'une série de plusieurs sections de métadonnées contenant des données provenant d'une unité d'accès aux métadonnées.
01	Dernière section de métadonnées d'une série de plusieurs sections de métadonnées contenant des données provenant d'une unité d'accès aux métadonnées.
00	Section de métadonnées d'une série de plusieurs sections de métadonnées contenant des données provenant d'une unité d'accès aux métadonnées, à l'exception toutefois de la première et de la dernière.

version_number (numéro de version) – Codé sur 5 bits, ce champ est le numéro de version de l'ensemble de la table de métadonnées. Le numéro de version doit augmenter d'une unité modulo 32 à chaque fois que les informations contenues dans la table de métadonnées sont modifiées. Lorsque le champ current_next_indicator est mis à '1', le numéro de version doit être celui de la table de métadonnées applicable à ce moment-là. Lorsque le champ current_next_indicator est mis à '0', le numéro de version doit être celui de la table de métadonnées suivante applicable.

current_next_indicator (indicateur d'applicabilité de la table suivante) – Lorsqu'il est mis à '1', ce champ codé sur 1 bit indique que la table de métadonnées envoyée est applicable à ce moment-là. Lorsqu'il est mis à '0', il indique que la table envoyée n'est pas encore applicable et doit être la prochaine table de métadonnées à devenir valide.

section_number (numéro de section) – Codé sur 8 bits, ce champ indique le numéro de la section de métadonnées. Le numéro de la première section dans la table de métadonnées doit être 0x00. Il doit augmenter d'une unité à chaque nouvelle section contenue dans la table de métadonnées.

last_section_number (numéro de la dernière section) – Codé sur 8 bits, ce champ indique le numéro de la dernière section (c'est-à-dire celle dont le numéro de section est le plus élevé) de la table de métadonnées complète dont cette section fait partie.

metadata_byte (octet de métadonnées) – Codé sur 8 bits, ce champ contient les octets contigus d'une unité d'accès aux métadonnées.

CRC_32 (contrôle CRC sur 32 bits) – Codé sur 32 bits, ce champ doit contenir la valeur de contrôle CRC qui donne des zéros à la sortie des registres du décodeur défini à l'Annexe A après traitement de toute la section de métadonnées (metadata_section).

2.12.7 Utilisation du carrousel de données DSM-CC pour le transport de métadonnées

Les outils DSM-CC définis dans l'ISO/CEI 13818-6 pour les carrousels de données peuvent être utilisés dans le cas où un mécanisme d'acheminement par carrousel doit être utilisé sans qu'il soit nécessaire d'exprimer l'organisation hiérarchique de la structure des métadonnées dans le mécanisme de transport. Des informations relatives au carrousel dans lequel les métadonnées sont contenues sont incluses dans le descripteur de métadonnées défini aux 2.6.60 et 2.6.62. L'attribution d'une valeur de type de flux (stream_type) déterminée permet de signaler l'acheminement des métadonnées dans le carrousel de données DSM-CC. A noter que la signalisation des services de métadonnées dans un carrousel de données DSM-CC est obligatoire, bien qu'elle ne soit pas définie dans la présente Spécification.

2.12.8 Utilisation du carrousel d'objets DSM-CC pour le transport de métadonnées

Dans le cas où il est nécessaire d'utiliser un mécanisme d'acheminement par carrousel qui permette d'exprimer l'organisation hiérarchique de la structure des métadonnées dans le mécanisme de transport, les outils DSM-CC et les structures de fichier définies dans l'ISO/CEI 13818-6 pour les carrousels d'objets d'utilisateur à utilisateur peuvent être utilisés. Ces structures de fichier indiquent les outils à utiliser pour structurer les métadonnées comme on le jugera bon pour procéder à une analyse syntaxique efficace des métadonnées et pour exprimer leur organisation hiérarchique. Les informations nécessaires pour identifier le carrousel dans lequel les métadonnées sont contenues figurent dans le descripteur de métadonnées défini aux 2.6.60 et 2.6.61. Il peut s'agir de l'identificateur IOP:IOR() défini aux 11.3.1 et 5.7.2.3 de l'ISO/CEI 13818-6 DSM-CC. L'attribution d'une valeur de type de flux (stream_type) déterminée signale l'acheminement des métadonnées dans le carrousel d'objets DSM-CC. A noter que la signalisation des services de métadonnées dans un carrousel d'objets DSM-CC est obligatoire, bien qu'elle ne soit pas définie dans la présente Spécification.

2.12.9 Signalisation relative aux métadonnées

La signalisation relative aux métadonnées recouvre quatre domaines distincts:

- la signalisation des services et des flux de métadonnées;
- la signalisation du contenu que doit utiliser un système de métadonnées;
- l'association des métadonnées au contenu;
- la signalisation des données de configuration du décodeur.

2.12.9.1 Signalisation des services et des flux de métadonnées

L'acheminement de métadonnées est signalé par une valeur de type de flux (stream_type) de 0x15 à 0x19 indiquant parmi les cinq méthodes décrites aux 2.12.4 à 2.12.8 celle qui est utilisée pour le transport des métadonnées et, s'il y a lieu, par une valeur d'identificateur de flux (stream_id) de 0xFC indiquant un flux de métadonnées.

Pour identifier de manière univoque les différents services de métadonnées, le système de transport attribue à chacun d'entre eux une valeur d'identificateur de service de métadonnées (metadata_service_id); la valeur ainsi attribuée doit être propre au flux de transport ou de programme qui achemine le service de métadonnées. Si les métadonnées sont acheminées dans des paquets PES ayant une valeur d'identificateur de flux (stream_id) de 0xFC, dans des sections de métadonnées ou dans des sections de téléchargement synchronisé ISO/CEI 13818-6, la valeur metadata_service_id attribuée est signalée expressément dans l'en-tête de la cellule metadata_AU_cell ou de la section de métadonnées. En cas d'utilisation d'un carrousel ISO/CEI 13818-6 pour acheminer les métadonnées, la signalisation des services de métadonnées relève alors de l'application. Le descripteur de métadonnées indique le format des métadonnées, fournit les

informations relatives aux données de configuration du décodeur et est lié au service de métadonnées du fait qu'il achemine des informations relatives au service de métadonnées auquel il est associé.

2.12.9.2 Signalisation du contenu que doit utiliser un système de métadonnées

Les 2.6.56 et 2.6.57 définissent un descripteur d'étiquetage de contenu qui peut être utilisé pour attribuer une référence propre au format d'application des métadonnées – l'enregistrement content_reference_id_record – à des contenus audiovisuels ou autres acheminés sur un flux de transport ou de programme MPEG-2. L'enregistrement content_reference_id_record peut être utilisé par le système de métadonnées comme une étiquette désignant le contenu considéré. Celui-ci peut représenter, par exemple, un programme ou un flux ou des segments de ceux-ci. Le descripteur d'étiquetage de contenu fournit en outre des informations relatives à la base de temps pour le contenu servant de référence temporelle émanant des métadonnées, dont le décalage de temps constant entre la base de temps pour les métadonnées et la base de temps appliquée pour le contenu. Le descripteur permet l'acheminement de données privées. Le format d'application des métadonnées (metadata_application_format) peut définir des restrictions applicables à l'enregistrement content_reference_record, telle une limitation de la durée de validité de cet enregistrement.

2.12.9.3 Association des métadonnées au contenu

Les 2.6.58 et 2.6.59 définissent le descripteur de pointeur de métadonnées permettant d'associer un service de métadonnées pris isolément aux contenus audiovisuels ou autres d'un flux UIT-T H.222.0 | ISO/CEI 13818-1. Les métadonnées sont associées au contenu dans le contexte défini par l'emplacement du descripteur. Dans un flux de transport, le descripteur peut figurer dans la table PMT dans la boucle de descripteur pour le programme ou un flux élémentaire, mais il peut aussi figurer dans des tableaux non définis dans la présente Spécification, tels que des tableaux décrivant des bouquets de services de diffusion.

Le descripteur de pointeur de métadonnées indique à partir du contexte du contenu le service de métadonnées associé à ce contenu. Le descripteur indique la valeur de l'identificateur metadata_service_id qui est attribuée au service de métadonnées associé, ainsi qu'un ou plusieurs emplacements des métadonnées associées. Celles-ci peuvent se trouver, par exemple, dans le même flux de transport que le contenu, ou dans un autre flux de transport, mais également dans un emplacement autre qu'un flux UIT-T H.222.0 | ISO/CEI 13818-1, tel que l'Internet.

2.12.9.4 Signalisation des données de configuration du décodeur

Pour décoder les métadonnées, il peut être nécessaire d'avoir accès aux données de configuration du décodeur de métadonnées. Si elles sont nécessaires, les données de configuration du décodeur doivent être contenues dans un des services de métadonnées du même flux de programme UIT-T H.222.0 | ISO/CEI 13818-1 que le service de métadonnées. Si les données de configuration du décodeur sont nécessaires pour décoder un service de métadonnées, le descripteur de métadonnées achemine ces données ou communique les informations permettant d'extraire les données de configuration du décodeur du même service de métadonnées ou d'un autre. Un tel autre service peut être trouvé dans un flux de transport en recherchant dans la table PMT un descripteur de métadonnées (metadata_descriptor) dont l'identificateur de service de métadonnées (metadata_service_id) soit tel que spécifié dans le champ decoder_config_metadata_service_id (et présentant le même format de métadonnées (metadata_format) et le même format d'application de métadonnées (metadata_application format)).

2.12.9.5 Aperçu général de la signalisation des métadonnées

La Figure 2-9 donne un exemple de signalisation des métadonnées, dans lequel un programme unique, le "programme de contenu", achemine le contenu (ou nature), les métadonnées étant acheminées dans un programme séparé, le "programme de métadonnées". Dans cet exemple, le programme de métadonnées et le programme de contenu coexistent sur le même flux de transport.

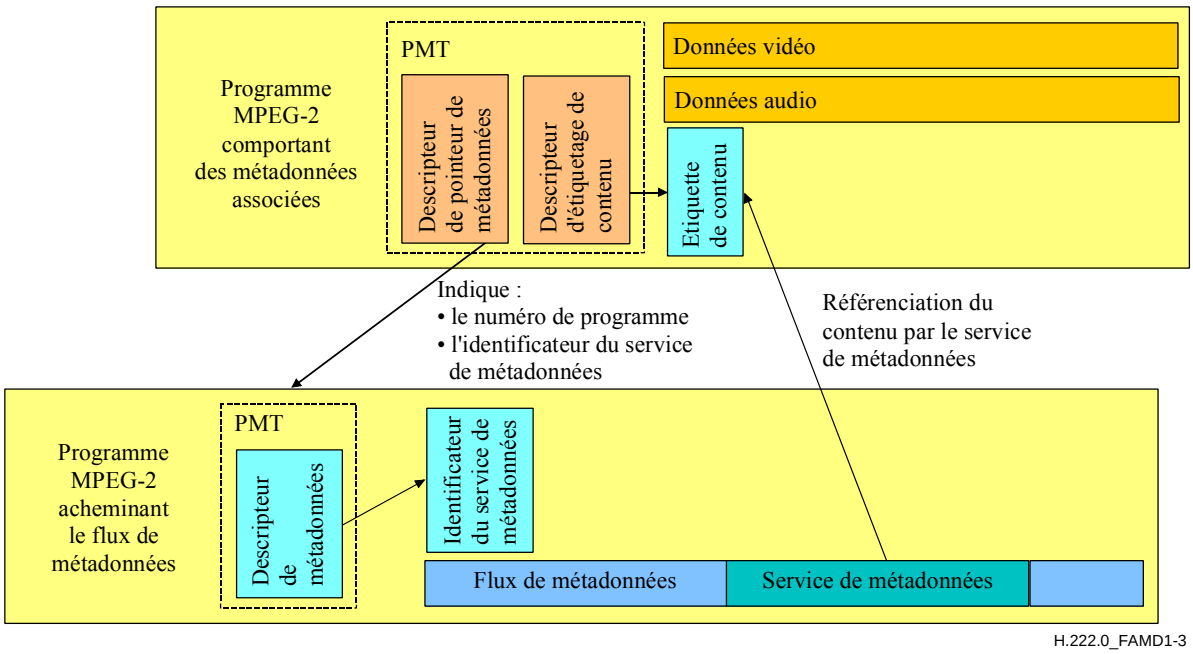


Figure 2-9 – Signalisation et référenciation des métadonnées

Le programme de contenu comporte deux descripteurs relatifs aux métadonnées: le descripteur d'étiquetage de contenu (content_labeling) et le descripteur du pointeur de métadonnées (metadata_pointer). Le descripteur content_labeling associe une étiquette, représentée sur la figure par la case "étiquette de contenu" et codée dans les champs d'identificateur de référence de contenu (content_reference_id) du descripteur, au contenu. Cette étiquette peut alors être utilisée par le service de métadonnées, pour indiquer la nature de l'intégralité ou d'une partie du contenu, ou par un segment temporel déterminé. Par exemple, le descripteur content_labeling peut comporter l'étiquette "Nouvelles du 1/1/02", les métadonnées pouvant alors se rapporter à un extrait des "Nouvelles du 1/1/02", en indiquant par exemple l'heure précise de passage de cet extrait.

Le descripteur du pointeur de métadonnées indique où se trouve le service de métadonnées correspondant au contenu considéré. Dans l'exemple présenté ici, les métadonnées sont acheminées dans un programme séparé, mais elles pourraient tout aussi bien être acheminées dans le même programme que le contenu, ou communiquées par d'autres moyens non abordés dans la présente Spécification, par exemple depuis une adresse URL. Ce descripteur indique également la valeur de l'identificateur du service de métadonnées qui est attribuée au service de métadonnées. Cela est indispensable car un flux de métadonnées pourrait acheminer divers services de métadonnées pour de nombreux programmes différents et chaque programme doit pouvoir identifier de manière univoque son propre service de métadonnées.

Dans le programme de métadonnées, le descripteur des métadonnées indique à quel service de métadonnées d'un flux de métadonnées il s'applique. S'il est utilisé, le descripteur des métadonnées précise de manière détaillée où trouver l'information de configuration du décodeur.

En identifiant un descripteur de pointeur de métadonnées dans la table PMT par décodage du programme de contenu, le récepteur extrait le descripteur de métadonnées du programme de métadonnées. Si nécessaire, les données de configuration du décodeur sont extraites en premier, le décodeur étant ensuite configuré en conséquence, après quoi le décodage du service de métadonnées peut commencer.

2.12.10 Modèle STD applicable aux métadonnées

Le modèle STD définit les contraintes normatives applicables aux flux UIT-T H.222.0 | ISO/CEI 13818-1 qui acheminent des métadonnées. Pour décoder les métadonnées figurant dans le descripteur STD, on utilise les modèles T-STD et P-STD ordinaires, avec une mémoire tampon B_n, à l'entrée de laquelle le débit des métadonnées est de Rx_n, le débit des métadonnées passant ensuite à $R_{metadata}$ à la sortie de la mémoire tampon B_n et jusqu'à l'entrée dans le décodeur de métadonnées, puis à $D_{metadata}$ dans ce décodeur (Voir la Figure 2-10).

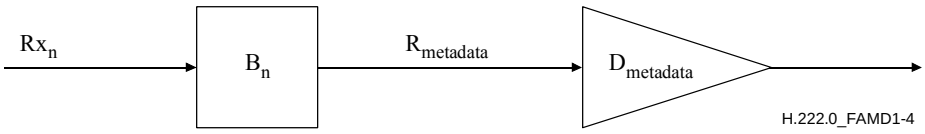


Figure 2-10 – Décodage des métadonnées dans le descripteur STD

Les métadonnées entrent dans la mémoire tampon B_n au débit de Rx_n. Dans le décodeur P-STD, le débit Rx_n est égal au débit du flux de programmes. Dans le décodeur T-STD, le débit Rx_n est le débit à la sortie de la mémoire TB_n et est égal au débit défini par le champ metadata_input_ leak_rate du descripteur STD de métadonnées. La capacité BS_n de la mémoire tampon B_n est égale à la capacité définie dans le champ metadata_buffer_size du descripteur STD de métadonnées. En cas d'acheminement en mode synchrone, le décodage des métadonnées est instantané et s'effectue sous la surveillance des horodateurs PTS. Au moment où le décodage s'effectue, c'est-à-dire lorsque l'horloge STC indique la même heure que l'horodateur PTS, les métadonnées associées sont instantanément extraites de la mémoire B_n. En cas d'acheminement en mode asynchrone, l'extraction des métadonnées de la mémoire B_n se fait à un débit $R_{metadata}$ égal au débit défini par le champ metadata_output_leak_rate du descripteur STD de métadonnées. La mémoire B_n ne doit pas déborder.

A noter que le modèle STD définit les contraintes applicables à l'acheminement des métadonnées, sans toutefois définir de contraintes applicables à la base de temps utilisée dans les métadonnées.

2.13 Acheminement des données ISO 15938

2.13.1 Introduction

L'acheminement de métadonnées sur un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1 tel que défini au 2.12 permet d'acheminer des données ISO 15938 par un codage approprié du champ metadata_format. Dans ce paragraphe, un exemple précis est défini pour les besoins du transport de données ISO 15938. L'acheminement de données ISO 15938 doit satisfaire à chacune des conditions définies au 2.12. Toutefois, les conditions définies dans ce paragraphe sont également applicables au transport de données ISO 15938.

2.13.2 Données de configuration du décodeur ISO 15938

Le décodage de données ISO 15938 est subordonné à l'obtention des données de configuration du décodeur. Par conséquent, en cas d'acheminement de données ISO 15938 dans un flux UIT-T H.222.0 | ISO/CEI 13818-1, le descripteur des métadonnées doit signaler l'acheminement des données de configuration du décodeur associées dans le même flux UIT-T H.222.0 | ISO/CEI 13818-1 en codant les fanions decoder_config_flags à la valeur '001', '010', '011' ou '100'.

2.14 Acheminement de données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10

2.14.1 Introduction

On définit ici l'acheminement d'un flux élémentaire Rec. UIT-T H.264 | ISO/CEI 14496-10 dans des systèmes Rec. UIT-T H.222.0 | ISO/CEI 13818-1, tant pour les flux de programme que pour les flux de transport. Généralement, un flux Rec. UIT-T H.264 | ISO/CEI 14496-10 sera un élément d'un programme Rec. UIT-T H.222.0 | ISO/CEI 13818-1, comme défini par la table PMT dans le cas d'un flux de transport et par le mappage PSM dans le cas d'un flux de programme. La gestion de l'acheminement et des tampons des flux vidéo AVC est définie au moyen des paramètres existants de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (par exemple, PTS et DTS) et des informations présentes dans un flux vidéo AVC.

L'acheminement d'un flux vidéo AVC dans un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1 définit un mappage précis entre les paramètres STD et les paramètres HRD qui peuvent être présents dans le flux vidéo AVC. Des conditions sont définies pour la présence de paramètres HRD dans le flux vidéo AVC pour faire en sorte qu'il soit possible de vérifier si chaque condition STD est remplie pour chaque flux vidéo AVC acheminé dans un flux de transport ou un flux de programme.

NOTE 1 – Bien que l'information de synchronisation présente dans le flux vidéo AVC puisse ne pas être basée sur l'utilisation d'une horloge à 90 kHz, les horodates PTS et DTS doivent être exprimées dans l'unité valant 90 kHz.

Lorsqu'un flux Rec. UIT-T H.264 | ISO/CEI 14496-10 est acheminé dans un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1, les données Rec. UIT-T H.264 | ISO/CEI 14496-10 codées seront contenues dans des paquets PES. Les données Rec. UIT-T H.264 | ISO/CEI 14496-10 codées seront conformes au format de flux d'octets défini dans l'Annexe B de la Rec. UIT-T H.264 | ISO/CEI 14496-10, avec les contraintes suivantes:

- Chaque unité d'accès AVC doit contenir une unité NAL de délimitation d'unité d'accès;

  NOTE 2 – Selon la Rec. UIT-T H.264 | ISO/CEI 14496-10, si une unité NAL de délimitation d'unité d'accès est présente, elle doit être la première unité NAL figurant dans une unité d'accès AVC. L'unité NAL de délimitation d'unité d'accès facilite la détection de la limite entre images; elle évite de devoir traiter le contenu des en-têtes de tranche et elle est surtout utile dans les profils de base et étendu où l'ordre des tranches peut être arbitraire.

- Chaque unité NAL de flux d'octets qui achemine le délimiteur d'unité d'accès doit contenir exactement un élément de syntaxe zero_byte.

  NOTE 3 – La syntaxe et la sémantique des unités NAL de flux d'octets sont définies dans l'Annexe B de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

- Tous les ensembles de paramètres de séquence et d'image (SPS et PPS) nécessaires au décodage du flux vidéo AVC doivent être présents dans ce flux vidéo AVC.

  NOTE 4 – Selon la Rec. UIT-T H.264 | ISO/CEI 14496-10, les paramètres SPS et PPS peuvent être remis par des moyens extérieurs. La présente Spécification ne prend pas en charge une telle remise et nécessite donc que les paramètres SPS et PPS soient acheminés dans le flux vidéo AVC.

- Chaque séquence vidéo AVC qui contient des paramètres hrd_parameters() avec le fanion low_delay_hrd_flag mis à '1', doit acheminer des paramètres VUI dans lesquels le fanion timing_info_present_flag sera mis à '1'.

  NOTE 5 – Si le fanion low_delay_hrd_flag est mis à '1', un sous-remplissage du tampon est permis dans le décodeur STD; voir 2.14.3 et 2.14.4. Le fait de mettre le fanion timing_info_present_flag à '1' garantit que le flux vidéo AVC contient suffisamment d'informations pour pouvoir déterminer l'instant de sortie du tampon DPB et l'instant de retrait du tampon CPB des unités d'accès AVC, y compris en cas de sous-remplissage.

Pour fournir des informations d'affichage spécifiques telles que le rapport largeur/hauteur, il est fortement recommandé que chaque flux vidéo AVC achemine des paramètres VUI avec suffisamment d'informations pour garantir que le flux vidéo AVC décodé puisse être affiché correctement par les récepteurs.

2.14.2 Acheminement dans des paquets PES

Les données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 sont acheminées dans des paquets PES sous la forme d'octets PES_packet_data_bytes, l'une des 16 valeurs de stream_id attribuées à la vidéo étant utilisée et le flux vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10 étant signalé au moyen de la valeur de type de flux attribuée dans la table PMT ou le mappage PSM (voir Tableau 2-34). Le niveau le plus élevé qui puisse se produire dans un flux vidéo AVC et le profil auquel l'ensemble du flux est conforme doivent être signalés au moyen du descripteur vidéo AVC. Si un descripteur vidéo AVC est associé à un flux vidéo AVC, ce descripteur sera acheminé dans la boucle de descripteur de l'entrée de flux élémentaire figurant dans la table de mappage de programme dans le cas d'un flux de transport ou dans le mappage de flux de programme, lorsqu'il est présent, dans le cas d'un flux de programme. La Rec. UIT-T H.222.0 | ISO/CEI 13818-1 ne spécifie pas la présentation des flux Rec. UIT-T H.264 | ISO/CEI 14496-10 dans le contexte d'un programme.

Pour la mise en paquets PES, aucune contrainte d'alignement de données particulière ne s'applique. Pour la synchronisation et la gestion STD, des horodates PTS et, si nécessaire, des horodates DTS sont codées dans l'en-tête du paquet PES qui achemine les données de flux élémentaire vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10. Pour le codage des horodates PTS et DTS, les contraintes et la sémantique définies au 2.4.3.7 et au 2.7 s'appliquent.

2.14.3 Extensions du décodeur STD

2.14.3.1 Extensions du décodeur T-STD

Le décodeur T-STD contient un tampon de transport TB_n et un tampon de multiplexage MB_n avant le tampon EB_n pour le décodage de chaque flux élémentaire vidéo n Rec. UIT-T H.264 | ISO/CEI 14496-10. Voir la Figure 2-11.

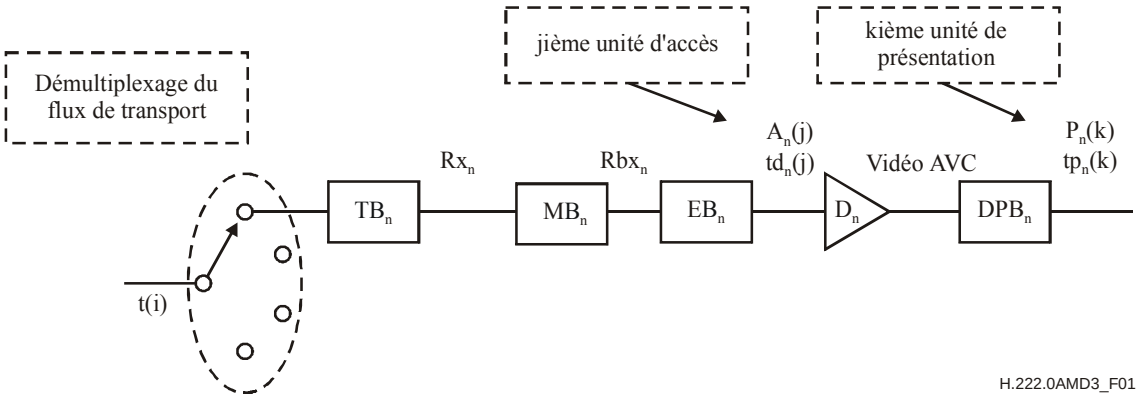


Figure 2-11 – Extensions du décodeur T-STD pour les données vidéo
Rec. UIT-T H.264 | ISO/CEI 14496-10

Gestion du tampon DPB_n

L'acheminement de données vidéo AVC sur un flux Rec. UIT-T H.222.0 | ISO/CEI 13818-1 n'a pas d'effet sur la taille du tampon DPB_n. Pour le décodage d'un flux vidéo AVC dans le décodeur STD, la taille du tampon DPB_n est telle que définie dans la Rec. UIT-T H.264 | ISO/CEI 14496-10. Le tampon DPB doit être géré comme indiqué dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10 (paragraphes C.2 et C.4). Une unité d'accès AVC décodée entre instantanément dans le tampon DPB_n après son décodage, c'est-à-dire à l'instant de son retrait du tampon CPB. Une unité d'accès AVC décodée est présentée à l'instant de sortie du tampon DPB. Si le flux vidéo AVC fournit des informations insuffisantes pour déterminer l'instant de retrait du tampon CPB et l'instant de sortie du tampon DPB des unités d'accès AVC, ces instants doivent être déterminés dans le décodeur STD à partir des horodates PTS et DTS de la manière suivante:

1) l'instant du retrait du tampon CPB de l'unité d'accès AVC n est l'instant indiqué par DTS(n), où DTS(n) est la valeur de l'horodate DTS de l'unité d'accès AVC n;
2) l'instant de sortie du tampon DPB de l'unité d'accès AVC n est l'instant indiqué par PTS(n), où PTS(n) est la valeur de l'horodate PTS de l'unité d'accès AVC n.

NOTE 1 – Les séquences vidéo AVC dans lesquelles le fanion low_delay_hrd_flag des paramètres hrd-parameters() est mis à 1 acheminent suffisamment d'informations pour déterminer l'instant de sortie du tampon DPB et l'instant de retrait du tampon CPB de chaque unité d'accès AVC. Ainsi, pour les unités d'accès AVC pour lesquelles un sous-remplissage du décodeur STD peut se produire, l'instant de retrait du tampon CPB et l'instant de sortie du tampon DPB sont définis par les paramètres HRD et non par les horodates DTS et PTS.

Gestion des tampons TB_n, MB_n et EB_n

L'entrée dans le tampon TB_n et sa taille TBS_n sont spécifiées au 2.4.2.3. Pour les tampons MB_n et EB_n, et pour le débit Rx_n entre TB_n et MB_n, ainsi que le débit Rbx_n entre MB_n et EB_n les contraintes suivantes s'appliquent à l'acheminement d'un flux Rec. UIT-T H.264 | ISO/CEI 14496-10:

Taille EBS_n du tampon EB_n:

EBS_n = cpb_size

où la taille cpb_size est la taille Cpbsize[cpb_cnt_minus1] du tampon CPB pour le format de flux d'octets signalé dans les paramètres hrd_parameters() NAL acheminés dans les paramètres VUI du flux vidéo AVC. En l'absence de paramètres hrd_parameters() NAL dans le flux vidéo AVC, la taille cpb_size doit être la taille définie comme étant 1200 × MaxCPB dans l'Annexe A de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour le niveau du flux vidéo AVC.

Taille MBS_n du tampon MB_n:

$$MBS_n = BS_{mux} + BS_{oh} + 1200 \times MaxCPB[niveau] - cpb_size$$

où BS_{oh}, la taille du tampon pour les préfixes de paquet, est définie comme étant:

$$BS_{oh} = (1/750) \text{ seconde} \times \max\{1200 \times MaxBR[niveau], 2\,000\,000 \text{ bit/s}\}$$

et BS_{mux}, la taille du tampon de multiplexage supplémentaire, est définie comme étant:

$$BS_{mux} = 0{,}004 \text{ seconde} \times \max\{1200 \times MaxBR[niveau], 2\,000\,000 \text{ bit/s}\}$$

où MaxCPB[niveau] et MaxBR[niveau] sont définis pour le format de flux d'octets au Tableau A.1 (Limites de niveau) de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour le niveau du flux vidéo AVC, et

où cpb_size est la taille CpbSize[cpb_cnt_minus1] du tampon CPB pour le format de flux d'octets signalé dans les paramètres hrd_parameters() NAL acheminés dans des paramètres VUI du flux vidéo AVC. Si les paramètres hrd_parameters() NAL sont absents du flux vidéo AVC, la taille cpb_size sera la taille 1200 × MaxCPB définie dans l'Annexe A de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour le niveau du flux vidéo AVC.

Débit Rx_n:

En l'absence de données dans le tampon TB_n, Rx_n est égal à zéro.

Sinon, Rx_n = bit_rate

où bit_rate est le débit binaire BitRate[cpb_cnt_minus1] du flux de données dans le tampon CPB pour le format de flux d'octets signalé dans les paramètres hrd_parameters() NAL acheminés dans des paramètres VUI du flux vidéo AVC. Si ces paramètres hrd_parameters() NAL sont absents du flux vidéo AVC, le débit bit_rate sera de 1200 × MaxBR[niveau], défini dans l'Annexe A de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour le niveau du flux vidéo AVC.

Transfert entre tampons MB_n et EB_n

Si le descripteur AVC_timing_and_HRD_descriptor est présent avec le fanion hrd_management_valid_flag mis à '1', le transfert de données du tampon MB_n au tampon EB_n suivra le mécanisme défini par le décodeur HRD pour l'arrivée des données dans le tampon CPB comme défini à l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

Sinon, on utilisera la méthode avec fuite pour transférer les données du tampon MB_n au tampon EB_n, à savoir:

Débit Rbx_n:

$$Rbx_n = 1200 \times MaxBR[niveau]$$

où MaxBR[niveau] est défini pour le format de flux d'octets au Tableau A.1 (Limites de niveau) de la Rec. UIT-T H.264 | ISO/CEI 14496-10 pour chaque niveau.

S'il y a des données de charge utile de paquet PES dans le tampon MB_n, et si le tampon EB_n n'est pas plein, la charge utile du paquet PES est transférée du tampon MB_n au tampon EB_n à un débit égal à Rbx_n. Si le tampon EB_n est plein, les données ne sont pas retirées du tampon MB_n. Quand un octet de données est transféré du tampon MB_n au tampon EB_n, tous les octets d'en-tête de paquet PES qui se trouvent dans le tampon MB_n et qui précèdent cet octet sont instantanément retirés et rejetés. S'il n'y a aucune donnée de charge utile de paquet PES dans le tampon MB_n, aucune donnée n'est retirée de ce tampon MB_n. Toutes les données qui entrent dans le tampon MB_n en sortent. Tous les octets de données de charge utile de paquet PES entrent instantanément dans le tampon EB_n lorsqu'ils quittent le tampon MB_n.

Retrait d'unités d'accès AVC du tampon EB_n

Chaque unité d'accès AVC $A_n(j)$ présente dans le tampon EB_n en est retirée immédiatement à l'instant $td_n(j)$. Cet instant de décodage $td_n(j)$ est spécifié par l'horodate DTS ou à partir de l'instant de retrait du tampon CPB, tel que déterminé à partir des informations contenues dans le flux vidéo AVC.

Retard introduit par le décodeur STD

La durée totale du passage de toute donnée Rec. UIT-T H.264 | ISO/CEI 14496-10 autre qu'une image fixe AVC dans les tampons TB_n, MB_n et EB_n des décodeurs modèles du système devra vérifier $td_n(j) - t(i) \leq 10$ secondes pour tout j et tous les octets i de l'unité d'accès AVC $A_n(j)$.

La durée totale du passage d'une image fixe AVC dans les tampons TB_n, MB_n, et EB_n des décodeurs modèles du système devra vérifier $td_n(j) - t(i) \leq 60$ secondes pour tout j et tous les octets i de l'unité d'accès AVC $A_n(j)$.

Conditions de gestion des tampons

Les flux de transport seront construits de telle manière que les conditions suivantes de gestion des tampons soient satisfaites:

- aucun dépassement de capacité du tampon TB_n ne se produira et ce tampon sera vide au moins une fois par seconde;
- aucun dépassement de capacité des tampons MB_n, EB_n et DPB_n ne se produira;
- aucun sous-remplissage du tampon EB_n ne se produira sauf lorsque des paramètres VUI sont présents pour la séquence vidéo AVC avec le fanion low_delay_hrd_flag mis à '1'. Un sous-remplissage du

tampon EB_n se produit pour l'unité d'accès AVC $A_n(j)$ lorsqu'un ou plusieurs octets de $A_n(j)$ ne sont pas présents dans le tampon EB_n à l'instant du décodage $td_n(j)$.

NOTE 2 – Un flux vidéo AVC peut acheminer une information permettant de déterminer la conformité du flux vidéo AVC avec le décodeur HRD, comme spécifié dans l'Annexe C de la Rec. UIT-T H.264 | ISO/CEI 14496-10. La présence de cette information peut être signalée dans un flux de transport au moyen du descripteur de décodeur HRD et de synchronisation AVC avec le fanion hrd_management_valid_flag mis à '1'. Indépendamment de la présence de cette information, la conformité d'un flux vidéo AVC avec le décodeur T-STD garantit que les conditions de gestion du tampon CPB_n du décodeur HRD sont remplies lorsque chaque octet du flux vidéo AVC est introduit et retiré de ce tampon exactement à l'instant où l'octet est introduit et retiré du tampon EB_n du décodeur T-STD.

2.14.3.2 Extensions du décodeur P-STD

Le décodeur P-STD utilisé pour le décodage d'un flux élémentaire Rec. UIT-T H.264 | ISO/CEI 14496-10 inclut un tampon de multiplexage B_n et un décodeur D_n suivi d'un tampon DPB_n; voir la Figure 2-12. Pour chaque flux vidéo AVC n, la taille BS_n de la mémoire B_n du décodeur P-STD est définie par le champ P-STD_buffer_size de l'en-tête de paquet PES.

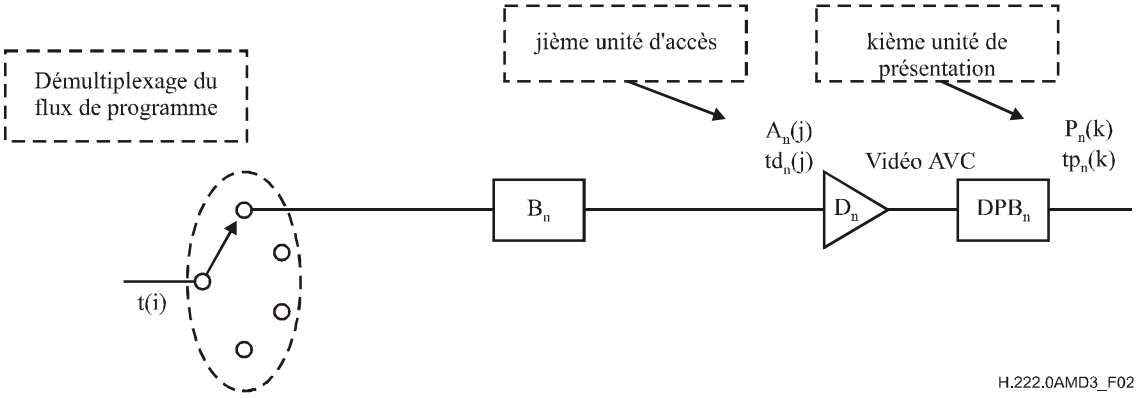


Figure 2-12 – Extensions du décodeur P-STD pour les données vidéo Rec. UIT-T H.264 | ISO/CEI 14496-10

Gestion du tampon DPB_n

Le tampon DPB_n sera géré exactement de la même manière que celui du décodeur T-STD; voir le 2.14.3.1.

Gestion du tampon B_n

Les données de l'unité d'accès AVC entrent dans le tampon B_n comme spécifié au 2.5.2.2. A l'instant $td_n(j)$, l'unité d'accès AVC $A_n(j)$ est décodée et instantanément retirée du tampon B_n. L'instant de décodage $td_n(j)$ est spécifié par l'horodate DTS ou par l'instant du retrait du tampon CPB, déterminé à partir de l'information contenue dans le flux vidéo AVC. Après le décodage, l'unité d'accès AVC entre instantanément dans le tampon DPB_n ou elle sort sans entrer dans le tampon DPB_n, conformément aux règles spécifiées dans la Rec. UIT-T H.264 | ISO/CEI 14496-10.

Retard introduit par le décodeur STD

La durée totale du passage de toute donnée Rec. UIT-T H.264 | ISO/CEI 14496-10 autre qu'une image fixe AVC dans le tampon B_n des décodeurs modèles du système devra vérifier $td_n(j) - t(i) \leq 10$ secondes pour tout j et tous les octets i de l'unité d'accès AVC $A_n(j)$.

La durée totale du passage d'une image fixe AVC dans le tampon B_n des décodeurs modèles du système devra vérifier $td_n(j) - t(i) \leq 60$ secondes pour tout j et tous les octets i de l'unité d'accès AVC $A_n(j)$.

Conditions de gestion de tampon

Les flux de transport seront construits de telle manière que les conditions suivantes de gestion de tampons soient satisfaites:

- aucun dépassement de capacité du tampon B_n ne se produira;
- aucun sous-remplissage du tampon B_n ne se produira, sauf lorsque des paramètres VUI sont présents pour la séquence vidéo AVC avec le fanion low_delay_hrd_flag mis à '1' ou lorsque l'état trick_mode est vrai. Un sous-remplissage du tampon B_n se produit pour l'unité d'accès AVC $A_n(j)$ lorsque un ou plusieurs octets de $A_n(j)$ ne sont pas présents dans le tampon B_n à l'instant du décodage $td_n(j)$.

Annexe A

Modèle de décodeur pour le contrôle CRC

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

A.0 Décodeur modèle de contrôle CRC

Le décodeur modèle du contrôle CRC sur 32 bits est spécifié dans la Figure A.1.

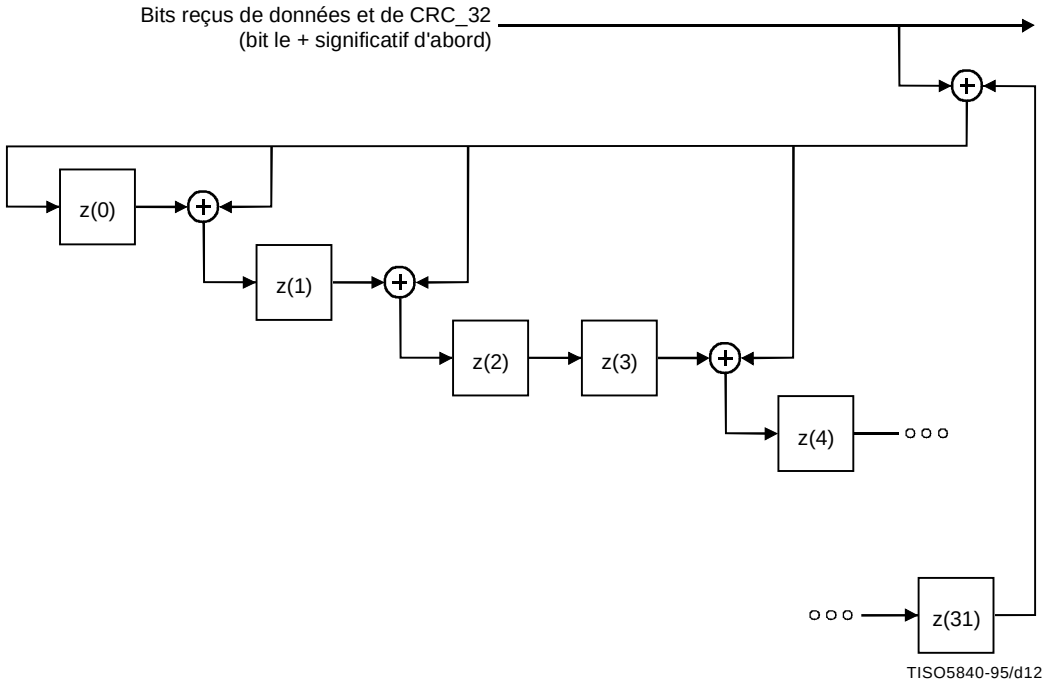


Figure A.1 – Décodeur modèle du contrôle CRC sur 32 bits

Le décodeur du contrôle CRC sur 32 bits fonctionne au niveau des éléments binaires. Il se compose de 14 mélangeurs '+' et de 32 éléments de délai z(i). Le signal d'entrée dans le décodeur CRC est mélangé au signal sortant de l'élément z(31) et le résultat est injecté dans l'élément z(0) ainsi qu'à l'une des entrées de chaque mélangeur restant. L'autre entrée de chaque mélangeur restant est la sortie de l'élément z(i), tandis que la sortie de chaque mélangeur restant est envoyée à l'entrée de l'élément z(i + 1), avec i = 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22 et 25. Voir la Figure A.1 ci-dessus.

Cela représente un contrôle CRC calculé par le polynôme ci-après:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \qquad \text{(A-1)}$$

Ce sont des octets qui sont reçus à l'entrée du décodeur CRC. Chaque octet est injecté dans le décodeur CRC bit par bit, celui qui a le poids le plus significatif (msb) venant en premier. C'est-à-dire qu'à partir de l'octet 0x01 (dernier octet du préfixe du code de déclenchement), les sept premiers '0' entrent dans le décodeur CRC, suivis par le '1' unique. Avant le traitement par contrôle CRC des données d'une section, la sortie de chaque élément de délai z(i) est mise à sa valeur initiale '1'. Après cette initialisation, chaque octet de la section est envoyé à l'entrée du décodeur CRC, y compris les quatre octets du champ CRC_32. Après injection dans le décodeur du dernier bit du dernier octet de champ CRC_32, c'est-à-dire après injection dans l'élément z(0) du signal mélangé avec la sortie de l'élément z(31), on examine la sortie de tous les éléments z(i) de délai. S'il n'y a pas d'erreur, chacune des sorties des éléments z(i) doit revenir à zéro. Le codeur CRC code le champ CRC_32 de manière que cette valeur zéro soit vérifiée aux sorties.

Annexe B

Commande et contrôle des supports d'enregistrement numérique (DSM-CC)

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

B.0 Introduction

Le protocole de commande DSM-CC est un protocole propre à une application qui est destinée à offrir les principales fonctions de commande et les principales opérations permettant de gérer un flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 sur des supports d'enregistrement numérique. Ce protocole DSM-CC s'insère dans les couches inférieures du modèle OSI, au-dessus de la couche Réseau et au-dessous de la couche Application.

La commande DSM-CC doit être transparente dans le sens suivant:

- elle est indépendante du support DSM utilisé;
- elle est indépendante du fait que le support DSM soit local ou distant;
- elle est indépendante du protocole de couche Réseau avec lequel elle est en interface;
- elle est indépendante des divers systèmes d'exploitation utilisés pour les supports DSM.

B.0.1 Objet

De nombreuses applications du protocole de commande DSM-CC selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 nécessitent un accès à un flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, enregistré sur divers supports d'enregistrement numérique, locaux ou distants. Chaque support DSM a ses propres séquences de commande et chaque utilisateur doit donc connaître différents jeux de séquences de commande DSM afin d'avoir accès aux flux binaires selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 à partir de différents supports DSM. Cela suscite de nombreuses difficultés pour concevoir une interface avec un système d'application conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 ou avec l'ISO/CEI 11172-1. Pour résoudre ces difficultés, la présente annexe suggère un ensemble commun de séquences de commande DSM, indépendant du support DSM utilisé. Les informations de cette annexe sont fournies à titre purement indicatif, tandis que l'ISO/CEI 13818-6 définit l'extension des commandes DSM-CC dans un cadre plus large.

B.0.2 Futures applications

Au-delà des applications immédiatement supportées par les séquences actuelles de commande DSM, les futures applications fondées sur des extensions de ces séquences pourraient offrir les fonctions suivantes.

Vidéo à la demande

Des programmes vidéo sont fournis sur demande d'un client par l'entremise de diverses voies de communication. Le client peut sélectionner un programme vidéo à partir d'une liste de programmes offerts par un serveur de données vidéo. De telles applications pourraient être utilisées par des hôtels, des distributeurs de télévision par câble, des établissements scolaires, des hôpitaux, etc.

Services vidéo interactifs

Dans ces applications, l'utilisateur envoie de fréquents messages de commande pour contrôler la manipulation de données vidéo et audio enregistrées. Ces services peuvent comprendre des jeux vidéo, des excursions vidéo pilotées par l'utilisateur, du téléachat, etc.

Réseaux vidéo

Diverses applications peuvent avoir besoin d'échanger des données audio et vidéo enregistrées, par l'intermédiaire d'un réseau téléinformatique donné. Les utilisateurs pourraient acheminer des informations audiovisuelles, de terminal à terminal, au moyen d'un réseau vidéo. Exemples de ce type d'application: la publication assistée par ordinateur et les applications multimédias.

B.0.3 Avantages

En spécifiant des séquences de commande DSM indépendantes des supports DSM, les utilisateurs peuvent exécuter un décodage selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 sans avoir à comprendre entièrement les détails de fonctionnement du support DSM qu'ils utilisent.

Les séquences de commande DSM sont des codes qui offrent aux utilisateurs la garantie que les flux binaires conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 peuvent être lus et enregistrés avec la même sémantique, quels que soient le

support DSM et l'interface de l'utilisateur. Il s'agit de séquences fondamentales pour la commande du fonctionnement des supports DSM.

B.0.4 Fonctions fondamentales

B.0.4.1 Sélection du flux

Le protocole DSM-CC permet de sélectionner un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, sur lequel il est possible d'effectuer des opérations successives, y compris la création d'un nouveau flux binaire. Les paramètres de cette fonction comprennent:

- un indice renvoyant au flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (le mappage entre cet indice et un nom propre à une application donnée n'est pas dans le domaine d'application du présent protocole DSM-CC);
- un mode de fonctionnement (extraction/enregistrement).

B.0.4.2 Extraction

Le protocole de commande DSM-CC permet:

- de lire un certain flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1;
- d'effectuer cette lecture à partir d'un instant de présentation donné;
- de régler la vitesse de lecture (normale ou rapide);
- de régler la durée de la lecture (jusqu'à un instant de présentation spécifié, jusqu'à la fin du flux binaire en cas d'avance en lecture ou jusqu'à son début en cas de retour en lecture ou jusqu'à l'envoi d'un ordre d'arrêt);
- de choisir le sens de lecture (avance ou retour);
- d'effectuer une pause;
- de reprendre la lecture;
- de changer le point d'accès au flux binaire;
- d'arrêter la lecture.

B.0.4.3 Enregistrement

Le protocole de commande DSM-CC permet:

- de lancer l'enregistrement d'un flux binaire valide pendant une durée spécifiée;
- d'arrêter cet enregistrement.

Le protocole de commande DSM-CC offre un sous-ensemble utile mais limité de caractéristiques fonctionnelles pouvant être requises dans des applications sur supports DSM conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Il est parfaitement prévu d'ajouter d'importantes capacités supplémentaires, au moyen d'extensions ultérieures.

B.1 Eléments généraux

B.1.1 Domaine d'application

Le domaine d'application de la présente annexe consiste à mettre au point une Recommandation | Norme internationale spécifiant un ensemble utile de séquences pour la commande de supports d'enregistrement numérique (DSM, *digital storage media*) de flux binaires conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Ces séquences pourront télécommander un support d'enregistrement numérique de manière globale, indépendamment de la nature de chaque support DSM et pourront s'appliquer à tout flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 enregistré sur un support DSM.

B.1.2 Aperçu général du mode d'application du protocole DSM-CC

La syntaxe et la sémantique actuelles du protocole de commande DSM-CC couvrent les applications de supports DSM pour un utilisateur individuel. Le système de l'utilisateur est capable d'extraire un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 et est aussi (sur option) capable de créer un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. La Figure B.1 présente, sous forme de communication par voies de service, l'interface par laquelle les commandes et acquittements DSM sont envoyés. Si une communication par voies de service n'est pas disponible, on peut aussi acheminer ces données de commande DSM-CC en les insérant dans les flux binaires selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

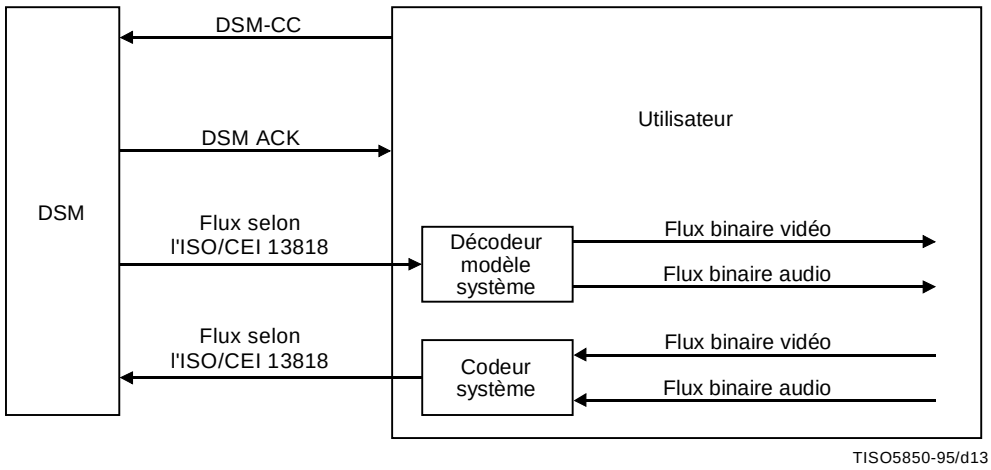


Figure B.1 – Configuration d'une application de commande et DSM-CC

B.1.3 Transmission des commandes et acquittements du protocole DSM-CC

Le protocole DSM-CC est codé sous forme de flux binaire DSM-CC, conformément à la syntaxe et à la sémantique définies dans B.2.2 à B.2.9. Le flux binaire DSM-CC peut être transmis aussi bien sous la forme d'un flux binaire autonome que sous la forme d'un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

Lorsque le flux binaire DSM-CC est transmis en mode autonome, sa relation avec le flux binaire de données systèmes et avec le processus de décodage est illustrée à la Figure B.2. Dans ce cas, le flux DSM-CC n'est pas encastré dans le flux des données systèmes. Ce mode de transmission peut être utilisé dans les applications où le support DSM est directement raccordé au décodeur conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Il peut également être utilisé dans les applications où le flux binaire DSM-CC peut être commandé et transmis au moyen d'autres types de multiplexeurs en couche Réseau.

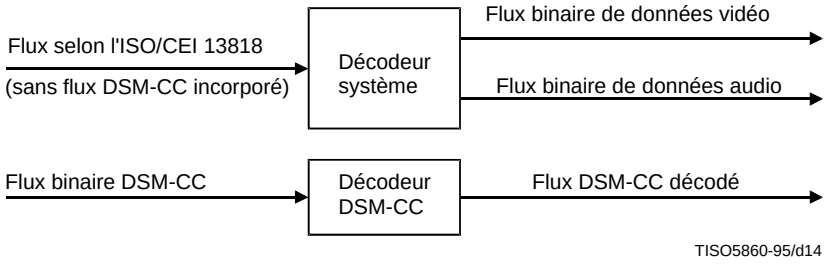


Figure B.2 – Flux binaire DSM-CC décodé sous forme de flux binaire autonome

Pour certaines applications, il est souhaitable de transmettre le protocole DSM-CC dans le cadre d'un flux binaire de données systèmes selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, de manière que certaines caractéristiques des flux binaires de données systèmes conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 puissent être appliquées également au flux binaire de données DSM-CC. Dans ce cas, le flux DSM-CC est encastré par le multiplexeur de données systèmes dans ces dernières.

Le flux binaire DSM-CC est codé comme suit par le codeur de données systèmes. Tout d'abord, le flux binaire DSM-CC est paqueté en un flux élémentaire paqueté (flux PES) conformément à la syntaxe décrite au 2.4.3.6. Ensuite, ce flux PES est multiplexé pour devenir soit un flux de programme (PS) soit un flux de transport (TS), selon les exigences du support de transmission. Les procédures de décodage sont inverses par rapport aux procédures de codage. Elles sont illustrées dans le schéma de la Figure B.3, représentant un décodeur de données systèmes.

Dans la Figure B.3, la sortie du décodeur de données systèmes se compose d'un flux binaire de données vidéo, d'un flux binaire de données audio et d'un flux binaire de protocole DSM-CC. Ce dernier flux est identifié par le champ stream_id, dont la valeur est égale à la séquence binaire '1111 0010' qui est définie par l'identificateur de flux indiqué dans le Tableau 2-22. Une fois le flux DSM-CC identifié, il suit les règles spécifiées par T-STD ou P-STD.

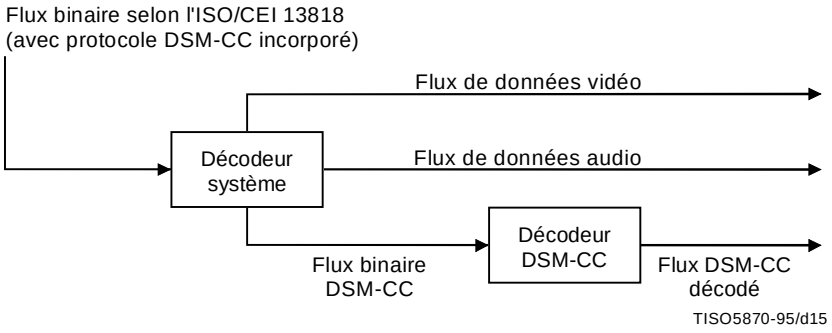


Figure B.3 – Flux de commande DSM-CC décodé sous forme d'élément du flux système

B.2 Eléments techniques

B.2.1 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent:

B.2.1.1 DSM-CC: commandes et contrôles de support d'enregistrement numérique spécifiés par la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 pour la commande de supports d'enregistrement numérique locaux ou distants, contenant un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.2 DSM ACK: message d'acquittement (accusé de réception) envoyé par le récepteur de la commande DSM-CC à l'initiateur de cette commande.

B.2.1.3 flux binaire MPEG: flux de données systèmes de l'ISO/CEI 11172-1, flux de programme de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 ou flux de transport de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.4 serveur de protocole DSM-CC: système, local ou distant, utilisé pour enregistrer et extraire un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.5 point d'accès aléatoire: point d'un flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 qui possède la propriété que, pour au moins un de ses flux élémentaires, l'unité d'accès suivante, 'N', complètement contenue dans ce flux binaire, peut être décodée sans référence à des unités d'accès précédentes et que, pour chacun de ses flux élémentaires, toutes les unités d'accès ayant des horodateurs indiquant un instant de présentation identique ou ultérieur sont complètement contenues dans la suite du flux binaire et peuvent être décodées par un décodeur modèle du système sans accès à des informations situées avant le point d'accès aléatoire. Le flux binaire enregistré sur le support d'enregistrement numérique peut posséder certains points d'accès aléatoire; la sortie du support DSM peut comporter des points d'accès aléatoire supplémentaires, créés par la manipulation des données enregistrées par le support DSM lui-même (par exemple l'enregistrement de matrices de quantification permettant de créer un en-tête de séquence chaque fois que nécessaire). Un point d'accès aléatoire est associé à un horodateur PTS, qui est (réellement ou implicitement) celui de l'unité 'N'.

B.2.1.6 valeur actuelle de l'horodateur PTS opérationnel: valeur de l'horodateur PTS réel ou implicite qui est associé au dernier point d'accès aléatoire précédant la dernière unité d'accès fournie par le protocole DSM à partir du flux binaire actuellement sélectionné, conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Si aucune unité d'accès n'a été fournie par ce flux binaire conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, la valeur actuelle de l'horodateur PTS opérationnel est le premier point d'accès aléatoire dans le flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.7 flux binaire du protocole DSM-CC: séquence d'éléments binaires conforme à la syntaxe du B.2.2.

B.2.2 Spécification de la syntaxe du protocole DSM-CC

- Chaque séquence de commande DSM doit commencer par un code de début tel qu'indiqué dans le Tableau B.1.
- Chaque séquence de commande DSM doit avoir un champ packet_length spécifiant le nombre d'octets contenus dans un paquet DSM-CC.
- Si le flux binaire DSM-CC est transmis en tant que paquet PES tel que défini au 2.4.3.6, le champ packet_start_code_prefix est identique au champ packet_start_code_prefix de paquet PES et le champ packet_length est identique au champ packet_length de paquet PES spécifié au 2.4.3.6. Autrement dit, si le paquet DSM-CC est encapsulé dans un paquet PES, le code de début de paquet PES est le seul code de début au commencement du paquet.

- La séquence de commande ou d'acquittement proprement dite doit suivre le champ packet_length.
- Un flux de données d'acquittement doit être envoyé par le récepteur du flux binaire de commande DSM une fois que l'opération requise a été déclenchée ou effectuée, en fonction de l'ordre spécifique reçu.
- Le protocole DSM est chargé de présenter à tout moment un flux binaire de données systèmes conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Cela peut impliquer la manipulation des bits de mode d'enrichissement définis au 2.4.3.6.

Tableau B.1 – Syntaxe DSM-CC

Syntaxe	Nombre de bits	Mnémonique
DSM_CC() {		
packet_start_code_prefix	**24**	**bslbf**
stream_id	**8**	**uimsbf**
packet_length	**16**	**uimsbf**
command_id	**8**	**uimsbf**
If (command_id == '01') {		
control()		
} else if (command_id == '02') {		
ack()		
}		
}		

B.2.3 Sémantique des champs contenus dans la spécification syntaxique du protocole DSM-CC

packet_start_code_prefix – Champ codé sur 24 bits; avec le champ stream_id qu'il précède, il constitue un champ DSM-CC packet_start_code qui identifie le début d'un flux binaire de paquets DSM-CC. Le champ packet_start_code_prefix est la chaîne binaire '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id – Champ codé sur 8 bits spécifiant l'identification du flux binaire; il prend la valeur '1111 0010' dans le cas du flux binaire DSM-CC. Voir le Tableau 2-23.

packet_length – Champ codé sur 16 bits indiquant le nombre d'octets dans le paquet DSM-CC suivant le dernier octet du champ.

command_id (identificateur de commande) – Codé sur un octet, ce champ contient un entier arithmétique qui indique si un flux binaire est une séquence de commande ou un acquittement. Les valeurs affectées à ce champ sont définies dans le Tableau B.2.

Tableau B.2 – Valeurs assignées à l'identificateur command_id

Valeur	command_id
0x00	Valeur interdite
0x01	Commande
0x02	Acquittement
0x03-0xFF	Valeur réservée

B.2.4 Couche de commande

Contraintes relatives à l'activation des fanions dans la couche de commande DSM-CC

- Un seul, au plus, des fanions de sélection, lecture et enregistrement doit être mis à la valeur '1' pour chaque séquence de commande DSM. Si aucun de ces fanions n'est activé, cette commande est ignorée.
- Un seul, au plus, des fanions de pause (pause_mode), de reprise (resume_mode), d'arrêt d'enregistrement (stop_mode), de lecture (play_flag) et de saut (jump_flag) doit être activé pour chaque commande d'extraction. Si aucun de ces fanions n'est activé, cette commande est ignorée.
- Un seul, au plus, des fanions d'enregistrement (record_flag) et d'arrêt d'enregistrement (stop_mode) doit être choisi pour chaque commande d'enregistrement. Si aucun de ces fanions n'est activé, la commande est ignorée.

Voir Tableau B.3.

Tableau B.3 – Commande DSM-CC

Syntaxe	Nombre de bits	Mnémonique
control() {		
select_flag	**1**	**bslbf**
retrieval_flag	**1**	**bslbf**
storage_flag	**1**	**bslbf**
reserved	**12**	**bslbf**
marker_bit	**1**	**bslbf**
If (select_flag == '1') {		
bitstream_id [31..17]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
bitstream_id [16..2]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
bitstream_id [1..0]	**2**	**bslbf**
select_mode	**5**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (retrieve_flag == '1') {		
jump_flag	**1**	**bslbf**
play_flag	**1**	**bslbf**
pause_mode	**1**	**bslbf**
resume_mode	**1**	**bslbf**
stop_mode	**1**	**bslbf**
reserved	**10**	**bslbf**
marker_bit	**1**	**bslbf**
if (jump_flag == '1') {		
reserved	**7**	**bslbf**
direction_indicator	**1**	**bslbf**
time_code()		
}		
if (play_flag == '1'){		
speed_mode	**1**	**bslbf**
direction_indicator	**1**	**bslbf**
reserved	**6**	**bslbf**
time_code()		
}		
}		
if (storage_flag == '1') {		
reserved	**6**	**bslbf**
record_flag	**1**	**bslbf**
stop_mode	**1**	**bslbf**
if (record_flag == '1') {		
time_code()		
}		
}		
}		

B.2.5 Sémantique des champs contenus dans la couche de commande

marker_bit (bit marqueur) – Fanion codé sur 1 bit qui a toujours la valeur '1' afin d'éviter une émulation de code de déclenchement.

reserved_bits (bits de valeur réservée) – Codé sur 12 bits, ce champ est réservé pour une future extension du protocole de commande DSM-CC selon la présente Recommandation | Norme internationale. Jusqu'alors, il prendra la valeur '0000 0000 0000'.

select_flag (fanion de sélection) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie une opération de sélection d'un flux binaire. S'il est mis à '0', il ne peut y avoir d'opération de sélection.

retrieval_flag (fanion d'extraction) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie qu'une action spécifique d'extraction (pour lecture) va se produire. L'opération commence à partir de la valeur actuelle de l'horodateur PTS opérationnel.

storage_flag (fanion d'enregistrement) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie qu'une opération d'enregistrement va être exécutée.

bitstream_ID (identificateur de flux binaire) – Codé sur 32 bits, ce champ est subdivisé en trois parties définies dans la syntaxe de manière à former un entier non signé qui spécifie quel flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 il y a lieu de sélectionner. Il appartient au serveur de commande DSM de mapper de manière biunivoque les noms des flux binaires selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, enregistrés dans sa mémoire de commandes DSM, avec une série de numéros pouvant être représentés par l'identificateur de flux binaire bitstream_ID.

select_mode (mode à sélectionner) – Codé sur 5 bits, ce champ contient un entier non signé qui spécifie le mode d'exploitation voulu du flux binaire. Le Tableau B.4 ci-dessous spécifie les modes définis.

Tableau B.4 – Valeurs assignées au mode à sélectionner

Code	Mode
0x00	Valeur interdite
0x01	Enregistrement
0x02	Extraction
0x03-0x1F	Valeur réservée

jump_flag (fanion de saut) – Ce fanion codé sur 1 bit spécifie que le pointeur de lecture doit sauter à une nouvelle unité d'accès. Le nouvel horodateur PTS est spécifié par un code temporel différentiel, se rapportant à la valeur actuelle de l'horodateur PTS opérationnel. Cette activité n'est valide que lorsque le flux binaire actuel, conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, est en mode d'"arrêt".

play_flag (fanion de lecture) – Ce fanion codé sur 1 bit spécifie la lecture d'un flux binaire pendant un certain temps. Lorsque ce fanion est activé, la vitesse, le sens et la durée de lecture sont des paramètres additionnels contenus dans le flux binaire. La lecture commence à partir de la valeur actuelle de l'horodateur PTS opérationnel.

pause_mode (mode de pause) – Ce fanion codé sur 1 bit spécifie une pause de lecture et conserve le pointeur de lecture sur la valeur actuelle de l'horodateur PTS opérationnel.

resume_mode (mode de reprise) – Ce fanion codé sur 1 bit spécifie que l'action de lecture doit être reprise depuis la valeur actuelle de l'horodateur PTS opérationnel. La reprise n'a de sens que si le flux binaire actuel est dans l'état de "pause" et est mis en mode d'avance en lecture à vitesse normale.

stop_mode (mode d'arrêt) – Ce fanion codé sur 1 bit spécifie l'arrêt d'une transmission de flux binaire.

direction_indicator (indicateur du sens de lecture) – Ce fanion codé sur 1 bit indique le sens de la lecture. S'il est mis à '1', il indique une avance en lecture. Sinon, il indique un retour en lecture.

speed_mode (mode de vitesse) – Ce fanion codé sur 1 bit spécifie la grandeur de vitesse. S'il est mis à '1', il spécifie que la vitesse est en lecture normale. S'il est mis à '0', il spécifie que la vitesse est en lecture rapide (c'est-à-dire en avance rapide ou en retour rapide).

record_flag (fanion d'enregistrement) – Ce fanion codé sur 1 bit spécifie qu'un utilisateur demande l'enregistrement du flux binaire sur un support DSM pendant une durée spécifiée ou jusqu'à réception d'une commande d'arrêt, selon ce qui arrive en premier.

B.2.6 Couche d'acquittement

Contraintes sur l'activation des fanions dans une commande DSM-CC

Pour chaque flux binaire d'acquittement de commande DSM, un seul des fanions de sélection, d'extraction, d'enregistrement et d'erreur doit être sélectionné (mis à '1') (voir Tableau B.5).

Tableau B.5 – Acquittement de commande DSM-CC

Syntaxe	Nombre de bits	Mnémonique
ack() {		
select_ack	**1**	**bslbf**
retrieval_ack	**1**	**bslbf**
storage_ack	**1**	**bslbf**
error_ack	**1**	**bslbf**
reserved	**10**	**bslbf**
marker_bit	**1**	**bslbf**
cmd_status	**1**	**bslbf**
If (cmd_status == '1' &&		
(retrieval_ack == '1' || storage_ack == '1')) {		
time_code()		
}		
}		

B.2.7 Sémantique des champs contenus dans la couche d'acquittement

select_ack (acquittement de sélection) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie que la commande ack() vise à accuser réception d'une commande de sélection.

retrieval_ack (acquittement d'extraction) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie que la commande ack() vise à accuser réception d'une commande d'extraction.

storage_ack (acquittement d'enregistrement) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie que la commande ack() vise à accuser réception d'une commande d'enregistrement.

error_ack (acquittement d'erreur) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie qu'il y a une erreur de support DSM. Les erreurs actuellement définies sont une EOF (fin de fichier lors d'une avance en lecture ou début de fichier lors d'un retour en lecture) dans le cas d'une extraction de flux, ainsi qu'un message de disque plein lors d'un enregistrement de flux. Si ce fanion est mis à '1', le champ cmd_status n'est pas défini. Dans un cas comme dans l'autre, le flux binaire actuel reste sélectionné.

cmd_status (statut de commande) – Ce fanion codé sur 1 bit, s'il est mis à '1', spécifie que la commande est acceptée; s'il est mis à 0, la commande est rejetée. Selon le type de commande reçu, les sémantèmes suivants sont applicables:

- si le champ select_ack est activé et que le fanion cmd_status soit à '1', celui-ci spécifie que le flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est sélectionné et que le serveur est prêt à fournir le mode de fonctionnement choisi. La valeur actuelle de l'horodateur PTS opérationnel est réglée sur le premier point d'accès aléatoire du flux binaire nouvellement choisi, conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Si le champ cmd_status est mis à '0', l'opération a échoué et aucun flux binaire n'est sélectionné;
- si le champ retrieval_ack est activé et que le fanion cmd_status soit à '1', celui-ci spécifie que l'opération d'extraction est lancée pour toutes les commandes d'extraction. La position de la valeur actuelle de l'horodateur PTS opérationnel est signalée par le champ time_code suivant;
- pour la commande d'activation du fanion de lecture play_flag avec infinite_time_flag != '1', un deuxième acquittement sera envoyé pour accuser réception du fait que l'opération de lecture s'est terminée à l'expiration de la durée définie par la commande play_flag;
- si le champ cmd_status est mis à '0' dans un acquittement d'extraction, l'opération a échoué. Les raisons possibles de cet échec sont par exemple un identificateur de flux non valide, un saut au-delà de la fin d'un fichier, ou une fonction non gérée, comme le retour en lecture en vitesse normale;
- si le champ storage_ack est activé, le champ cmd_status spécifie que l'opération d'enregistrement est en cours de lancement pour la commande record_flag ou a été réalisée pour la commande stop_mode. Le pointeur PTS de la dernière unité d'accès complète qui a été enregistrée est acheminé par le champ time_code suivant;
- si l'opération d'enregistrement se termine à l'expiration d'une durée définie par la commande storage_flag, un autre acquittement doit être envoyé et la valeur actuelle de l'horodateur PTS opérationnel doit être signalée;

- si le champ cmd_status est mis à '0' dans un acquittement d'enregistrement, l'opération a échoué. Les raisons possibles de cet échec sont par exemple un identificateur de flux non valide ou l'incapacité du support DSM de mémoriser les données.

B.2.8 Code temporel

Contraintes sur le code temporel

- Une opération d'avance en lecture, de durée spécifiée par un champ time_code, se termine dès qu'on observe que l'horodateur PTS réel ou implicite d'une unité d'accès est tel que cette durée soit inférieure à la différence, modulo 2^{33}, entre la valeur de cet horodateur PTS et la valeur actuelle de l'horodateur PTS opérationnel au début de l'opération.
- Une opération de retour en lecture, de durée spécifiée par un champ time_code, se termine dès qu'on observe que l'horodateur PTS réel ou implicite d'une unité d'accès est tel que cette durée soit inférieure à la différence, modulo 2^{33}, entre la valeur actuelle de l'horodateur PTS opérationnel au début de l'opération et la valeur de cet horodateur PTS.
- Pour toutes les commandes contenues dans la couche de commande, le code temporel est spécifié sous la forme d'une durée rapportée à la valeur actuelle de l'horodateur opérationnel.
- Pour toutes les commandes contenues dans la couche d'acquittement, le code temporel est spécifié par la valeur actuelle de l'horodateur opérationnel.

Voir Tableau B.6.

Tableau B.6 – Code temporel

Syntaxe	Nombre de bits	Mnémonique
time_code() {		
reserved	**7**	**bslbf**
infinite_time_flag	**1**	**bslbf**
if (infinite_time_flag == '0') {		
reserved	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
}		

B.2.9 Sémantique des champs contenus dans la structure de code temporel

infinite_time_flag (fanion de temps infini) – S'il est mis à '1', ce fanion codé sur 1 bit spécifie que la période de temps est infinie. Il est mis à '1' dans des applications telles qu'il n'est pas possible de définir à l'avance une période de temps pour une opération spécifique.

PTS [32..0] – Ce champ contient l'horodateur de présentation de l'unité d'accès contenue dans le flux binaire. Selon la fonction choisie, il peut s'agir d'une valeur absolue ou d'un délai relatif, exprimé en périodes de l'horloge système à 90 kHz.

Annexe C

Informations spécifiques du programme

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

C.0 Explications relatives aux informations spécifiques du programme (PSI) contenues dans un flux de transport

Le paragraphe 2.4.4 contient la syntaxe normative, la sémantique et le texte concernant les informations spécifiques du programme. Dans tous les cas, la conformité avec les contraintes du 2.4.4 est requise. La présente annexe donne des explications sur la manière d'utiliser les fonctions d'informations PSI et présente des exemples sur la façon dont ces fonctions peuvent être utilisées en pratique.

C.1 Introduction

La présente Recommandation | Norme internationale donne une méthode permettant de décrire le contenu de paquets de flux de transport afin de démultiplexer et de présenter des programmes. Le codage spécifié tient compte de cette fonction au moyen des informations spécifiques du programme (PSI). La présente annexe en développe l'utilisation.

On peut considérer les informations PSI comme appartenant à six tables:

1) la table d'association de programmes (PAT, *program association table*);
2) la table de mappage de programme (PMT, *program map table*) pour le flux de transport;
3) la table d'informations sur le réseau (NIT, *network information table*);
4) la table d'accès conditionnel (CAT, *conditional access table*);
5) la table de description de flux de transport (TSDT, *transport stream description table*);
6) la table d'informations de commande IPMP (ICIT, *IPMP control information table*).

Le contenu des tables PAT, PMT, CAT et TSDT est spécifié dans la présente Recommandation | Norme internationale. Le contenu de la table ICIT est défini dans l'ISO/CEI 13818-11 (MPEG-2 IPMP).

La table NIT est privée mais la valeur d'identification des paquets de transport qui l'acheminent est spécifiée dans la table PAT. Les tables NIT et ICIT doivent toutes deux être conformes à la structure définie dans la présente Recommandation | Norme internationale.

C.2 Mécanisme fonctionnel

Les tables énumérées ci-dessus sont théoriques en ce sens qu'elles n'ont jamais besoin d'être régénérées sous une forme spécifiée à l'intérieur d'un décodeur. Bien qu'on puisse considérer ces structures comme de simples tableaux, on peut les subdiviser avant de les expédier dans des paquets de transport. La syntaxe permet d'effectuer cette opération car elle autorise la subdivision des tables en sections et offre une méthode normative d'insertion de ces tables dans des capacités utiles de paquets de flux de transport. On indique également comment acheminer des données privées sous un format analogue. Cela est avantageux car on peut utiliser dans le décodeur le même traitement de base pour les informations PSI et pour les données privées, ce qui contribue à diminuer les coûts. On trouvera en Annexe D des indications relatives à l'optimisation du placement des informations PSI dans le flux de transport.

Chaque section est identifiée de façon univoque par la combinaison des éléments suivants:

i) **champ table_id**

Codé sur un octet, le champ table_id identifie la table à laquelle la section appartient:

- Les sections dont le champ table_id a la valeur 0x00 appartiennent à une table d'association de programmes.
- Les sections dont le champ table_id a la valeur 0x01 appartiennent à une table d'accès conditionnel.
- Les sections dont le champ table_id a la valeur 0x02 appartiennent à une table de mappage de programme pour un flux de transport.
- Les sections dont le champ table_id a la valeur 0x03 appartiennent à la section TS_description_section.
- Les sections dont le champ table_id a la valeur 0x04 appartiennent à la section ISO_IEC_14496_scene_description_section.

- Les sections dont le champ table_id a la valeur 0x05 appartiennent à la section ISO_IEC_14496_object_descriptor_section.
- Les sections dont le champ table_id a la valeur 0x06 appartiennent à la section metadata_section.
- Les sections dont le champ table_id a la valeur 0x07 appartiennent à la section IPMP_Control_Information_section.

L'utilisateur peut attribuer à ce champ d'autres valeurs pour des fins privées.

Il est possible de mettre en place des filtres examinant le champ table_id pour déterminer si une nouvelle section appartient ou non à une table recherchée.

ii) **champ table_id_extension**

Codé sur 16 éléments binaires, ce champ existe en cas de section en version longue. Dans la table d'association de programmes, il sert à repérer l'identificateur du flux de transport, qui se présente sous la forme d'une étiquette définie par l'utilisateur, permettant de distinguer un flux de transport d'un autre flux à l'intérieur d'un réseau ou parmi plusieurs réseaux. Dans la table d'accès conditionnel, ce champ n'a pour le moment aucune signification particulière; il est donc marqué "réservé", c'est-à-dire qu'il doit être codé avec la valeur 0xFFFF. L'UIT-T | ISO/CEI pourra toutefois définir une signification dans une future version révisée de la présente Recommandation | Norme internationale. Dans une table de mappage de programme pour un flux de transport, ce champ contiendra le numéro de programme et identifiera donc le programme auquel les données de la section se rapportent. Le champ table_id_extension peut également servir de point de filtrage dans certains cas.

iii) **champ section_number**

Ce champ permet au décodeur de réordonnancer les sections d'une table particulière afin de les remettre dans leur ordre d'origine. La présente Recommandation | Norme internationale ne prescrit pas impérativement que les sections soient transmises dans leur ordre numérique; mais cela est recommandé, sauf si l'on souhaite transmettre certaines sections de la table plus fréquemment que d'autres, par exemple pour des raisons d'accès aléatoire.

iv) **champ version_number**

Lorsque les caractéristiques du flux de transport, décrites dans les informations PSI, se modifient (par exemple si des programmes supplémentaires sont ajoutés, si un programme donné se compose de flux élémentaires différents), les données des nouvelles informations PSI doivent être envoyées avec les informations actualisées car la version transmise le plus récemment des sections marquées comme étant "actuelles" doit toujours être valide. Il est nécessaire que les décodeurs soient en mesure de déterminer si la section reçue le plus récemment est identique à la section qu'ils ont déjà traitée ou mémorisée (auquel cas cette dernière section peut être ignorée) ou si elle est différente et peut donc annoncer un changement de configuration. Ce repérage s'effectue par l'envoi d'une section comportant les mêmes champs d'identification de table, d'extension d'identificateur de table et de numéro de section que la précédente section contenant les données recherchées, mais avec la valeur suivante de numéro de version.

v) **champ current_next_indicator**

Il importe de connaître le point du flux binaire à partir duquel les informations PSI sont valides. Chaque section peut donc être numérotée comme étant valide "maintenant" (actuelle) ou valide dans le futur immédiat (prochaine table applicable). Cela permet de transmettre une future configuration avant le changement réel, ce qui donne au décodeur la possibilité de s'y préparer. Il n'est toutefois pas obligatoire de transmettre d'avance la prochaine version d'une section; mais, si cette opération est effectuée, il doit s'agir de la prochaine version correcte de cette section.

C.3 Mappage de sections dans des paquets du flux de transport

Les sections sont insérées directement dans les paquets du flux de transport, c'est-à-dire sans insertion préalable dans des paquets de flux PES. Les sections peuvent ne pas commencer au tout début des paquets de transport car le début de la première section contenue dans la capacité utile d'un paquet de transport sera signalé par le champ pointer_field, dont la présence est indiquée par le fait que le champ payload_unit_start_indicator est mis à la valeur '1' dans les paquets d'informations PSI. (Dans les paquets non porteurs d'informations PSI, cet indicateur signale qu'un paquet PES commence dans le paquet de transport.) Le champ pointer_field indique le début de la première section contenue dans le paquet du flux de transport. Il n'y a jamais plus d'un seul champ pointer_field dans un paquet de transport car le début d'une autre section pourra être identifié par comptage de la longueur de la première section et de toutes sections ultérieures, la syntaxe ne permettant aucune discontinuité entre les sections contenues dans un paquet du flux de transport.

Il importe de noter que, dans les paquets du flux de transport ayant la même valeur d'identificateur PID, chaque section doit être terminée avant que la suivante puisse être autorisée à commencer; sinon, il ne serait pas possible de déterminer

à quel en-tête de section les données appartiennent. Si une section se termine avant la fin d'un paquet de transport mais qu'il ne soit pas opportun d'ouvrir une autre section, un procédé de bourrage permettra de remplir l'espace intermédiaire. Chaque octet restant dans le paquet sera bourré de valeur 0xFF. Par conséquent, la valeur 0xFF est interdite pour le champ table_id, afin de ne pas le confondre avec un octet de bourrage. Une fois qu'un octet 0xFF est apparu à la fin d'une section, le reste du paquet de transport doit être bourré d'octets 0xFF, ce qui permettra à un décodeur d'ignorer ce reste de paquet de transport. On peut également effectuer le bourrage au moyen du mécanisme normal du champ adaptation_field.

C.4 Fréquences de répétition et accès aléatoire

Dans les systèmes mettant en jeu l'accès aléatoire, il est recommandé de retransmettre plusieurs fois les sections d'informations PSI, même si aucun changement de configuration n'intervient; car, dans le cas général, un décodeur aura besoin des données d'informations PSI pour identifier le contenu du flux de transport et pour pouvoir commencer à le décoder. La présente Recommandation | Norme internationale ne prescrit aucune fréquence de répétition ou d'apparition des sections d'informations PSI. Il est cependant évident que le fait de répéter fréquemment des sections est utile aux applications à accès aléatoire tout en provoquant une montée en débit des données PSI. Si des mappages avec des programmes sont statiques ou quasi statiques, on peut les mémoriser dans le décodeur pour accéder plus rapidement aux données et ne pas attendre qu'elles soient retransmises. Le compromis entre l'immobilisation de mémoire nécessaire et l'influence sur le temps d'acquisition du canal peut être calculé par le constructeur du décodeur.

C.5 En quoi consiste un programme?

Le concept de programme possède une définition précise dans le cadre de la présente Recommandation | Norme internationale [programme (système) (voir 2.1.60)]. Pour un flux de transport, la base de temps est définie par la référence temporelle de programme. Cela crée en fait une voie virtuelle à l'intérieur du flux de transport.

On notera que cette définition n'est pas celle qui est couramment utilisée en télédiffusion, où un "programme" est un ensemble de flux élémentaires possédant non seulement une base de temps commune mais aussi un même instant de début et de fin. Une série de "programmes de télédiffuseurs" (désignés dans la présente annexe par le terme "événement") peut être transmise en séquence dans un flux de transport utilisant le même numéro de programme pour créer un canal de télévision "conventionnel pour la télédiffusion" (parfois appelé "service").

Les descriptions d'événements peuvent être acheminées dans des sections privées.

Un programme est désigné par un champ de numéro de programme qui n'a de signification qu'à l'intérieur d'un flux de transport donné. Ce champ contient un entier non signé de 16 éléments binaires, qui permet donc d'insérer 65535 programmes différents à l'intérieur d'un même flux de transport (la valeur 0 du champ program_number est réservée pour l'identification de la table NIT). Lorsque plusieurs flux de transport sont mis à la disposition du décodeur (par exemple dans un réseau de câbles), celui-ci doit, pour pouvoir démultiplexer correctement un programme, recevoir d'une part l'identificateur du flux de transport (afin de trouver le multiplex correct) et le numéro de programme du service (afin de trouver le programme correct à l'intérieur du multiplex).

Le mappage dans le flux de transport peut être effectué au moyen de la table NIT facultative. On notera que cette table d'informations sur le réseau peut être stockée dans une mémoire rémanente du décodeur afin de diminuer le temps d'acquisition du canal. Dans ce cas, il suffit de transmettre cette table assez souvent pour permettre des opérations de reconfiguration par réinitialisation en temps utile du décodeur. Le contenu de la table NIT est d'ordre privé mais doit suivre au moins la structure de base des sections.

C.6 Affectation du numéro de programme

Il n'est pas toujours pratique de regrouper tous les flux élémentaires ayant en partage la même référence temporelle, pour constituer un même programme. On peut concevoir un flux de transport multiservice ne possédant qu'un seul ensemble de références PCR, commun à tous les services. En général, un diffuseur préférera cependant subdiviser logiquement le flux de transport en plusieurs programmes, fondés sur une même identification (d'emplacement) de référence PCR (champ PCR_PID). Cette méthode de répartition des flux élémentaires en programmes pseudo-indépendants peut avoir plusieurs applications, dont on trouvera ci-dessous deux exemples:

i) *transmissions plurilingues vers des marchés distincts*

Un même flux de données vidéo peut être assorti de plusieurs flux de données audio, en plusieurs langues. Il est recommandé de donner un exemple du descripteur de langue selon l'ISO 639, associé à chaque flux audio, afin de permettre la sélection correcte du programme et de ses données audio. Il est rationnel d'avoir plusieurs définitions de programme avec différents numéros de programme, où tous les programmes se rapportent au même flux vidéo et au même identificateur de référence temporelle PCR_PID mais avec des identificateurs de paquets audio différents. Il est tout aussi rationnel

et possible de donner une liste de tous les flux audio associés à un même programme de flux vidéo, du moment que cette liste ne dépasse pas la limite de capacité de section (1024 octets).

ii) *définitions de programme très longues*

La longueur d'une section est limitée à 1024 octets (y compris son en-tête et sa séquence de contrôle CRC_32). C'est-à-dire qu'aucune définition de programme ne peut dépasser cette longueur. Dans la grande majorité des cas, cette dimension est adéquate, même si chaque flux élémentaire possède plusieurs descripteurs. On peut toutefois envisager des cas de dépassement de cette limite, pour des systèmes à débits très élevés. Il sera alors possible, en général, de trouver des méthodes de répartition des références de flux, de manière à ne plus avoir à les énumérer sur la même liste. Certains flux élémentaires pourront être désignés au titre de plusieurs programmes et d'autres au titre d'un seul mais pas les deux à la fois.

C.7 Utilisation des informations PSI dans un système type

Un système de communication, surtout dans les applications de télédiffusion, peut se composer de nombreux flux de transport individuels. Chacune des quatre structures de données PSI peut apparaître dans chacun des flux de transport d'un même système. Chaque flux de transport doit toujours contenir une version complète de la table d'association de programmes (PAT) énumérant tous les programmes contenus dans ce flux, ainsi qu'une version complète de la table de mappage de programme pour le flux de transport (PMT), contenant les définitions complètes des programmes contenus dans le flux de transport. Si certains flux sont embrouillés, il faut également qu'une table d'accès conditionnel (CAT) soit présente afin d'énumérer les flux de messages de gestion d'habilitation (EMM, *entitlement management message*). La présence d'une table d'informations sur le réseau (NIT) est entièrement facultative.

Les tables d'informations PSI sont mappées dans les paquets du flux de transport au moyen de la structure de section décrite ci-dessus. Chaque section possède, dans son en-tête, un champ table_id qui permet de mélanger, dans des paquets de transport ayant la même valeur de PID ou même dans un seul paquet de transport, des sections issues de tables PSI et des données contenues dans des sections privées. On notera cependant qu'à l'intérieur des paquets d'un même PID, une section doit toujours être transmise complètement avant que la section suivante puisse être commencée. Cela n'est toutefois possible que pour les paquets étiquetés comme contenant une section de table PMT ou pour les paquets de table NIT, car des sections privées ne peuvent pas être mappées dans des paquets de table PAT ou CAT.

Il est requis que toutes les sections de table PAT soient mappées dans des paquets du flux de transport avec PID =0x0000 et que toutes les sections de table CAT soient mappées dans des paquets avec PID = 0x0001. Les sections de table PMT peuvent être mappées dans des paquets ayant une valeur choisie par l'utilisateur, indiquée dans la table PAT comme étant le champ PMT_PID pour chaque programme. De même, l'identificateur PID des paquets de transport acheminant des données de table NIT est choisi par l'utilisateur mais doit être indiqué par l'entrée "program_number == 0x00" dans la table PAT, si la table NIT existe.

Le contenu des flux paramétriques d'accès conditionnel est entièrement privé mais des messages EMM et ECM peuvent également être envoyés dans des paquets de transport afin de respecter les prescriptions de la présente Recommandation | Norme internationale.

Les tables de données privées peuvent être envoyées dans des flux conformes à la syntaxe de section privée. Ces tables pourront par exemple être utilisées dans un environnement de télédiffusion pour décrire un service, un événement imminent, un horaire de diffusion et des informations associées.

C.8 Relations entre structures d'informations PSI

La Figure C.1 donne un exemple de relations entre les quatre structures d'informations PSI et le flux de transport. D'autres exemples sont possibles mais la figure montre les liaisons de base.

Les paragraphes suivants décriront chaque table d'informations PSI.

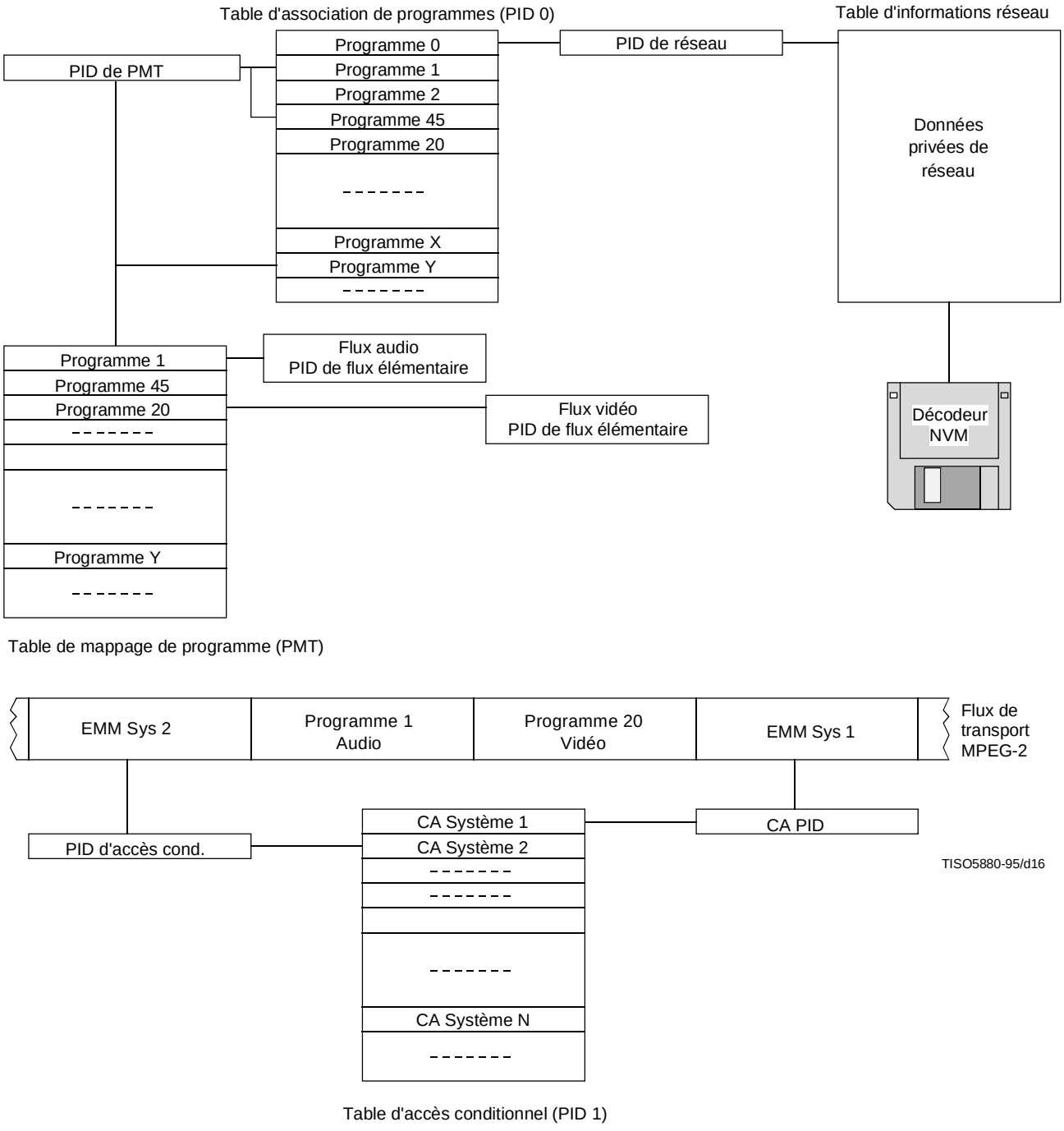


Figure C.1 – Relations bijectives entre programme et réseau

C.8.1 Table d'association de programmes (PAT)

Chaque flux de transport doit toujours contenir une table d'association de programmes complète et valide. La table PAT indique la correspondance entre un numéro de programme et l'identificateur PID des paquets de transport qui acheminent la définition de ce programme (l'identificateur PMT_PID). La table PAT peut être subdivisée en (au plus) 255 sections avant d'être mappée dans les paquets du flux de transport. Chaque section contient une partie de la table PAT complète. Une telle subdivision peut être souhaitable afin de minimiser la perte de données en environnement exposé aux erreurs. C'est-à-dire que l'on peut localiser des pertes de paquet ou des erreurs binaires dans des sections plus petites de la table PAT, ce qui permet de continuer à recevoir et à décoder correctement d'autres sections. Si toutes les informations d'une table PAT étaient mises dans une même section, une erreur provoquant un changement de bit dans l'identificateur de table (par exemple) provoquerait la perte de toute la table PAT. Cela reste cependant autorisé du moment que la section ne dépasse pas la longueur maximale de 1024 octets.

Le numéro de programme 0 (zéro) est réservé pour spécifier le PID de réseau. Il pointe sur les paquets du flux de transport qui contiennent la table d'informations sur le réseau (NIT).

La table d'association de programmes est toujours transmise sans embrouillage.

C.8.2 Table de mappage de programme (PMT)

La table de mappage de programme donne le mappage entre un numéro de programme et les flux élémentaires qui le composent. Cette table est présente dans les paquets du flux de transport ayant une ou plusieurs valeur(s) de PID, sélectionnées à titre privé. Ces paquets de transport peuvent contenir d'autres structures privées, définies par l'octet table_id. Il est possible d'avoir des sections de table PMT pour flux de transport se rapportant à différents programmes contenus dans des paquets de transport ayant une même valeur de PID.

La présente Recommandation | Norme internationale prescrit une identification minimale des programmes: le numéro de programme, l'identificateur de référence PCR, les types de flux élémentaires et les identificateurs de flux élémentaire. D'autres informations peuvent être acheminées, concernant des programmes ou des flux élémentaires, au moyen de la structure syntaxique descriptor(). Voir C.8.6.

Des données privées peuvent également être acheminées dans des paquets de flux de transport étiquetés comme contenant des sections de table PMT pour flux de transport. Pour cela, on utilisera le champ private_section(), dans lequel l'application décide si le numéro de version et l'indicateur de prochaine table applicable ont des valeurs de champ concernant une même section ou concernant de nombreuses sections faisant partie d'une table privée plus vaste.

NOTE 1 – Les paquets de flux de transport contenant la table PMT sont transmis sans embrouillage.

NOTE 2 – Il est possible de transmettre des informations relatives à des événements au moyen de descripteurs privés contenus dans les champs de type TS_program_map_section().

C.8.3 Table d'accès conditionnel (CAT)

La table d'accès conditionnel (CAT) donne l'association entre un ou plusieurs systèmes d'accès conditionnel, leurs flux de messages EMM et tous paramètres spéciaux qui leur sont associés.

NOTE – Le contenu (privé) des paquets de flux de transport contenant des messages EMM et des paramètres d'accès conditionnel (le cas échéant), sera généralement chiffré (embrouillé).

C.8.4 Table d'informations sur le réseau (NIT)

Le contenu de la table NIT est privé et n'est pas spécifié par la présente Recommandation | Norme internationale. Il contiendra, en général, des mappages entre des services sélectionnés par l'utilisateur et des identificateurs de flux de transport, des fréquences porteuses, des numéros de répéteurs à bord de satellites, des caractéristiques de modulation, etc.

C.8.5 Sections privées

Les champs de section privée [private_section()] peuvent se présenter sous deux formes de base: la version courte (ne comportant que les champs jusqu'à section_length inclus) ou la version longue (comportant tous les champs jusqu'à last_section_number inclus et contenant le champ CRC_32 après les octets de données privées).

Des sections privées peuvent apparaître dans des identificateurs PID étiquetés sous la forme PMT_PID ou dans des paquets de flux de transport ayant d'autres valeurs de PID, contenant exclusivement des sections privées, y compris l'identificateur PID attribué à la table NIT. Si les paquets de transport du PID acheminant les sections privées sont identifiés comme ayant un PID transportant des champs de type private_section (avec valeur 0x05 affectée au champ stream_type), seules des sections privées peuvent apparaître dans des paquets de transport ayant cette valeur de PID; ces sections pouvant être de type court ou de type long.

C.8.6 Descripteurs

La présente Recommandation | Norme internationale définit plusieurs descripteurs normatifs. On peut en définir beaucoup d'autres. Tous les descripteurs ont un format commun: {étiquette, longueur, données}. Tous les descripteurs définis en privé doivent suivre ce format. La portion données de ces descripteurs privés est définie à titre privé.

S'il se trouve dans une section de table PMT pour flux de transport, un seul descripteur [le champ CA_descriptor()] est utilisé pour indiquer l'emplacement (valeur du PID de paquets de transport) des données de message ECM associées à des flux élémentaires. S'il se trouve dans une section de table CAT, il se rapporte à des messages EMM.

Afin d'augmenter le nombre de descripteurs de données privées disponibles, on pourra faire appel au procédé suivant. Une étiquette de descripteur de données privées peut être définie à titre privé afin de construire un descripteur composite. Cela implique la définition d'un descripteur secondaire qui deviendra le premier champ des octets de données privées du descripteur de données privées. La structure décrite est indiquée dans les Tableaux C.1 et C.2.

Tableau C.1 – Descripteur composite (élément composite_descriptor)

Syntaxe	Nombre de bits	Mnémonique
Composite_descriptor(){		
descriptor_tag(privately defined)	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++){		
sub_descriptor()		
}		
}		

Tableau C.2 – Descripteur secondaire

Syntaxe	Nombre de bits	Mnémonique
sub_descriptor() {		
sub_descriptor_tag	**8**	**uimsbf**
sub_descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**uimsbf**
}		
}		

C.9 Affectation spectrale et temps d'acquisition du signal

Toute implémentation d'un flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 doit formuler des demandes d'affectation spectrale raisonnables en termes d'informations PSI et, dans les applications mettant en jeu l'accès aléatoire, doit normalement viser à une acquisition rapide du signal. Le présent paragraphe analysera ce problème et donnera quelques exemples pour la télédiffusion.

La nature discontinue du flux de paquets de transport permet de disperser les informations PSI dans les données multiplexées avec une granularité fine. Cela apporte une notable flexibilité pour construire et transmettre ces informations.

Le temps d'acquisition du signal dépend, dans un codeur réel, de nombreux facteurs, comme les suivants: la pente de montée en surtension d'accord pour le multiplexage par répartition en fréquence (FDM), le temps de démultiplexage, les en-têtes de séquence, la fréquence d'images de type I (à codage non prédictif) et les processus d'extraction-traitement de clé d'embrouillage.

Le présent paragraphe traite des incidences, en termes de débit comme de temps d'acquisition du signal, de la syntaxe des informations PSI contenue dans les 2.4.4.4 et 2.4.4.9. On part du principe que l'on n'a pas besoin de recevoir dynamiquement, à chaque changement de programme, la table d'accès conditionnel. Cette hypothèse vaut également pour les flux de messages EMM privés. Ces flux ne contiennent en effet pas les composantes, à variation rapide, des messages ECM utilisés pour l'embrouillage (chiffrement) des flux élémentaires.

On a également négligé, dans l'analyse ci-dessous, le temps d'acquisition et de traitement des messages ECM.

Les Tableaux C.3 et C.4 indiquent les valeurs d'affectation spectrale pour une série de configurations du flux de transport. Un axe de ces tableaux est le nombre de programmes contenus dans un même flux de transport. L'autre axe est la fréquence de transmission des informations PSI dans ce flux de transport.

Tableau C.3 – Affectation spectrale de la table d'association de programmes
Nombre de programmes par flux de transport

		1	5	10	32	128
Fréquence des informations de table PAT (Hz)	1	1504	1504	1504	1504	4512
	10	15040	15040	15040	15040	45120
	25	37600	37600	37600	37600	112800
	50	75200	75200	75200	75200	225600
	100	150400	150400	150400	150400	451200
NOTE – Etant donné que 46 champs **program_association_sections** s'intègrent dans un même paquet de transport, les nombres indiqués dans le tableau ne changent pas avant la dernière colonne.						

Tableau C.4 – Affectation spectrale de la table de mappage du flux de programme
Nombre de programmes par flux de transport

		1	5	10	32	128
Fréquence des informations de PMT (Hz)	1	1504	1504	3008	7520	28576
	10	15040	15040	30080	75200	285760
	25	37600	37600	75200	188000	714400
	50	75200	75200	150400	376000	1428800
	100	150400	150400	300800	601600	2857600

Cette fréquence sera un facteur déterminant du temps d'acquisition du signal, en fonction des structures des informations PSI.

Les deux tableaux d'affectation spectrale sont fondés sur l'hypothèse qu'on ne fournit que le minimum d'informations relatives au mappage des programmes. C'est-à-dire que les valeurs d'identification PID et les types de flux sont indiqués sans descripteurs additionnels. Tous les programmes sont, dans cet exemple, composés de deux flux élémentaires. Les associations de programmes ont une longueur de 2 octets, alors que l'information minimale sur le mappage de programme a une longueur de 26 octets. Une certaine servitude binaire s'ajoutera en raison des numéros de version, des indicateurs de longueur de section, etc. Cette servitude sera de l'ordre de 1% à 3% de l'affectation spectrale totale des informations PSI à des sections de longueur moyenne à maximale (de quelques centaines d'octets à 1024 octets). On n'en tiendra donc pas compte ici.

Les hypothèses précédentes permettent d'insérer quarante-six (46) associations de programmes dans un même paquet de transport de table PAT (si aucun champ d'adaptation n'est présent). De même, sept (7) champs TS_program_map_section s'intégreront dans un seul paquet de transport. On pourra noter que, pour faciliter les opérations de suppression/adjonction, il est possible de ne transmettre qu'un seul (1) champ TS_program_map_section par identificateur PMT_PID. Cela peut cependant se traduire par un élargissement intempestif de l'affectation spectrale des informations PSI.

Le fait d'utiliser une fréquence de 25 Hz pour les deux tableaux d'informations PSI apporte au temps d'acquisition du signal une contribution de type cas le moins favorable, soit environ 80 ms. Cela ne se produira que lorsque les données requises de la table PAT seront "manquées à flux tendu" et que, une fois la table PAT acquise et décodée, les données requises de la table PMT seront également "manquées à flux tendu". Ce doublage du temps d'acquisition dans le cas le moins favorable est un des inconvénients apportés par l'augmentation – due à la structure de la table PAT – du niveau de non-visibilité directe des adresses réelles des données. On pourra réduire cet effet en coordonnant la transmission des paquets PAT et PMT associés. On peut penser que cet inconvénient est compensé par l'avantage de cette méthode pour effectuer des opérations de remultiplexage après suppression/adjonction de données.

Compte tenu de cette fréquence de 25 Hz, on peut construire les exemples suivants (qui laissent tous une marge confortable pour les diverses servitudes de couche Liaison de données, de correction d'erreurs sans voie de retour, d'accès conditionnel et d'aiguillage).

Canal de TV par câble à 6 MHz

- cinq programmes à 5,2 Mbit/s: 26,5 Mbit/s (servitude binaire de transport incluse)
- spectre total des informations PSI: 5,2 kbit/s
- spectre d'accès conditionnel: 500 kbit/s

  spectre total de transport selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1: 27,1 Mbit/s
- servitude binaire des informations PSI: 0,28 %

Canal à porteuse optique de niveau 3 (OC-3 à 155 Mbit/s)

- 32 programmes à 3,9 Mbit/s: 127,5 Mbit/s (servitude binaire de transport incluse)
- spectre total des informations PSI: 225,6 kbit/s
- spectre d'accès conditionnel: 500 kbit/s

  spectre total de transport selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1: 128,2 Mbit/s
- servitude binaire des informations PSI: 0,18 %

Canal de répéteur en bande C à bord de satellite

- 128 programmes audio à 256 kbit/s: 33,5 Mbit/s (servitude binaire de transport incluse)
- spectre total des informations PSI: 826,4 kbit/s
- spectre d'accès conditionnel: 500 kbit/s

  spectre total de transport selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1: 34,7 Mbit/s
- servitude binaire des informations PSI: 2,4% (chiffre qui serait en fait plus faible si chaque programme n'utilisait qu'un seul identificateur PID)

Comme prévu, le pourcentage de servitude augmente pour les services à plus faible débit étant donné qu'il est alors possible d'acheminer beaucoup plus de services dans un même flux de transport. Cette servitude binaire n'est cependant pas excessive dans tous les cas de figure. On pourra utiliser, pour les données d'informations PSI, des débits supérieurs (à 25 Hz) afin d'atténuer leur incidence sur le temps d'acquisition de canal, au prix de modestes augmentations du débit nécessaire.

Annexe D

Implications de la présente Recommandation | Norme internationale (systèmes) quant au modèle de synchronisation et à ses applications

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

D.0 Introduction

La Rec. UIT-T H.222.0 | ISO/CEI 13818-1 présente un modèle spécifique de gestion du temps pour l'échantillonnage, le codage, la mémorisation dans le codeur, l'émission, la réception, la mémorisation dans le décodeur, le décodage et la présentation de données audionumériques combinées à des données vidéonumériques. Ce modèle est directement intégré dans la spécification des prescriptions syntaxiques et sémantiques des flux de données conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. En admettant qu'un système de décodage reçoit un flux binaire conforme, acheminé correctement selon le modèle de gestion du temps, il est aisé d'implémenter ce décodeur de manière qu'il produise en sortie des signaux audio et vidéo de haute qualité, convenablement synchronisés. Il n'est toutefois pas prescrit, à titre normatif, que les décodeurs soient pilotés de façon à fournir des signaux de sortie d'un tel niveau de qualité pour la présentation. Dans les applications où les données ne sont pas injectées dans le décodeur avec la base de temps correcte, il reste possible d'obtenir en sortie le signal souhaité de présentation; mais de telles capacités ne sont généralement pas garanties. La présente annexe informative décrit en détail le modèle de gestion du temps selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1; elle contient quelques suggestions sur la façon de piloter des systèmes de décodage afin de répondre à certaines applications représentatives.

D.0.1 Modèle de gestion du temps

La Rec. UIT-T H.222.0 | ISO/CEI 13818-1 offre un modèle de gestion du temps dans lequel toutes les images numérisées et tous les échantillons audio numérisés qui pénètrent dans le codeur sont présentés à la sortie du décodeur exactement une fois chacun, après un délai de transmission de bout en bout constant. Les fréquences des échantillons, c'est-à-dire la fréquence des trames vidéo et celle des échantillons audio, sont précisément les mêmes dans l'étage de décodage et dans l'étage de codage. Ce modèle de gestion du temps est schématisé dans la Figure D.1.

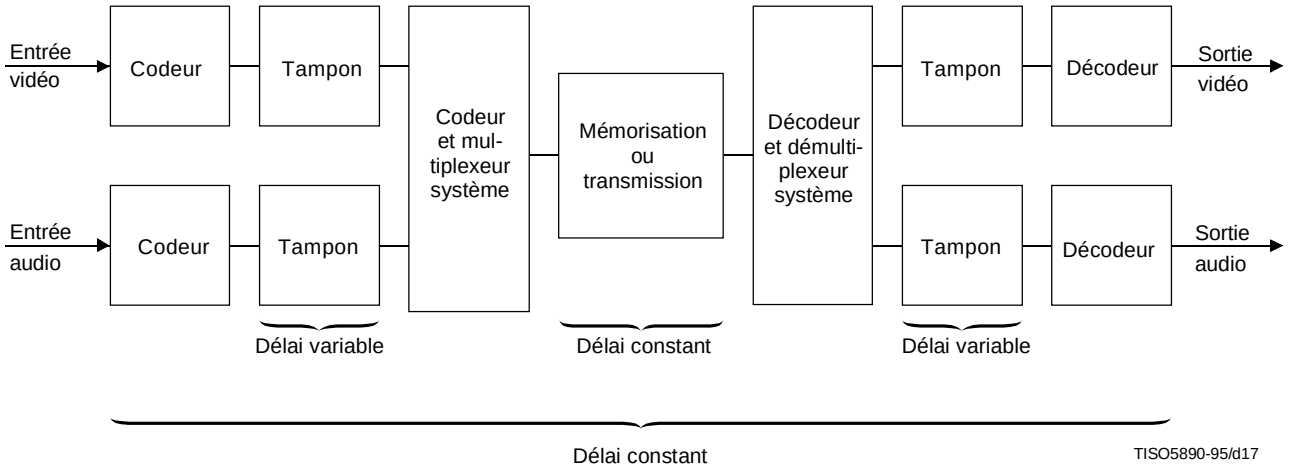


Figure D.1 – Modèle à délai constant

Comme indiqué sur la Figure D.1, le délai entre l'entrée des données dans le codeur et leur sortie du décodeur ou leur présentation est constant dans ce modèle[1)], alors que le délai dû à chaque mémorisation dans le codeur et dans le décodeur est variable. Non seulement le délai dû à la mémorisation dans chacun de ces tampons est variable à l'intérieur du conduit numérique d'un même flux élémentaire mais ce délai de mémorisation diffère selon qu'il s'agit d'un conduit vidéo ou d'un conduit audio. La position relative du codage binaire représentant des données audio ou des données

1) Un délai constant, tel qu'indiqué pour le système entier, est nécessaire pour effectuer une synchronisation correcte. Certains écarts sont toutefois possibles. Le délai dû au réseau est considéré comme constant, avec tolérance de légères variations. L'adaptation réseau peut entraîner de plus grandes variations du délai réseau. Ces deux éléments seront examinés plus bas.

vidéo dans le flux audiovisuel combiné ne fournira donc pas d'informations de synchronisation. L'emplacement relatif des données audio et vidéo codées n'est contraint que par le décodeur modèle du système (STD), de manière que les tampons de ce décodeur aient un comportement toujours correct. Les données audio et vidéo codées qui représentent des sons et des images à présenter en simultanéité pourront donc se trouver temporellement dissociées à l'intérieur du flux binaire codé. Cet écart pourra atteindre une seconde, ce qui est le délai maximal autorisé dans le modèle STD pour la mémorisation dans le décodeur.

Les fréquences d'échantillonnage audio et vidéo dans le codeur diffèrent nettement l'une de l'autre et peuvent être ou ne pas être en rapport exact et fixe l'une avec l'autre. Cela dépendra du fait que le flux audiovisuel combiné est un flux de programme ou un flux de transport et du fait que les fanions System_audio_locked et System_video_locked sont ou non activés dans le flux de programme. La durée d'un bloc d'échantillons audio (constituant une unité de présentation audio) n'est généralement pas la même que celle d'une image vidéo complète.

Il existe dans le codeur une unique et commune horloge système. Cette horloge sert à construire des pointeurs temporels ('horodateurs') qui indiquent les instants corrects de présentation et de décodage des données audio et vidéo. Elle sert également à construire des pointeurs temporels qui indiquent, à intervalles échantillonnés, les valeurs instantanées de l'horloge système proprement dite. Les pointeurs temporels qui indiquent les instants de présentation des données audio et des données vidéo sont appelés horodateurs de présentation (PTS); ceux qui indiquent les instants de décodage sont appelés horodateurs de décodage (DTS); et ceux qui indiquent la valeur de l'horloge système sont appelés référence temporelle du système (SCR) dans les flux de programme et référence temporelle du programme (PCR) dans les flux de transport. C'est la présence de cette horloge système commune dans le codeur, ce sont les horodateurs qui sont créés sur la base de cette horloge, c'est la régénération de l'horloge dans le décodeur et c'est l'usage correct des horodateurs qui donnent la possibilité de synchroniser convenablement le fonctionnement du décodeur.

Les implémentations du codeur peuvent ne pas suivre exactement ce modèle; mais le flux de données issu pratiquement du codeur, du système de mémorisation, du réseau et d'un ou de plusieurs multiplexeurs doit être précisément conforme au modèle. (L'acheminement des données peut s'en écarter un peu, selon l'application.) Dans la présente annexe, le terme "horloge système du codeur" sera utilisé pour désigner soit l'horloge système commune telle qu'elle est effectivement décrite dans ce modèle, soit la fonction équivalente, quelle que soit la façon de l'implémenter.

Etant donné que le délai de transmission de bout en bout dans tout le système est constant, les présentations des données audio et vidéo seront synchronisées avec précision. La construction de flux binaires de données systèmes est soumise à des contraintes telles que la garantie ne soit donnée que lorsque ces données sont décodées par un décodeur conforme à ce modèle et muni de tampons de capacité appropriée, il n'y aura jamais ni surremplissage ni sous-remplissage de ces mémoires, sauf exceptions spécifiques autorisant un sous-remplissage intentionnel.

Afin que le système de décodage apporte le délai précis qui fera que le délai de bout en bout reste constant, il est nécessaire que ce décodeur soit calé sur une horloge système dont la fréquence de fonctionnement et la valeur instantanée absolue correspondent à celles de ce décodeur. Les informations nécessaires pour acheminer l'horloge système du codeur sont codées dans les références SCR ou PCR; cette fonction est expliquée ci-dessous.

Les décodeurs qui sont implémentés conformément à ce modèle de gestion du temps, tels qu'ils ne présentent qu'une seule fois exactement, à débit constant, chaque échantillon audio et chaque image vidéo (sauf exceptions spécifiques codées intentionnellement) et tels que leurs mémoires tampons aient le comportement décrit dans ce modèle, sont signalés dans la présente annexe comme étant des décodeurs précisément rythmés ou produisant des signaux précisément rythmés. La présente Norme internationale ne prescrit pas que les décodeurs implémentés doivent présenter les données audio et vidéo conformément à ce modèle; il est possible de construire des décodeurs qui n'ont pas de délai constant ou, ce qui est équivalent, qui ne présentent pas exactement une seule fois chaque image ou échantillon audio. Dans ces implémentations, il se peut toutefois que le synchronisme entre données audio et vidéo présentées ne soit pas précis et que le comportement des tampons de décodage ne suive pas le modèle de référence du décodeur. Il importe d'éviter un surremplissage des mémoires tampons de décodage car cela peut provoquer une perte de données pouvant avoir des conséquences notables sur le processus de décodage résultant. Cette annexe couvre en premier lieu le fonctionnement de tels décodeurs précisément rythmés ainsi que certaines des options qui sont offertes lors de leur implémentation.

D.0.2 Synchronisation de la présentation des données audio et vidéo

Dans le cadre du codage de données conformément à la présente Recommandation | Norme internationale, il existe des pointeurs temporels relatifs aux instants de présentation et de décodage des images vidéo et des blocs d'échantillons audio. Ces images et ces blocs sont appelés "unités de présentation" (PU en abrégé). Les ensembles d'éléments binaires codant pour les unités de présentation et inclus dans le flux binaire selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 sont appelés "unités d'accès" (AU en abrégé). Une unité d'accès audio est abrégée en "AAU" et une unité d'accès vidéo par "VAU". Dans ISO/CEI 13818-3 concernant les données audio, le terme "trame audio" a le même sens qu'une unité d'accès audio (AAU, *audio access unit*) ou qu'une unité de présentation audio (APU, *audio presentation unit*), selon le contexte. Une unité de présentation vidéo (VPU, *video presentation unit*) est une image décodée et une unité d'accès vidéo (VAU, *video access unit*) est une image codée.

Certaines unités AAU et VAU (pas nécessairement toutes) ont des pointeurs PTS associés. Un pointeur PTS indique l'instant où il convient de présenter à l'utilisateur l'unité de présentation qui résulte du décodage de l'unité d'accès associée à ce pointeur PTS. Les pointeurs PTS audio et les pointeurs PTS vidéo sont tous les deux des échantillons issus d'une même base de temps, appelée "horloge de base du système" (STC). Avec les valeurs correctes des pointeurs PTS audio et vidéo incluses dans le flux de données et avec la présentation des unités APU et AVU aux instants indiqués par les pointeurs PTS appropriés sur la base de l'horloge STC commune, le système de décodage peut réaliser une synchronisation précise des données audio et vidéo présentées. Bien que l'horloge STC ne fasse pas partie du contenu normatif de la présente Recommandation | Norme internationale et que les informations équivalentes soient désignées dans la présente Recommandation | Norme internationale par des termes tels que "fréquence d'horloge système", l'horloge STC est un élément important et pratique pour expliquer le modèle de gestion du temps. Il est d'utilité générale d'utiliser des codeurs et des décodeurs qui soient calés d'une certaine façon sur l'horloge STC.

Les pointeurs PTS sont nécessaires pour acheminer des informations précises de relations temporelles entre données audio et vidéo car les unités APU et VPU ont généralement des durées nettement différentes et généralement sans rapport entre elles. Par exemple, des unités APU comportant chacune 1152 échantillons à une fréquence d'échantillonnage de 44 100 échantillons par seconde ont une durée d'environ 26,12 ms; et des unités VPU à une fréquence image de 29,97 Hz ont une durée d'environ 33,76 ms. En général, les limites temporelles des unités APU et VPU coïncident rarement, sinon jamais. Des pointeurs PTS séparés pour les données audio et vidéo apportent les renseignements permettant de préciser la relation temporelle entre unités APU et VPU, sans exiger aucune relation spécifique entre durée et intervalle de ces unités.

Les valeurs des champs des pointeurs PTS sont définies par rapport à celles du décodeur modèle du système (STD), ce qui est une contrainte normative fondamentale pour tous les flux binaires. Le décodeur STD est un modèle mathématique qui représente un décodeur théorique qui spécifie précisément le mouvement de tous les éléments binaires entrant et sortant des mémoires tampons du décodeur. La contrainte sémantique de base qui est imposée au flux binaire est que les mémoires tampons contenues dans le décodeur STD ne doivent jamais présenter de surremplissage ni de sous-remplissage, sauf exceptions spécifiques concernant, dans des cas spéciaux, le sous-remplissage. Dans le modèle STD, le décodeur virtuel est toujours synchronisé exactement avec la source de données et les instants de décodage et de présentation des données audiovisuelles sont toujours exactement synchronisés. Bien qu'il soit exact et cohérent, le décodeur STD est un peu simplifié par rapport aux implémentations concrètes de décodeurs, afin de rendre sa spécification plus claire et d'en faciliter une large implémentation dans divers systèmes de décodage. Dans le modèle STD, en particulier, chacune des opérations sur le flux binaire est effectuée instantanément dans le décodeur, sauf évidemment le temps passé par les éléments binaires dans les mémoires tampons du décodeur. Dans un système de décodage réel, les décodeurs audio et vidéo individuels ne travaillent pas de façon instantanée et leurs délais doivent être pris en considération lors de la conception de l'implémentation. Par exemple, si des images vidéo sont décodées dans exactement l'intervalle de présentation d'une seule image, 1/P (où P est la fréquence de trame) et que les données vidéo comprimées arrivent au décodeur à un débit R, le processus de suppression des bits associés à chaque image sera retardé d'un temps 1/P par rapport à l'instant indiqué dans les champs des pointeurs PTS et DTS, si bien que la mémoire tampon du décodeur de données vidéo doit être plus grande de R/P que celle qui est spécifiée dans le modèle STD. La présentation des données vidéo sera d'autant retardée par rapport au décodeur STD et il y aura lieu de gérer les pointeurs PTS en conséquence. Comme les données vidéo sont retardées, il y a lieu de retarder le décodage et la présentation des données audio d'une durée similaire afin d'assurer une synchronisation correcte. Ce retardement du décodage et de la présentation de données audio et vidéo dans un décodeur pourra, par exemple, être effectué par addition d'une constante aux valeurs actuelles des pointeurs PTS opérationnels dans le décodeur.

Une autre différence entre le décodeur STD et une implémentation concrète de décodeur précisément rythmé est que dans le modèle STD on part de l'hypothèse explicite que le signal final de sortie audio et vidéo sera présenté instantanément à l'utilisateur, sans autre délai. Cela n'est pas toujours le cas en pratique, en particulier avec des écrans d'affichage par tube à rayons cathodiques. Il y aura donc lieu de prendre également en compte ce délai supplémentaire lors de l'étude de conception. Les codeurs sont appelés à coder des données audio et vidéo de telle manière qu'un synchronisme correct soit réalisé lorsque ces données seront décodées avec le modèle STD. Le codeur doit tenir compte des délais d'injection et d'échantillonnage des données audio et vidéo, comme le temps d'intégration de charge optique d'un coupleur de caméra vidéo.

Dans le modèle STD, on suppose que la synchronisation est correcte et c'est sur cette base que l'on teste les pointeurs temporels et le comportement des mémoires tampons afin de déterminer si le flux binaire est valide. Il est évident que, dans un décodeur physique, une synchronisation précise n'est pas automatiquement réalisée, en particulier lors du démarrage et en présence d'une gigue de rythme. Une temporisation précise du décodeur est un but que doivent viser les concepteurs de décodeur. L'inexactitude du rythme d'un décodeur affecte le comportement de ses mémoires tampons. Ces points seront examinés plus en détail dans d'autres paragraphes de la présente annexe.

Le décodeur STD comprend des champs DTS (horodateurs de décodage) et des champs PTS (horodateurs de programme). Le pointeur DTS indique l'instant où une unité d'accès doit être extraite de la mémoire du décodeur pour décodage dans le modèle STD. Comme ce décodage des flux élémentaires audio et vidéo est instantané dans le modèle STD, les instants de décodage et de présentation sont identiques dans la plupart des cas. La seule exception se

produit avec des images vidéo qui ont subi un réordonnancement à l'intérieur du flux binaire codé, c'est-à-dire dans le cas de séquences vidéo sans faible délai d'images à codage I et à codage P. S'il y a réordonnancement d'images, on utilise un tampon dans le décodeur vidéo pour stocker une image décodée de type I ou P, selon le cas, jusqu'à ce qu'il y ait lieu de présenter cette image. Le pointeur DTS n'est pas codé si les instants de décodage et de présentation sont identiques dans le décodeur STD, c'est-à-dire en présence d'unités AAU, d'unités VAU à codage B, et d'unités VAU à codage I ou P dans des séquences vidéo à faible délai; sinon, le pointeur DTS aurait la même valeur que le pointeur PTS. Si un de ces deux types de pointeur est codé et n'a pas la même valeur que l'autre, celui-ci est également codé. Pour toutes les unités d'accès où seul le pointeur PTS est codé, on peut interpréter ce champ comme représentant un pointeur aussi bien de type PTS que DTS.

Etant donné qu'il n'y a pas besoin de valeurs PTS et DTS pour chaque unité AAU et VAU, le décodeur peut choisir d'interpoler les valeurs qui ne sont pas codées. Il est requis que les valeurs de pointeurs PTS ne soient pas séparées par des intervalles supérieurs à 700 ms dans chaque flux élémentaire audio ou vidéo. Ces intervalles de temps sont mesurés en instants de présentation, c'est-à-dire dans le même contexte que les valeurs des champs correspondants et non sur l'échelle des temps d'émission et de réception de ces champs. En cas de flux de données avec verrouillage des horloges système, vidéo et audio, comme défini dans la partie normative de la présente Recommandation | Norme internationale, chaque unité d'accès qui suit une unité pour laquelle un pointeur DTS ou PTS est explicitement codé possède un instant de décodage équivalant à la somme de l'instant de décodage de l'unité d'accès précédente plus un écart fixe et spécifié en unités d'horloge STC. Par exemple, pour des données vidéo codées à une fréquence image de 29,97 Hz, chaque image aura, par rapport à l'image précédente, un écart temporel de 3003 périodes de la portion rythmée à 90 kHz de l'horloge STC, lorsque le rapport entre les fréquences vidéo et horloge système est verrouillé. La même relation temporelle existe pour le décodage d'unités d'accès successives, bien que le délai de réordonnancement des images dans le décodeur ait une incidence sur la relation entre les unités d'accès au décodeur et les unités de présentation en sortie. Si le flux de données est codé de manière que la fréquence vidéo ou audio ne soit pas calée sur l'horloge système, l'écart temporel entre les instants de décodage d'unités d'accès successives pourra être estimé sur la base des mêmes valeurs comme indiqué ci-dessus; mais ces écarts temporels ne seront pas exacts parce que les relations entre fréquence de trame, fréquence d'échantillonnage audio et fréquence d'horloge système ne sont pas exactes dans le codeur.

On notera que les champs de pointeurs PTS et DTS ne suffisent pas à indiquer le remplissage correct des mémoires tampons du décodeur lors du démarrage ou à un autre moment; de même, ils n'indiquent pas la durée du délai qui devra s'écouler entre la réception des premiers bits d'un flux de données et le début de leur décodage. Ces informations seront extraites par combinaison des fonctions remplies par les champs PTS et DTS avec recalage correct d'horloge, ce qui est examiné ci-dessous. Dans le modèle STD, donc dans les décodeurs conformes à ce modèle, le comportement du tampon de décodage est totalement déterminé par les valeurs des références SCR (ou PCR), par les instants de leur réception ainsi que par les valeurs des pointeurs PTS et DTS, en admettant que les données sont acheminées conformément au modèle de gestion du temps. Ces informations spécifient le temps passé par les données codées dans les mémoires du décodeur. La quantité de données codées qui est en mémoire tampon n'est pas explicitement spécifiée mais ces renseignements ne sont pas nécessaires étant donné que le rythme est entièrement spécifié. On notera également que le degré de remplissage des mémoires de données peut varier considérablement dans le temps, d'une façon qui ne peut pas être prédite par le décodeur, sinon par utilisation correcte des pointeurs temporels.

Pour que les champs PTS des données audio et vidéo se rapportent correctement à une base STC commune, il est nécessaire de mettre à leur disposition, dans le système de décodage, une horloge commune, correctement rythmée. Cela sera étudié dans le paragraphe suivant.

D.0.3 Recalage sur l'horloge de base du système dans le décodeur

Le flux de données conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 contient, en plus des champs PTS et DTS, des pointeurs temporels sur l'horloge de référence. Ces références, échantillonnées sur l'horloge de base du système, sont applicables aussi bien à un décodeur qu'à un codeur. Elles ont une résolution d'une partie sur 27 000 000 par seconde et apparaissent à intervalles de 100 ms au plus dans les flux de transport ou de 700 ms au plus dans les flux de programme. En tant que telles, ces références peuvent servir à implémenter des boucles de commande de recalage temporel dans les décodeurs avec une précision suffisante pour toutes les applications répertoriées.

Dans le flux de programme, le champ de référence temporelle est appelé horloge de base du système (SCR). Dans le flux de transport, le champ de référence temporelle est appelé référence temporelle du programme (PCR). En général, les définitions des références SCR et PCR peuvent être considérées comme équivalentes, bien qu'elles diffèrent un peu. Dans le reste du présent paragraphe, on utilisera le terme SCR pour plus de clarté; les mêmes remarques s'appliqueront implicitement à la référence PCR, sauf indication contraire. Dans les flux de transport, la référence PCR fournit la référence temporelle pour un programme donné, qui se compose d'un ensemble de flux élémentaires ayant une base de temps commune et destinés à un décodage et à une présentation en synchronisme. Un même flux de transport peut comporter plusieurs programmes, chacun ayant une base de temps indépendante et un ensemble distinct de références PCR.

Le champ SCR indique la valeur correcte de l'horloge STC lorsque ce champ est reçu par le décodeur. Comme le champ SCR occupe plusieurs octets de données et que les flux de données systèmes sont définis comme des trains

d'octets, la référence SCR arrive, par définition, dans le décodeur au moment où le dernier octet du champ SCR_base est reçu par le décodeur. En variante, la référence SCR peut être interprétée comme l'instant où le champ SCR est censé arriver au décodeur, en supposant que l'horloge STC est déjà reconnue comme correcte. Le choix de l'interprétation à retenir dépend de la structure du système d'application. Dans les applications où l'origine des données peut être commandée par le décodeur, comme les supports DSM en lien local, le décodeur peut avoir une fréquence d'horloge STC autonome, si bien qu'il n'a pas à se recaler sur l'horloge STC centrale. Dans de nombreuses et importantes applications, cette hypothèse ne peut toutefois pas être faite à bon escient. Si l'on considère par exemple le cas d'un flux de données acheminé simultanément à plusieurs décodeurs, dont chacun possède sa propre horloge STC avec sa propre fréquence d'horloge indépendante, on ne pourra pas garantir que les références SCR arriveront au moment voulu à tous ces décodeurs. Un décodeur donné aura en général besoin des références SCR à un moment antérieur à celui de sa fourniture par la source, alors qu'un autre décodeur en aura besoin à un moment ultérieur. On ne peut pas résoudre cette différence au moyen d'une mémoire tampon de données de capacité finie pour une durée non délimitée de réception de données. Les informations ci-dessous répondront donc surtout au cas où l'horloge STC doit toujours asservir son rythme sur les références SCR (ou PCR) reçues.

Dans un flux de données selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, correctement construit et acheminé, chaque référence SCR arrive au décodeur à l'instant précis qui est indiqué par la valeur de cette référence SCR. Dans ce contexte, le terme "instant" implique que l'horloge STC a une valeur correcte. Dans cette hypothèse, cette valeur d'horloge STC est la même que celle de l'horloge STC du codeur au moment où la référence SCR a été mémorisée ou émise. Le codage peut toutefois avoir été effectué en temps non réel ou bien le flux de données peut avoir été modifié depuis son codage initial. En général, on pourra implémenter le codeur ou la source de données de diverses manières, telles que l'horloge STC du codeur puisse être une grandeur théorique.

Si la fréquence d'horloge du décodeur correspond exactement à celle du codeur, le décodage et la présentation des données audiovisuelles se feront exactement au même débit que dans le codeur, si bien que le délai aux extrémités sera constant. Avec des fréquences d'horloge codeur et décodeur adaptées, toute valeur correcte de référence SCR pourra être utilisée pour régler la valeur instantanée de l'horloge STC au décodeur et, à partir de cet instant, l'horloge STC au décodeur correspondra à celle du codeur sans qu'il soit besoin d'autre réglage. Cet état restera vrai jusqu'à ce qu'il y ait une solution de continuité dans le rythme, comme la fin d'un flux de programme ou l'apparition d'un indicateur de discontinuité dans un flux de transport.

En pratique, une fréquence indépendante d'horloge système dans un décodeur ne correspondra pas à la fréquence d'horloge système d'un codeur, échantillonnée et signalée par les valeurs de référence SCR. L'horloge STC du décodeur pourra être calée en asservissement par le codeur au moyen des références SCR reçues. La méthode type d'asservissement de l'horloge décodeur au flux de données reçu est celle d'une boucle PLL [asservissement par boucle de phase (PLL, *phase-locked loop*)]. On pourra utiliser des variantes de la méthode PLL de base ou d'autres méthodes, selon les nécessités de l'application spécifique.

La Figure D.2 décrit schématiquement une boucle PLL élémentaire, qui recale l'horloge STC dans un décodeur.

La Figure D.2 montre une boucle PLL classique, sauf que les termes de référence et de rétroaction sont des nombres (valeurs d'horloge STC et de référence SCR ou PCR) au lieu d'événements de signal, comme des fronts.

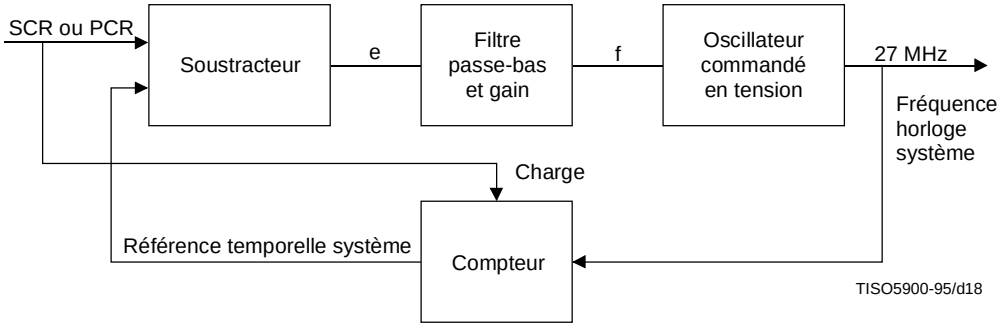


Figure D.2 – Recalage d'horloge STC par verrouillage de phase (PLL)

Dès qu'une nouvelle base de temps (c'est-à-dire un nouveau programme) commence à être acquise, l'horloge STC est calée sur la valeur actuellement codée dans les références SCR. Normalement, la première référence SCR est injectée directement dans le compteur d'horloge STC et la boucle PLL fonctionne ensuite en circuit fermé. Des variantes de cette méthode peuvent être appropriées, c'est-à-dire si on peut suspecter les valeurs des références SCR d'être entachées de gigue ou d'erreurs.

L'action en boucle fermée du verrouillage PLL est la suivante. Au moment où chaque référence SCR (ou PCR) arrive au décodeur, sa valeur est comparée à la valeur actuelle de l'horloge STC. La différence est un nombre dont une partie est exprimée en unités de 90 kHz et l'autre partie en termes égaux à 300 fois cette fréquence, c'est-à-dire 27 MHz. La valeur de cette différence est normalisée de manière à occuper l'espace d'un seul nombre, normalement en unités de 27 MHz, et est appelée "**e**", terme d'erreur dans la boucle. La séquence des termes "**e**" est injectée dans l'étage de filtrage passe-bas et de gain, conçu en fonction des besoins de l'application. A la sortie de cet étage, un signal de commande "**f**" pilote la fréquence instantanée de l'oscillateur commandé en tension (VCO, *voltage controlled oscillator*). La sortie de l'oscillateur VCO est un signal pilote de fréquence nominale 27 MHz; ce signal sert de fréquence d'horloge système dans le décodeur. Cette horloge à 27 MHz est injectée dans un compteur qui produit les valeurs actuelles d'horloge STC, composées d'une valeur de 90 kHz issue d'une division par 300 du signal à 27 MHz et d'une valeur de base à 90 kHz obtenue en accumulant les résultats à 90 kHz dans un compteur à 33 éléments binaires. Cette base, comptée sur 33 éléments binaires, des 90 kHz en sortie d'horloge STC est utilisée, en cas de besoin, pour effectuer des comparaisons avec les valeurs des pointeurs PTS et DTS. Le signal d'horloge STC complet est également envoyé en rétroaction dans le soustracteur.

L'intervalle maximal, délimité, entre références SCR successives (700 ms) ou PCR successives (100 ms) permet de concevoir et de monter des boucles PLL de stabilité reconnue. La largeur de bande de ces boucles PLL a une limite supérieure, imposée par la durée de cet intervalle. Comme indiqué ci-dessous, la boucle PLL requise possède, dans de nombreuses applications, une bande passante très étroite; si bien que cette limite ne se traduit pas par des contraintes notables sur la conception et la performance du décodeur.

Si la fréquence propre ou initiale de l'oscillateur VCO est assez proche de la valeur correcte de la fréquence d'horloge système du codeur, le décodeur sera sans doute en mesure de fonctionner de manière satisfaisante dès que l'horloge STC sera correctement initialisée, avant que la boucle PLL ait atteint un état de verrouillage défini. Pour une fréquence donnée d'horloge STC dans un décodeur, différant d'une quantité délimitée par rapport à la fréquence codée dans les références SCR et ne dépassant pas les limites absolues de fréquence requises par l'application de décodage, l'effet d'une désadaptation entre fréquences d'horloge STC dans le codeur et dans le décodeur sera (s'il n'y avait pas de boucle PLL) une augmentation progressive et inévitable du remplissage des mémoires tampons du décodeur, telle qu'un surremplissage ou un sous-remplissage finira par apparaître dans les mémoires tampons du décodeur, quelle que soit leur capacité limite. La durée admissible avant verrouillage de la fréquence d'horloge STC du décodeur sur celle du codeur dépendra donc de la capacité encore disponible dans la mémoire tampon du décodeur et de son temps d'accès.

Si les références SCR sont reçues par le décodeur avec des valeurs et un rythme qui reflètent des valeurs instantanées correctes des échantillons d'une horloge STC à fréquence constante dans le codeur, le terme d'erreur **e** convergera vers une valeur pratiquement constante une fois que la boucle aura atteint l'état de verrouillage. Cette situation de valeurs correctes de référence SCR est synonyme soit d'une mémorisation et d'une émission à délai constant des données entre codeur et décodeur soit, si ce délai n'est pas constant, de l'équivalent effectif d'une mémorisation et d'une émission à délai constant après correction des valeurs de SCR en fonction des variations du délai. Lorsque les valeurs de **e** convergent vers une constante, les variations de fréquence instantanée de l'oscillateur VCO tendent asymptotiquement vers zéro une fois la boucle verrouillée; l'oscillateur VCO est alors réputé avoir une très faible gigue ou pente de stabilisation en fréquence. Pendant que la boucle PLL est en cours de verrouillage, on peut commander exactement la vitesse d'excursion de fréquence d'oscillateur VCO (ou pente de stabilisation en fréquence) selon la conception de l'étage de filtrage passe-bas et de gain. En général, on peut étudier la pente de stabilisation en fréquence de l'oscillateur VCO de manière à répondre aux besoins des applications, sous réserve des contraintes en termes de capacité et temps d'accès des mémoires tampons.

D.0.4 Gigue des références SCR et PCR

Si un réseau ou un remultiplexeur de flux de transport fait varier le délai d'acheminement du flux de données entre le codeur ou système de stockage et le décodeur, ces variations tendent à introduire une différence entre les valeurs réelles et théoriques des références SCR (ou PCR) au moment de leur réception. Cette différence est appelée gigue de référence SCR ou PCR. Par exemple, si le délai d'acheminement d'une seule référence SCR est plus grand que le retard subi par d'autres champs analogues d'un même programme, cette référence SCR est en retard. De même, si ce délai est plus petit que celui d'autres champs de référence temporelle du programme, le champ de cette référence est en avance.

La gigue de rythme à l'entrée d'un décodeur est calculée en comparant les valeurs des références SCR et les instants où elles sont reçues. En admettant une structure de récupération d'horloge conforme à l'illustration de la Figure D.2, toute gigue de rythme de ce type se retrouvera dans les valeurs du terme d'erreur **e**; et les valeurs non nulles de **e** introduiront des variations dans les valeurs du signal **f** et donc des variations de fréquence dans l'horloge système à 27 MHz. Des variations de fréquence dans l'horloge récupérée seront acceptables ou non acceptables dans les systèmes de décodage, selon les exigences de chaque application. Par exemple, dans des décodeurs précisément rythmés qui produisent un signal vidéo composite, la fréquence d'horloge récupérée est normalement utilisée pour produire l'horloge des échantillons vidéo composites et la sous-porteuse chrominance. Les spécifications applicables à la stabilité en fréquence de sous-porteuse ne permettront qu'un réglage très lent de la fréquence d'horloge système. Dans les applications où une gigue notable affecte les références SCR ou PCR à l'entrée du décodeur et où la pente de stabilisation en fréquence de

l'horloge STC est soumise à des contraintes sévères, un fonctionnement correct ne sera peut-être pas possible compte tenu des contraintes imposées en termes de capacité et de temps d'accès complémentaires de mémoire tampon dans le décodeur.

La présence d'une gigue de référence SCR ou PCR peut être due, par exemple, à une transmission réseau impliquant un multiplexage de paquets ou de cellules ou un débit variable des paquets dans le réseau, dû par exemple à des retards de mise en file d'attente ou à un temps variable d'accès réseau pour des systèmes à supports partagés.

Le multiplexage ou le remultiplexage de flux de transport ou de programme modifie l'ordre et la position temporelle relative des paquets de données, donc aussi des références SCR ou PCR. La modification d'emplacement temporel des références SCR fait que les valeurs des références SCR précédemment correctes deviennent incorrectes car, en général, ces valeurs ne représentent pas correctement l'instant de leur arrivée par l'intermédiaire d'un réseau à retard constant. De même, un flux de programme ou de transport contenant des valeurs correctes de référence SCR ou PCR peut être acheminé par un réseau qui impose un délai variable au flux de données, sans corriger les valeurs SCR ou PCR. Il en résulte de nouveau une gigue des références SCR ou PCR, avec les effets correspondants sur la conception et la performance des décodeurs. Le pire cas de gigue imposée par un réseau aux références SCR ou PCR reçues dans un décodeur dépend d'un certain nombre de facteurs qui sont hors du domaine d'application de la présente Recommandation | Norme internationale, y compris la longueur des files d'attente créées dans chacun des commutateurs du réseau et le nombre total d'opérations de commutation ou de remultiplexage effectuées en cascade dans le réseau sur le flux de données.

Dans le cas d'un flux de transport, une correction des références PCR est nécessaire lors d'une opération de remultiplexage, créant un nouveau flux de transport à partir d'un ou de plusieurs autres flux de transport. Cette correction s'effectue par addition d'un terme de correction à la référence PCR; ce terme sera calculé comme suit:

$$\Delta PCR = del_{act} - del_{const}$$

où del_{act} est le délai réel subi par la référence PCR et où del_{const} est une constante utilisée pour toutes les références PCR de ce programme. La valeur qu'il conviendra d'utiliser pour la constante del_{const} dépendra de la stratégie utilisée par le codeur/multiplexeur original. Cette stratégie pourra consister, par exemple, à émettre les paquets aussitôt que possible, afin de conserver une marge pour le temps de leur transmission par des chaînons ultérieurs. Trois stratégies de multiplexage différentes sont représentées ensemble dans le Tableau D.1, avec la valeur appropriée de la constante del_{const}.

Tableau D.1 – Stratégie de remultiplexage

Stratégie	del_{const}
au plus tôt	del_{min}
au plus tard	del_{max}
au plus juste	del_{avg}

Lors de la conception d'un système, des accords privés peuvent devoir être conclus quant au type de stratégie à utiliser dans les codeurs/multiplexeurs car celle-ci peut avoir une incidence sur la possibilité d'effectuer ultérieurement un éventuel remultiplexage.

La grandeur de la gigue de multiplexage admissible n'est pas délimitée de manière normative dans la présente Recommandation | Norme internationale. Une valeur de 4 ms est toutefois censée être le maximum de gigue dans un système normalement étudié.

Dans les systèmes qui mettent en jeu des remultiplexeurs, des précautions particulières peuvent être nécessaires pour vérifier que les informations contenues dans le flux de transport sont cohérentes. Cela s'applique en particulier aux informations PSI et aux points de discontinuité. Des modifications aux tables d'informations PSI peuvent devoir être insérées dans un flux de transport de manière que des étapes de remultiplexage ultérieures ne déplacent jamais ces informations jusqu'au point où celles-ci deviendraient incorrectes. Par exemple, il convient parfois de ne pas envoyer une nouvelle version de section de table PMT dans les 4 ms après les données affectées par la modification.

De même, il peut être nécessaire qu'un codeur ou multiplexeur évite d'insérer des pointeurs PTS ou DTS dans une fenêtre de ± 4 ms autour d'un point de discontinuité.

D.0.5 Récupération d'horloge en présence d'une gigue de réseau

Plusieurs options existent pour la conception d'un décodeur dans les applications dont les horodateurs de référence temporelle reçus sont affectés d'une gigue notable. La façon dont le décodeur est conçu dépend en grande partie des

caractéristiques recherchées du signal en sortie de décodeur ainsi que des caractéristiques des données d'entrée et de la gigue.

Dans diverses applications, les décodeurs peuvent avoir différentes exigences en termes d'exactitude et de stabilité de l'horloge système récupérée. Le degré requis de cette stabilité et de cette exactitude peut être considéré comme pouvant être mesuré sur un même axe. Une extrémité de ce segment peut être considérée comme représentant les applications où l'horloge système récupérée est utilisée directement pour synthétiser une sous-porteuse chrominance à utiliser pour des données vidéo composites. Cette exigence se présente généralement lorsque les unités vidéo présentées sont de type précisément rythmé, comme décrit ci-dessus. C'est-à-dire que chaque image codée est présentée exactement une seule fois et que le signal de sortie est en vidéo composite conformément aux spécifications applicables. Dans ce cas, la sous-porteuse chrominance, la base de temps des pixels et la fréquence image ont chacune un rapport exactement spécifié les unes avec les autres et avec l'horloge système. La sous-porteuse de données vidéo composites doit toujours avoir au moins une exactitude et une stabilité suffisantes pour que toute boucle de verrouillage PLL de sous-porteuse chrominance d'un téléviseur normal puisse se verrouiller sur la sous-porteuse et que la phase des signaux de chrominance, démodulés sur la base de la sous-porteuse détectée, ne contienne pas de motifs visibles. Certaines applications nécessitent l'utilisation de l'horloge système pour produire une sous-porteuse pleinement conforme aux spécifications NTSC, PAL ou SECAM. Les spécifications de ces applications sont normalement plus sévères que celles qui sont imposées pour des téléviseurs classiques. Par exemple, la spécification SMPTE pour le système NTSC exige une précision de sous-porteuse de 3 millionièmes, avec une gigue maximale à court terme de 1 ns par durée de ligne horizontale et une dérive maximale à long terme de 0,1 Hz par seconde.

Dans les applications où l'horloge système n'est pas utilisée pour élaborer une sous-porteuse de chrominance, cette horloge peut encore servir à construire une base de temps pour les éléments d'image vidéo ou une fréquence d'échantillonnage audio. Ces horloges ont leurs propres caractéristiques de stabilité, qui dépendent des hypothèses faites au sujet du moniteur d'affichage en réception et du degré acceptable de dérive de la fréquence audio ("pleurage et scintillement") à la sortie du décodeur.

Dans les applications où chaque image et chaque échantillon audio ne sont pas présentés exactement une seule fois, c'est-à-dire où un "glissement" des images et échantillons audio est autorisé, l'horloge système peut avoir des exigences relativement peu sévères en termes de précision et de stabilité. Ce type de décodeur peut ne pas avoir une synchronisation précise des unités de présentation audiovisuelles et cette présentation peut ne pas avoir la même qualité qu'avec des décodeurs précisément rythmés.

Le choix des exigences en termes d'exactitude et de stabilité d'horloge système récupérée dépend de l'application. On examinera ci-dessous l'exigence la plus sévère, décrite ci-dessus, où l'horloge système doit servir à construire une sous-porteuse de chrominance.

D.0.6 Horloge système utilisée pour construire une sous-porteuse de chrominance

Les caractéristiques de conception du décodeur peuvent être déterminées d'après celles de la sous-porteuse résultante et du degré maximal acceptable de la gigue réseau. De même, si on connaît les caractéristiques fonctionnelles exigées de l'horloge système et les capacités théoriques du décodeur, on peut déterminer le degré maximal tolérable de gigue réseau. Bien que la formulation de telles prescriptions soit hors du domaine d'application de la présente Recommandation | Norme internationale, les valeurs numériques nécessaires pour spécifier le modèle sont présentées afin de poser plus clairement le problème et de donner un exemple de méthode visant à concevoir un prototype.

Avec un circuit de boucle PLL pour recalage d'horloge comme illustré à la Figure D.2, l'horloge système recalée doit toujours satisfaire aux exigences du pire cas d'excursion de fréquence par rapport à la valeur nominale, mesurée en millionièmes, ainsi que du pire cas de pente de stabilisation en fréquence, mesurée en millionièmes par seconde. La valeur crête à crête non corrigée de la gigue de rythme due au réseau peut être spécifiée en millisecondes. Dans une telle boucle PLL, la gigue de rythme due au réseau apparaît dans le schéma sous la forme du terme d'erreur **e**. Et comme la boucle PLL fait office de filtre passe-bas de gigue à son entrée, l'effet du pire cas sur la fréquence de sortie à 27 MHz se produira lorsque la fonction échelon du rythme de la référence PCR à l'entrée atteindra son amplitude maximale. La valeur **e** a donc une amplitude maximale qui est égale à la gigue crête à crête, représentée numériquement par 2**33 fois la gigue dans la portion de comptage de base du codage de références SCR ou PCR. La vitesse maximale de variation du signal de sortie du filtre passe-bas (LPF, *low pass filter*), **f**, avec cette valeur maximale de **e** à son entrée, détermine directement la pente maximale de stabilisation en fréquence du signal de sortie à 27 MHz. On peut spécifier un filtre passe-bas pour toute valeur maximale donnée du terme **e** et pour toute valeur maximale de la pente de stabilisation en fréquence **f**. Cependant, au fur et à mesure que l'on diminue le gain ou la fréquence de coupure du filtre passe-bas, on augmente le temps nécessaire pour que la boucle PLL se verrouille sur la fréquence représentée par les références SCR ou PCR. La mise en œuvre de boucles PLL possédant de très longues constantes de temps peut être réalisée par l'utilisation de techniques numériques de filtrage passe-bas et éventuellement de techniques de filtrage analogique. Dans des implémentations avec filtres passe-bas numériques où le terme fréquentiel **f** est injecté dans un oscillateur VCO analogique, ce terme **f** est quantifié par un convertisseur numérique-analogique dont l'échelonnement devra être pris en compte lors du calcul de la pente maximale de stabilisation en fréquence à la sortie.

Pour s'assurer que le terme d'erreur **e** converge vers une valeur proche de zéro, le gain en boucle ouverte du circuit PLL doit être très élevé, par exemple en appliquant une fonction d'intégration dans le filtre passe-bas du circuit PLL.

Pour une précision minimale donnée, il peut être logique de concevoir le circuit PLL de manière que sa fréquence de fonctionnement initiale satisfasse à cette précision minimale. Dans ce cas, avant que la boucle PLL soit verrouillée, la fréquence initiale de 27 MHz sera suffisamment précise pour répondre à la stabilité requise de fréquence en sortie. Si l'on pouvait négliger le fait que les mémoires tampons du décodeur finiront par déborder ou s'assécher, cette fréquence initiale d'horloge système serait suffisante pour une exploitation à long terme. Cependant, entre le moment où le décodeur commence à recevoir et à décoder les données et le moment où l'horloge système est calée sur l'instant et la fréquence d'horloge indiqués par les références temporelles SCR ou PCR reçues, les données arrivent dans les tampons à une vitesse différente de celle de leur extraction. En d'autres termes, le décodeur extrait les unités d'accès à des instants qui diffèrent de ceux du décodeur modèle du système (STD). Les tampons du décodeur continueront à se remplir ou à se vider différemment de ceux du modèle STD, selon l'évolution de la fréquence d'horloge système récupérée par rapport à la fréquence d'horloge du codeur. Le degré de remplissage des tampons du décodeur augmentera ou diminuera en fonction de la relation entre la fréquence initiale de l'oscillateur VCO et la fréquence d'horloge système du codeur. Dans l'hypothèse où cette relation est inconnue, il faut prévoir une capacité complémentaire de mémoire tampon afin de tenir compte de ces deux éventualités. Il y a lieu de concevoir le décodeur de manière que toutes les opérations de décodage soient retardées d'une durée au moins égale à celle qui est représentée par la capacité complémentaire de mémoire tampon qui est ajoutée pour le cas où la fréquence initiale de l'oscillateur VCO serait supérieure à celle de l'horloge du codeur, afin d'éviter un sous-remplissage de tampon. Si la fréquence initiale de l'oscillateur VCO n'est pas suffisamment précise pour répondre aux exigences indiquées en termes d'exactitude, il faut que le circuit PLL arrive à l'état verrouillé avant que le décodage puisse commencer; et différents ensembles de considérations s'appliquent au sujet du comportement du circuit PLL pendant ce temps et au sujet de la capacité complémentaire appropriée de mémoire tampon et de délai statique.

Une fonction échelon appliquée à la gigue de rythme d'entrée, produisant un échelonnement du terme d'erreur **e** dans le circuit PLL de la Figure D.2, doit toujours produire un terme de fréquence de sortie **f** tel que, lorsqu'il est multiplié par le gain d'oscillateur VCO, la vitesse maximale de variation soit inférieure à la pente de stabilisation en fréquence spécifiée. Le gain de l'oscillateur VCO est exprimé sous la forme de la grandeur de l'excursion de fréquence de sortie en réponse à une modification de la grandeur de commande à l'entrée. Une autre contrainte sur le filtre LPF dans le circuit PLL est que la valeur statique du terme d'erreur **e** doit, lorsque la boucle est verrouillée, être délimitée afin de délimiter la capacité complémentaire à ajouter aux mémoires tampons ainsi que le délai statique de décodage associé. Ce terme est minimisé lorsque le filtre LPF possède un gain en courant continu très élevé.

On peut juger utile de concevoir des circuits de recalage ou récupération d'horloge légèrement différents de ceux qui sont représentés sur la Figure D.2. Il est par exemple possible de construire une boucle de commande utilisant un oscillateur à commande numérique (NCO, *numerically controlled oscillator*) au lieu d'un VCO. Le NCO utilise un oscillateur à fréquence fixe et des périodes d'horloge sont insérées ou supprimées des événements périodiques normaux du signal de sortie, afin de régler le rythme de décodage et de présentation. Ce procédé peut présenter quelques difficultés lorsqu'il est utilisé avec des données vidéo composites car il expose soit à des déphasages gênants de la sous-porteuse ou à une gigue du rythme de balayage horizontal ou vertical. Une solution peut consister à régler la fréquence des balayages horizontaux au début de l'intervalle de suppression vertical tout en conservant la phase de la sous-porteuse de chrominance.

En résumé, selon les valeurs spécifiées pour les caractéristiques, il sera possible ou impossible de construire un décodeur qui reconstruira l'horloge système avec suffisamment de précision et de stabilité, tout en conservant les capacités voulues des mémoires tampons du décodeur et le délai de décodage ajouté.

D.0.7 Reconstruction des données vidéo et audio en composantes

Si le décodeur produit des données en composantes vidéo, les exigences en termes de précision et de stabilité du rythme seront généralement moins sévères que dans le cas des données vidéo composites. Normalement, la tolérance sur la fréquence est celle que les circuits de déviation d'affichage peuvent accepter. La tolérance sur la stabilité est déterminée par la nécessité d'éviter tout déplacement visible de l'image affichée.

Les principes décrits ci-dessus s'appliquent également ici, mais les exigences spécifiques sont généralement plus faciles à respecter.

La récupération de la fréquence des échantillons audio suivra aussi les mêmes principes mais l'exigence de stabilité sera déterminée par la grandeur de la variation subie à court terme et à long terme par la fréquence des échantillons. L'utilisation d'un procédé de verrouillage PLL, comme illustré dans le paragraphe précédent, permet d'obtenir une excursion à court terme très petite. L'excursion de fréquence à plus long terme se manifeste par une variation de la tonie perçue. Ici encore, on peut définir des règles de conception spécifiques, une fois fixées les limites de cette variation.

D.0.8 Glissement des trames

Dans certaines applications qui n'exigent pas de décodeurs précisément rythmés, l'horloge de base système du décodeur ne peut pas toujours ajuster sa fréquence de fonctionnement sur la fréquence indiquée par les références SCR (ou PCR) reçues. Cette horloge STC du décodeur peut au contraire avoir une fréquence indépendante de 27 MHz, tout en restant asservie au rythme des données reçues. Dans ce cas, la valeur d'horloge STC doit être actualisée en fonction des besoins afin de correspondre aux références SCR reçues. L'actualisation de l'horloge STC dès réception de références SCR provoque des discontinuités dans la valeur d'horloge STC. La grandeur de ces discontinuités dépend de la différence entre la fréquence de 27 MHz du décodeur et la fréquence de 27 MHz du codeur (c'est-à-dire de l'écart par rapport aux références SCR reçues), ainsi que de l'intervalle temporel entre références SCR ou PCR successivement reçues. Etant donné que la fréquence d'horloge système du décodeur à 27 MHz n'est pas calée sur celle des données reçues, elle ne peut pas être utilisée pour produire les horloges d'échantillonnage vidéo ou audio tout en sauvegardant les hypothèses de présentation précisément rythmée des unités VPU et APU exactement une fois ainsi qu'en conservant le même débit de présentation vidéo et audio dans le décodeur et dans le codeur, avec synchronisation audiovisuelle précise. Cette structure autorise de multiples possibilités d'implémentation de systèmes de décodage et de présentation.

Dans un certain type d'implémentation, les images complètes et les échantillons audio sont décodés au moment indiqué par l'horloge STC du décodeur bien qu'ils soient présentés à des instants légèrement différents, en fonction des rythmes d'échantillonnage produits localement. Selon les relations temporelles entre l'échantillonnage du décodeur et l'horloge système du codeur, les images et les échantillons audio pourront parfois être présentés plusieurs fois chacun ou pas du tout: ce phénomène (qui peut introduire des motifs perceptibles) est appelé "glissement de trames" (ou "glissement d'échantillons" dans le cas de données audio). Le synchronisme audiovisuel ne sera en général pas précis à cause des intervalles temporels pendant lesquels on répète ou supprime des images et éventuellement des unités de présentation audio (APU). Selon la mise en œuvre particulière, il faudra généralement augmenter la capacité des mémoires tampons du décodeur pour les données codées ou pour les données de présentation décodées. Le décodage pourra être effectué immédiatement avant la présentation et non pas à l'instant précis qui est indiqué dans l'horloge STC du décodeur; ou bien des unités de présentation décodées pourront être mises en mémoire en vue d'une présentation différée et, éventuellement, répétée. Si le décodage est effectué au moment de la présentation, il faut faire appel à un mécanisme autorisant la suppression des unités de présentation vidéo et audio sans gêner le décodage de données à codage prédictif.

D.0.9 Lissage de la gigue due au réseau

Dans certaines applications, il est parfois possible d'insérer un dispositif entre un réseau et un décodeur afin de diminuer le degré de gigue due au réseau. L'applicabilité d'une telle méthode dépend du type de flux reçus ainsi que de la grandeur et du type de gigue à prévoir.

La syntaxe des flux de transport comme des flux de programme indique le débit d'entrée prévue de ces flux dans un décodeur. Ces valeurs ne sont pas indiquées avec précision et ne peuvent pas être utilisées pour reconstruire exactement le rythme du flux de données. Elles peuvent toutefois être utiles dans le cadre d'un dispositif de lissage.

Par exemple, on peut recevoir du réseau un flux de transport tel que les données arrivent par rafales. Il est possible de mettre les données reçues en mémoire tampon et de les transférer de celle-ci au décodeur à un rythme à peu près constant, tel que le tampon reste à peu près rempli à la moitié de sa capacité.

Il ne convient cependant pas d'acheminer à débit constant un flux émis à débit variable car, avec ce type de flux, la mémoire de lissage ne sera pas toujours à moitié remplie. Un délai moyen constant dans la mémoire tampon nécessite une variation du degré de remplissage en fonction du débit des données. On peut estimer le rythme d'extraction des données dans la mémoire tampon et de leur présentation à l'entrée du décodeur d'après les informations de débit contenues dans le flux de données. Dans les flux de transport, le débit prévu est indiqué par les valeurs des champs de référence PCR et par le nombre d'octets de flux de transport qui les séparent. Dans les flux de programme, le débit prévu est explicitement spécifié par le champ Program_mux_rate bien que, comme spécifié dans la présente Recommandation | Norme internationale, le débit puisse tomber à zéro aux emplacements des références SCR, c'est-à-dire si la référence SCR arrive avant l'instant prévu lorsque les données sont acheminées au débit indiqué.

Dans le cas de flux à débit variable, le remplissage correct de la mémoire de lissage varie dans le temps et ne pourra pas toujours être déterminé exactement d'après les informations relatives au débit. Selon une méthode en variante, on peut utiliser les références SCR ou PCR pour calculer l'instant d'entrée des données dans la mémoire et régler l'instant de leur sortie du tampon. On peut construire une boucle de commande pour garantir un délai moyen constant de passage dans la mémoire tampon. On peut faire observer qu'un tel circuit est semblable à la boucle de commande illustrée par la Figure D.2. On peut obtenir le même résultat que par l'insertion d'un tel dispositif de lissage avant un décodeur en montant en série plusieurs boucles PLL de récupération d'horloge. La réjection de la gigue contenue dans le rythme reçu sera favorisée par l'effet cumulatif des filtres passe-bas contenus dans les circuits PLL en cascade.

Annexe E

Applications de transmission de données

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

E.0 Généralités

- Un multiplex de transport conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 sera utilisé pour transmettre des données, vidéo, audio et autres.
- Les flux élémentaires de données ne sont pas continus comme peuvent le sembler les flux vidéo et audio en télédiffusion.
- Bien qu'il soit possible de déterminer le début d'un paquet de flux PES de données, il n'est pas toujours possible de déterminer la fin d'un tel paquet d'après le début du paquet PES suivant, étant donné qu'un (ou plusieurs) paquet(s) de transport, contenant des paquets PES de données, peuvent être perdus.

E.1 Suggestion

Une solution au problème précédent peut consister à émettre, immédiatement après un paquet PES associé, le paquet PES suivant. S'il n'y a pas de paquet PES à envoyer, on peut envoyer à sa place un paquet PES sans capacité utile.

On trouvera dans le Tableau E.1 un exemple d'un tel paquet de flux PES.

Tableau E.1 – Exemple d'en-tête de paquet de flux PES

Champ des en-têtes de paquet de flux PES	Valeurs
packet_start_code_prefix	0x000001
stream_id	assigned
PES_packet_length	0x0003
'10'	'10'
PES_scrambling_control	'00'
PES_priority	'0'
data_alignment_indicator	'0'
copyright	'0'
original_or_copy	'0'
PTS_DTS_flags	'00'
ESCR_flag	'0'
ES_rate_flag	'0'
DSM_trick_mode_flag	'0'
additional_copy_info_flag	'0'
PES_CRC_flag	'0'
PES_extension_flag	'0'
PES_header_data_length	0x00

Annexe F

Présentation graphique de la syntaxe pour la présente Recommandation | Norme internationale

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

F.0 Introduction

La présente annexe informative présente sous forme graphique la syntaxe des flux de transport et des flux de programme. Elle ne saurait remplacer la section normative précédente.

Afin de rendre les schémas plus clairs, on n'a pas décrit ou représenté entièrement chaque champ. Des champs réservés peuvent être omis ou signalés par des zones hachurées. Les longueurs des champs sont indiquées en octets.

F.0.1 Syntaxe d'un flux de transport

Voir Figure F.1.

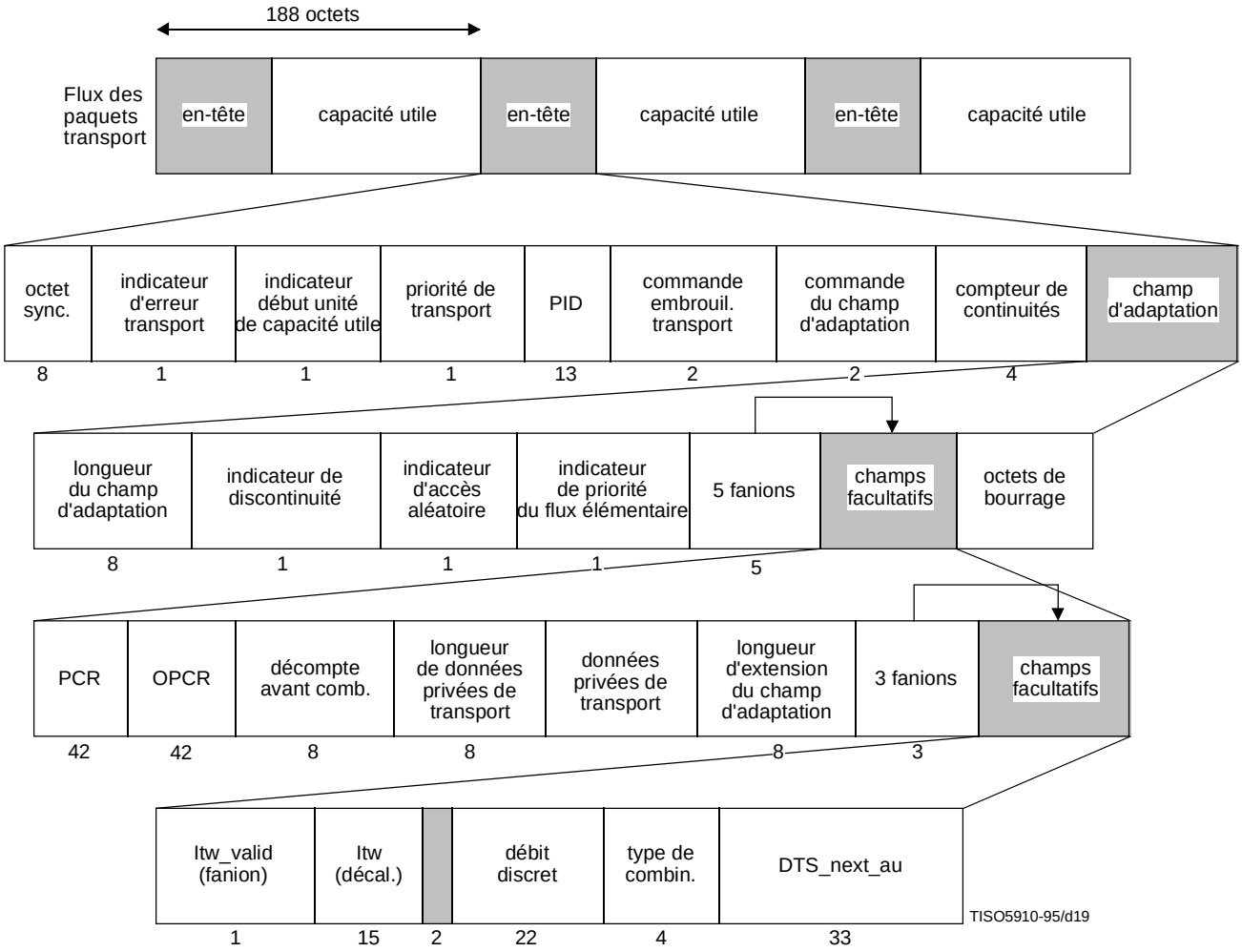


Figure F.1 – Diagramme syntaxique du flux de transport

F.0.2 Paquet de flux PES

Voir Figure F.2.

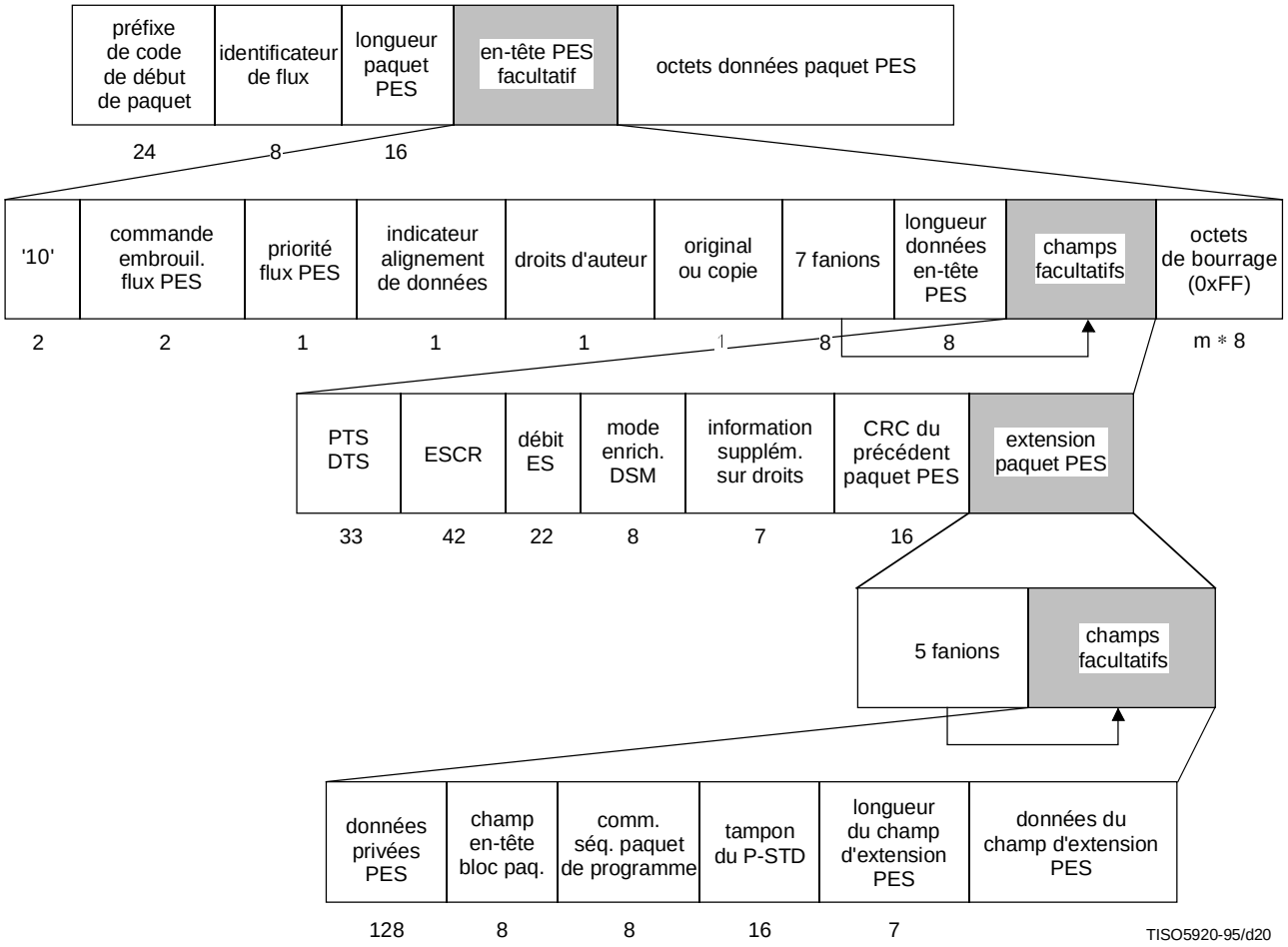


Figure F.2 – Diagramme syntaxique des paquets de flux PES

F.0.3 Section de table d'association de programmes

Voir Figure F.3.

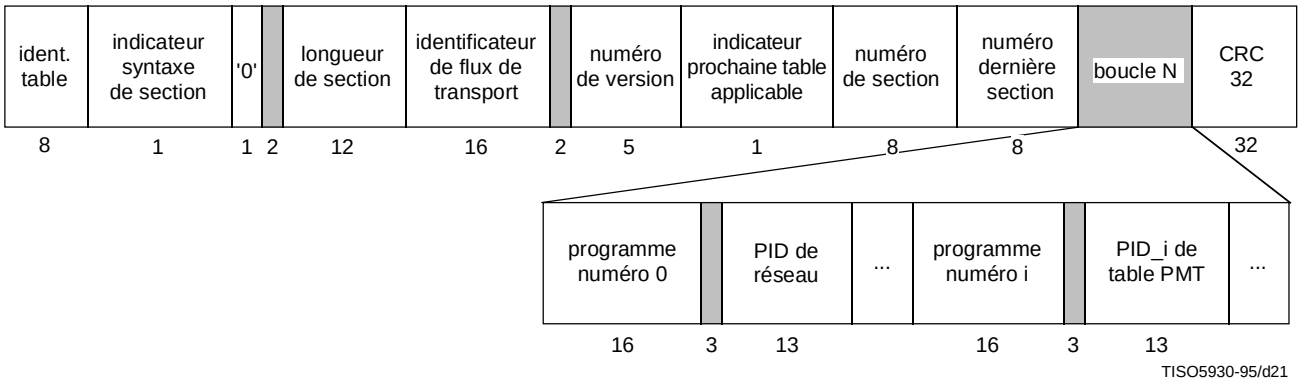


Figure F.3 – Diagramme de section de table d'association de programmes

F.0.4 Section de table d'accès conditionnel

Voir Figure F.4.

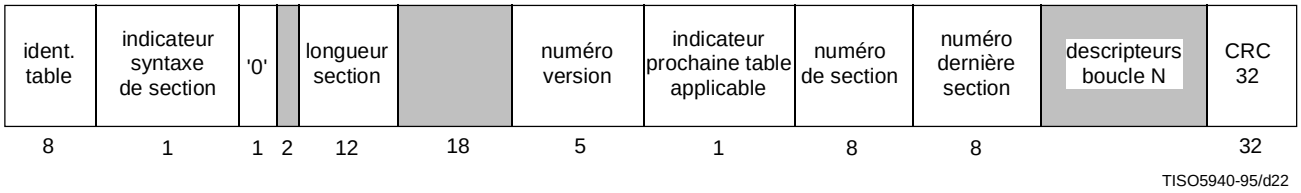


Figure F.4 – Diagramme de section de table d'accès conditionnel

F.0.5 Section de table de mappage de programme pour flux de transport

Voir Figure F.5.

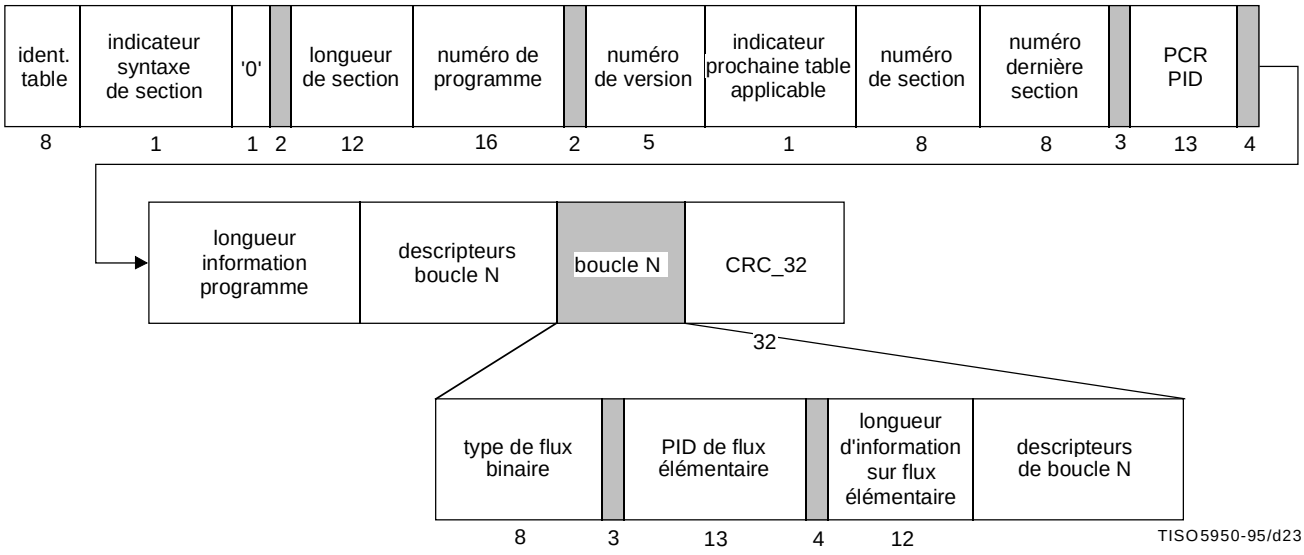


Figure F.5 – Diagramme de section de table de mappage de progamme pour flux de transport

F.0.6 Section de table de données privées

Voir Figure F.6.

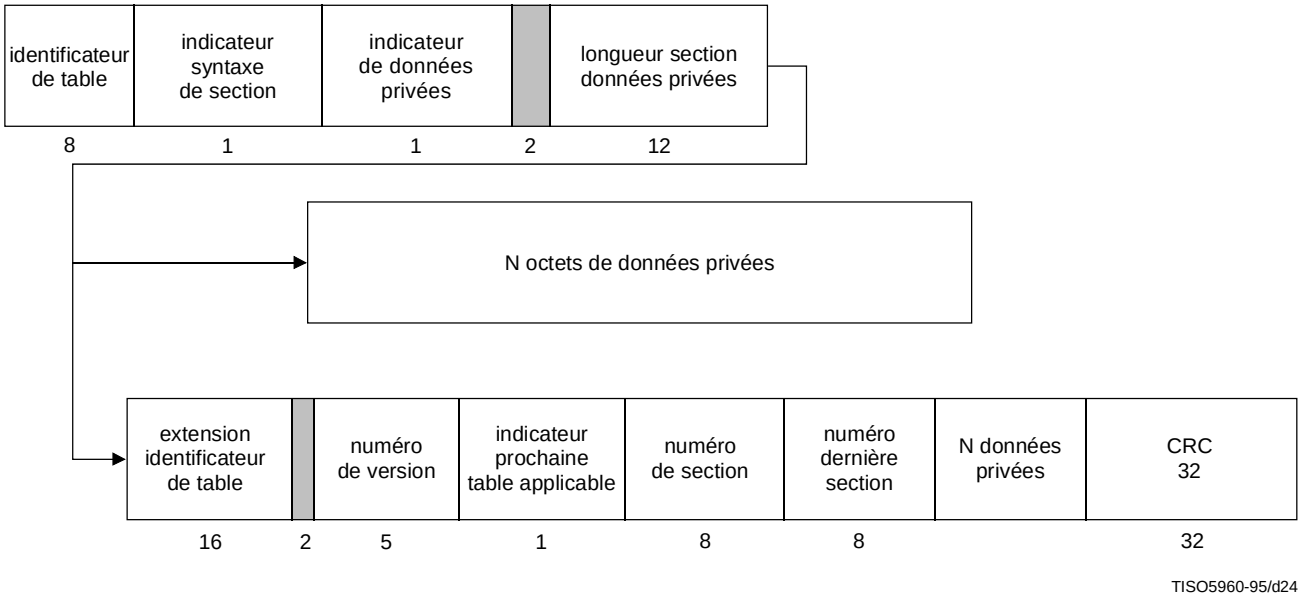


Figure F.6 – Diagramme de section de table de données privées

F.0.7 Flux de programme

Voir Figure F.7.

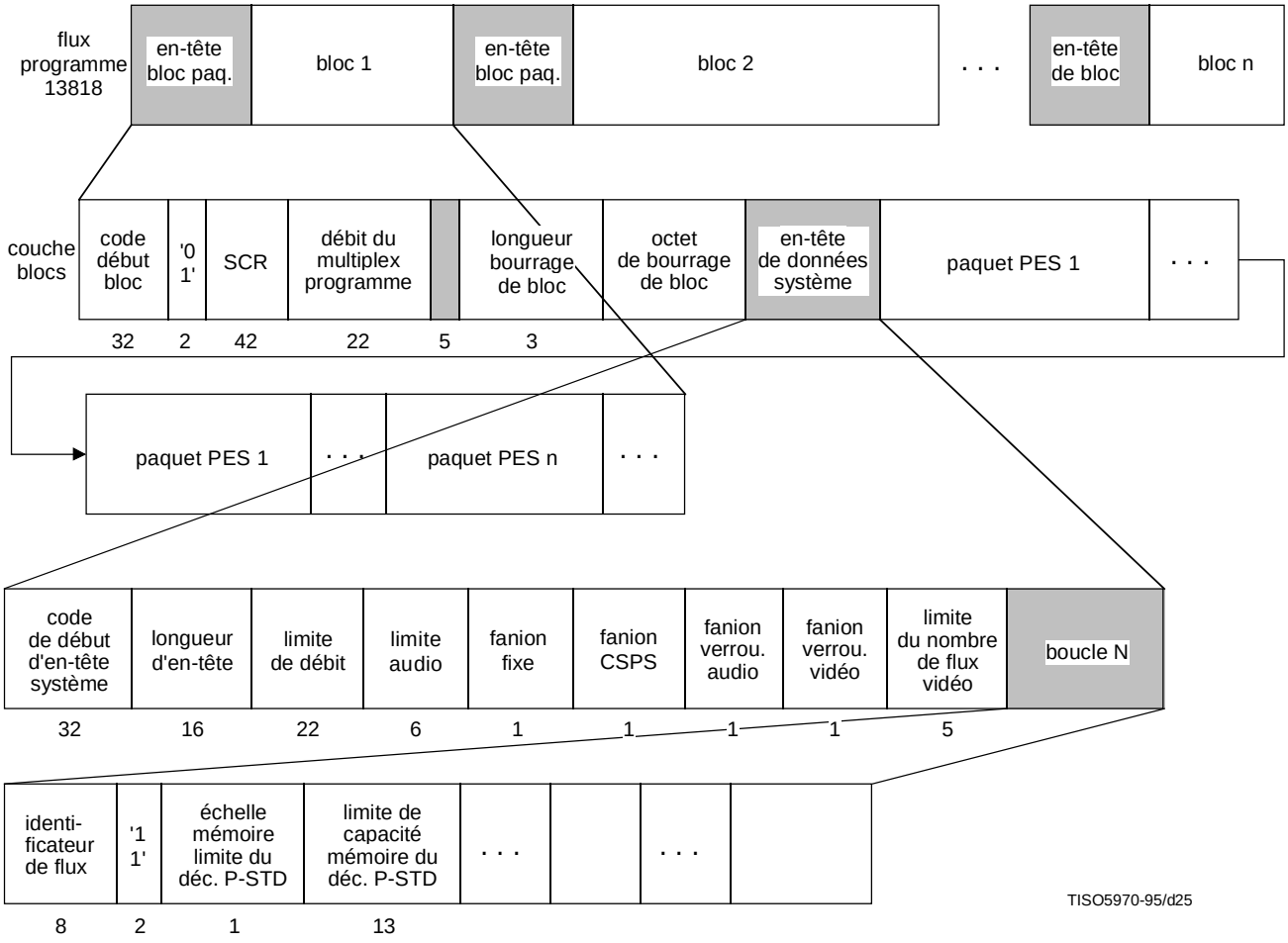


Figure F.7 – Diagramme du flux de programme

F.0.8 Mappage de flux de programme

Voir Figure F.8.

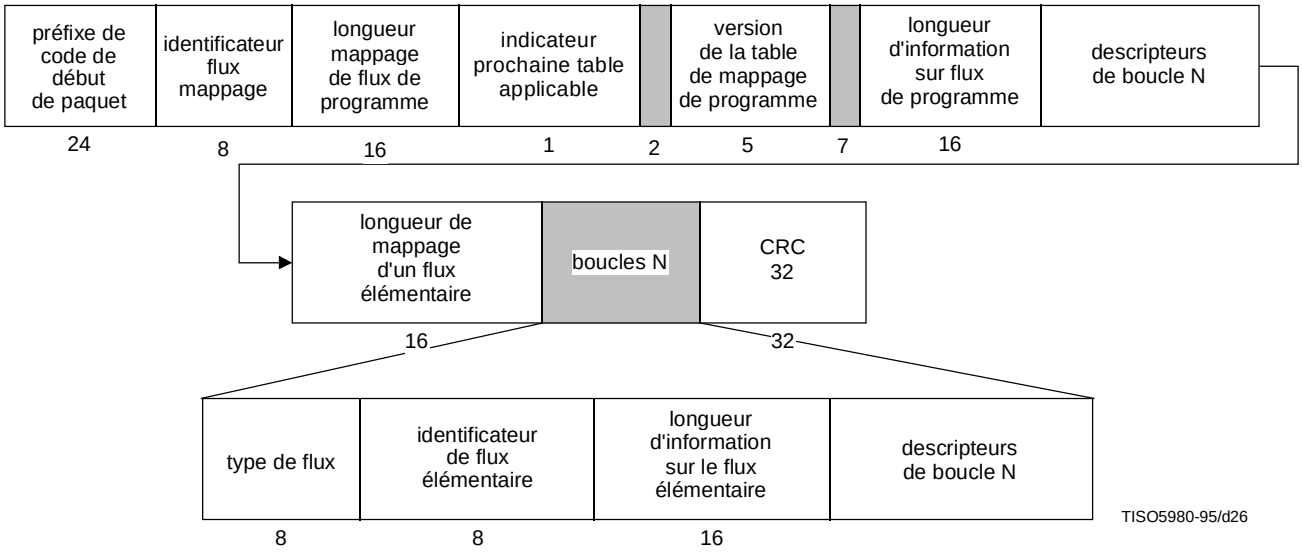


Figure F.8 – Diagramme du mappage de flux de programme

Annexe G

Informations générales

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

G.0 Informations générales

G.0.1 Emulation d'octets de synchronisation

Lors du choix de valeurs d'identificateur PID, il est recommandé d'éviter l'émulation périodique d'octets de synchronisation. Une telle émulation peut se produire à l'intérieur du champ PID ou sous la forme d'une combinaison du champ PID et de valeurs de fanions adjacents. Il est recommandé que l'émulation de l'octet d'émulation ne soit autorisée à se produire à la même position dans l'en-tête de paquet que sur un maximum de 4 paquets de transport consécutifs.

G.0.2 Etat d'image sautée et processus de décodage

Supposons que la séquence à afficher ne contienne que des images bitrames à codage I et à codage P. On désignera la prochaine image à décoder par le champ picture_next et l'image en cours d'affichage par le champ picture_current. Puisque le codeur vidéo peut sauter des macroblocs d'image, il est possible que tous les éléments binaires du champ picture_next ne soient pas présents dans les mémoires tampons EB_n ou B_n du décodeur STD au moment de supprimer ces bits pour un décodage et un affichage instantanés. Si ce cas se produit, aucun bit n'est supprimé de la mémoire tampon du décodeur et l'image désignée par le champ picture_current est de nouveau affichée. Lorsque l'instant d'affichage de l'image suivante arrive, si le reste des bits correspondant au champ picture_next se trouve maintenant dans la mémoire EB_n ou B_n du décodeur STD, tous les bits du champ picture_next sont éliminés et l'image désignée par le champ picture_next est affichée. Si tous les bits du champ picture_next ne sont pas présents dans la mémoire EB_n ou B_n, on répète le processus ci-dessus de réaffichage par picture_current, jusqu'à ce que l'image picture_next puisse être affichée. On notera que si un pointeur PTS précède le champ picture_next dans le flux binaire, ce pointeur est erroné (l'erreur représente un multiple de l'intervalle d'affichage de l'image, lequel peut lui-même dépendre de plusieurs paramètres) et doit être négligé.

Chaque fois que la situation de saut d'image décrite ci-dessus se produit, le codeur est appelé à insérer un pointeur PTS avant l'image à décoder après l'image désignée par picture_next. Cela permet au décodeur de vérifier immédiatement qu'il a correctement affiché la séquence d'images reçue.

G.0.3 Sélection de valeurs d'identificateur PID

Il est recommandé que les applications utilisent de faibles valeurs numériques pour les identificateurs PID (en évitant les valeurs réservées spécifiées dans le Tableau 2-4) et qu'elles regroupent ces valeurs autant que possible.

G.0.4 Emulation d'un code de début de paquet PES

Trois octets consécutifs ayant la valeur d'un préfixe de code de début de paquet (0x000001) peuvent, lorsqu'ils sont concaténés avec un quatrième octet, émuler les quatre octets d'un en-tête de paquet PES à un point non prévu du flux binaire.

Une telle "émulation de code de début" n'est pas possible dans les flux élémentaires de données vidéo mais est possible dans des flux élémentaires de données audio et de données non audiovisuelles. Elle est également possible à la limite d'un en-tête de paquet PES et d'une capacité utile de paquet PES, même si la capacité utile de ce paquet PES est de type vidéo.

Annexe H

Données privées

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

H.0 Données privées

Sont privées toutes les données qui ne sont pas codées conformément à une règle spécifiée par l'UIT-T | ISO/CEI et indiquée dans la présente Spécification. Le contenu de telles données n'est pas et ne doit pas être spécifié dans la présente Recommandation | Norme internationale. Le décodeur STD défini dans la présente Spécification n'est pas utilisable pour les données privées autres que celles liées aux opérations de démultiplexage. Un utilisateur privé peut définir chaque décodeur STD pour l'acheminement des flux privés.

Les données privées peuvent être acheminées par les éléments syntaxiques suivants, décrits dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

1) *Tableau 2-2 sur les paquets de flux de transport*

   Les octets de données du champ transport_packet() peuvent coder des données privées. Les données privées transportées sous cette forme sont désignées comme étant des "données privées d'utilisateur" dans le Tableau 2-34 relatif aux types de flux. Les paquets de flux de transport contenant des données privées peuvent également comporter des champs d'adaptation.

2) *Tableau 2-6 sur les champs d'adaptation de flux de transport*

   La présence d'octets facultatifs de données privées dans le champ d'adaptation est signalée par le fanion transport_private_data_flag. Le nombre d'octets de données privées est implicitement limité par la sémantique du champ adaptation_field_length, dont la valeur ne doit pas dépasser 183 octets.

3) *Tableau 2-21 sur les paquets de flux PES*

   Il existe deux possibilités pour transporter des données privées dans des paquets PES. La première utilise l'en-tête de paquet PES, avec le champ facultatif de 16 octets PES_private_data. La présence de ce champ est signalée par le fanion PES_private_data_flag, dont la présence est elle-même signalée par le fanion PES_extension_flag. Ces octets, s'ils sont présents, ne doivent pas émuler le préfixe de code de début de paquet lorsqu'ils sont pris en compte avec les champs adjacents.

   La deuxième possibilité consiste à utiliser le champ PES_packet_data_bytes. Les données privées transportées sous cette forme sont désignées dans le Tableau 2-34 relatif aux types de flux comme étant des "données de paquet PES". Cette catégorie de données privées peut être subdivisée en deux sous-catégories: le champ private_stream_1 désigne les données privées contenues dans des paquets PES qui suivent la syntaxe de ces paquets, de manière que tous les champs jusqu'au champ PES_header_data_length compris (sans y être limités) soient présents; le champ private_stream_2 désigne les données privées contenues dans des paquets PES où seuls les trois premiers champs doivent être présents, suivis du champ PES_packet_data_bytes contenant les données privées.

   On notera que des paquets PES existent aussi bien dans les flux de programme que dans les flux de transport. Les champs private_stream_1 et private_stream_2 existeront donc dans ces deux types de flux binaire.

4) *Descripteurs*

   Les descripteurs existent dans les flux de programme et dans les flux de transport. L'utilisateur peut définir une liste de descripteurs privés. Ces derniers doivent commencer par les champs descriptor_tag et descriptor_length. Pour les descripteurs privés, le champ descriptor_tag peut prendre une valeur comprise entre 64 et 255, conformément au Tableau 2-45. Ces descripteurs peuvent être placés à l'intérieur d'une table de mappage de flux de programme (Tableau 2-34), d'une section de table d'accès conditionnel (Tableau 2-32), d'une section de table de mappage de programme pour transport (Tableau 2-33) et dans toute section de table de données privées (Tableau 2-35).

   Le descripteur d'accès conditionnel désigne également, de manière spécifique, des octets de données privées.

5) *Section de table de données privées*

Le Tableau 2-35 sur la section de données privées offre un moyen supplémentaire pour acheminer des données privées, de nouveau sous deux formes. Selon le Tableau 2-34 d'affectation des types de flux, ce type de flux élémentaire peut être identifié comme contenant des données privées dans les sections de table d'informations PSI. Un type de section de table de données privées comporte uniquement les cinq premiers champs définis, et précède des données privées. Le fanion section_syntax_indicator doit être mis à '0' pour cette structure; pour l'autre type de structure, le fanion section_syntax_indicator doit être mis à '1' et toute la syntaxe jusqu'au champ last_section_number inclus doit être présente, suivie des octets de données privées et terminée par le contrôle CRC_32.

Annexe I

Conformité des systèmes et interface en temps réel

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

I.0 Conformité des systèmes et interface en temps réel

La conformité des flux de programme et des flux de transport à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 est spécifiée en termes de prescriptions normatives dans le corps de la présente Recommandation | Norme internationale. Ces spécifications comprennent, entre autres caractéristiques fonctionnelles, un décodeur modèle de système (T-STD et P-STD) qui modélise le comportement d'un décodeur théorique à l'entrée duquel un flux binaire est appliqué. Ce modèle et la vérification de conformité associée, ne comportent pas de renseignements sur l'aptitude de flux à l'acheminement en temps réel, sauf pour ce qui est de la précision de la fréquence d'horloge système présentée par le flux de transport et par le flux de programme. Tous les flux, de transport et de programme, doivent être conformes à la présente Recommandation | Norme internationale.

Il existe par ailleurs une spécification d'interface en temps réel pour l'injection dans un décodeur des flux de transport et de programme. La présente Recommandation | Norme internationale autorise la normalisation de l'interface entre décodeurs MPEG et adaptateurs à des réseaux, à des porteuses ou à des supports d'enregistrement. Les effets temporels des porteuses et l'incapacité pratique des adaptateurs à éliminer complètement ces effets entraînent l'apparition d'écarts par rapport aux instants théoriques d'acheminement des octets. Bien que tous les décodeurs MPEG ne soient pas tenus de respecter cette interface, les implémentations mettant en jeu cette dernière doivent être conformes aux spécifications. La présente Recommandation | Norme internationale couvre le comportement d'acheminement en temps réel des flux de transport et de programme aux décodeurs, de manière que les tampons mémorisant dans ces décodeurs des données codées ne subissent ni surremplissage ni sous-remplissage et qu'il soit garanti que les décodeurs soient en mesure d'effectuer la récupération d'horloge avec la qualité de fonctionnement exigée par leurs applications.

L'interface MPEG en temps réel spécifie l'écart maximal admissible par rapport aux instants théoriques d'acheminement des octets qui sont indiqués par les champs de référence temporelle de programme (PCR) et de référence temporelle du système (SCR) codés dans le flux binaire.

Annexe J

Interfaçage entre réseaux générateurs de gigue et décodeurs MPEG-2

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

J.0 Introduction

Dans la présente annexe, le terme "flux système" sera utilisé pour désigner aussi bien les flux de transport Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que les flux de programme Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Lorsque l'abréviation "STD" est utilisée, elle est censée désigner le décodeur P-STD (décodeur modèle du système pour les flux de programme) dans le cas de flux de programme et le décodeur T-STD (décodeur modèle du système pour les flux de transport) dans le cas de flux de transport.

Les instants théoriques d'acheminement des octets d'un flux système peuvent être déduits par analyse de ce flux. Un flux système est considéré comme conforme s'il peut être décodé par le STD, qui est un modèle mathématique représentant un décodeur idéal. Si un flux système conforme est émis sur un réseau générateur de gigue, les instants réels d'acheminement des octets pourront différer notablement des instants théoriques d'acheminement des octets. Dans de tels cas, il n'est pas toujours possible de décoder le flux système dans un tel décodeur idéal, car la gigue peut provoquer des surremplissages ou des sous-remplissages de mémoire tampon et donc gêner la récupération de la base de temps. Un important exemple d'un tel réseau générateur de gigue est le mode de transmission ATM.

La présente annexe a pour objet de donner des indications et des précisions aux entités chargées de faire transiter des flux systèmes sur des réseaux générateurs de gigue. Il est probable que l'on mettra au point, pour plusieurs types de réseau tel l'ATM, des modèles de conformité à chaque réseau transportant des flux systèmes. Le modèle STD, assorti d'une définition de l'interface en temps réel, peut jouer un rôle déterminant afin de mettre au point de tels modèles. Le paragraphe J.1 présente un cadre général pour la mise au point de modèles de conformité de réseau.

Le paragraphe J.2 développe trois exemples de codage en couche Réseau permettant de mettre en œuvre des adaptateurs à lissage de gigue réseau. Dans le premier exemple, on suppose un flux de données systèmes à débit constant et un ordonnancement de type premier entré-premier sorti pour le lissage de la gigue. Dans le deuxième exemple, la couche d'adaptation réseau comporte des pointeurs temporels facilitant le lissage de gigue. Dans l'exemple final, une horloge réseau commune est supposée être disponible de bout en bout, dont les signaux sont utilisés pour réaliser le lissage de gigue.

Le paragraphe J.3 présente deux exemples de réalisation de décodeur, dans lesquels la gigue due au réseau peut être prise en charge. Dans le premier exemple, on insère un adaptateur de réseau à lissage de gigue entre une terminaison de réseau et un décodeur MPEG-2. On suppose que le décodeur MPEG-2 est conforme à une spécification d'interface en temps réel avec un dispositif MPEG-2. Cette interface exige un décodeur MPEG-2 dont la tolérance à la gigue est supérieure à celle du décodeur théorique STD. L'adaptateur de réseau traite la gigue du flux binaire entrant et produit un flux système dont les instants réels d'acheminement d'octets sont conformes à la spécification d'interface en temps réel. Le premier exemple fait l'objet du J.3.1. Pour certaines applications, la méthode de l'adaptateur réseau sera trop coûteuse parce qu'elle nécessite deux étapes de traitement. Dans le deuxième exemple, les fonctions de suppression de gigue et de décodage MPEG-2 sont intégrées. Le traitement intermédiaire par le dispositif suppresseur de gigue est laissé de côté, si bien qu'une seule étape de récupération d'horloge est nécessaire. Les décodeurs qui effectuent de manière intégrée la suppression de gigue et le décodage sont désignés dans la présente annexe par le terme de "décodeurs intégrés d'adaptation au réseau" ou simplement "décodeurs intégrés". Ces décodeurs sont décrits au J.3.2.

Afin de construire soit des adaptateurs réseau ou des décodeurs intégrés, il faut partir de l'hypothèse d'une valeur maximale pour la gigue réseau de crête à crête. Afin de favoriser l'interopérabilité, il faut spécifier une limite de gigue de crête à crête pour chaque type de réseau considéré.

J.1 Modèle de conformité de réseau

La Figure J.1 présente une manière de modéliser la transmission d'un flux système par un réseau générateur de gigue.

Le flux système est injecté dans un dispositif de codage particulier au réseau, qui convertit les données systèmes en format spécifique du réseau. Cette structure peut comporter des informations destinées à faciliter la suppression de la gigue à la terminaison du réseau. Le décodeur du réseau comporte un décodeur propre au réseau et un décodeur Rec. UIT-T 222.0 | ISO/CEI 13818-1. Celui-ci est censé être conforme à une spécification d'interface en temps réel et peut avoir la même architecture que le décodeur STD, avec des mémoires tampons appropriées, de capacité augmentée afin d'offrir une plus grande tolérance à la gigue. Le décodeur propre au réseau supprime les données de format non Rec. UIT-T H.222.0 | ISO/CEI 13818-1 qui sont ajoutées par le codeur propre au réseau puis supprime la gigue à la terminaison du réseau. La sortie du décodeur propre au réseau est un flux système conforme à la spécification en temps réel.

On peut définir un décodeur modèle de réseau (NTD, *network target decoder*), fondé sur l'architecture ci-dessus. Un flux binaire de réseau conforme sera tel qu'il puisse être décodé par le NTD. Un décodeur propre au réseau sera conforme s'il peut décoder tout flux binaire de réseau susceptible d'être décodé par le NTD. Un décodeur propre au réseau pourra avoir en réalité, ou ne pas avoir, l'architecture du décodeur NTD.

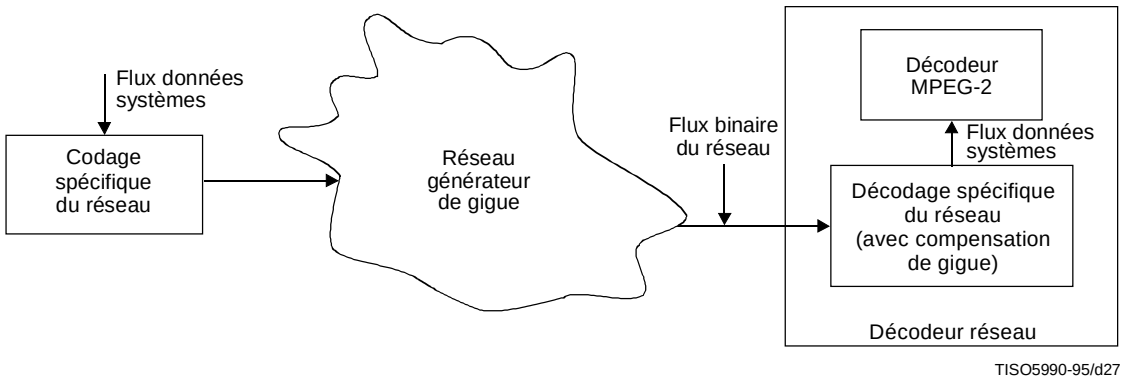


Figure J.1 – Envoi de flux systèmes sur un réseau générateur de gigue

J.2 Spécification de réseau pour le lissage de gigue

Dans le cas de flux systèmes à débit constant, le lissage de gigue pourra être effectué au moyen d'un dispositif du type premier entré-premier sorti. Dans la couche d'adaptation au réseau, il n'est pas nécessaire d'ajouter des données fournissant une aide spécifique pour la suppression de gigue. Une fois que les octets ajoutés par le codage propre au réseau ont été éliminés, les données du flux système sont placées dans une file de type premier entré-premier sorti (FIFO). Un circuit de verrouillage PLL maintient le tampon à peu près à la moitié de sa capacité en réglant le débit de sortie en fonction des variations de remplissage de la mémoire. Dans cet exemple, le degré de lissage de gigue obtenu dépendra de la capacité de la file FIFO et des caractéristiques du circuit PLL.

La Figure J.2 illustre une deuxième façon de réaliser le lissage de gigue. Dans cet exemple, on part du principe que la couche d'adaptation du réseau fournit des pointeurs temporels. Cette technique permet de lisser la gigue de flux systèmes à débit binaire aussi bien constant que variable.

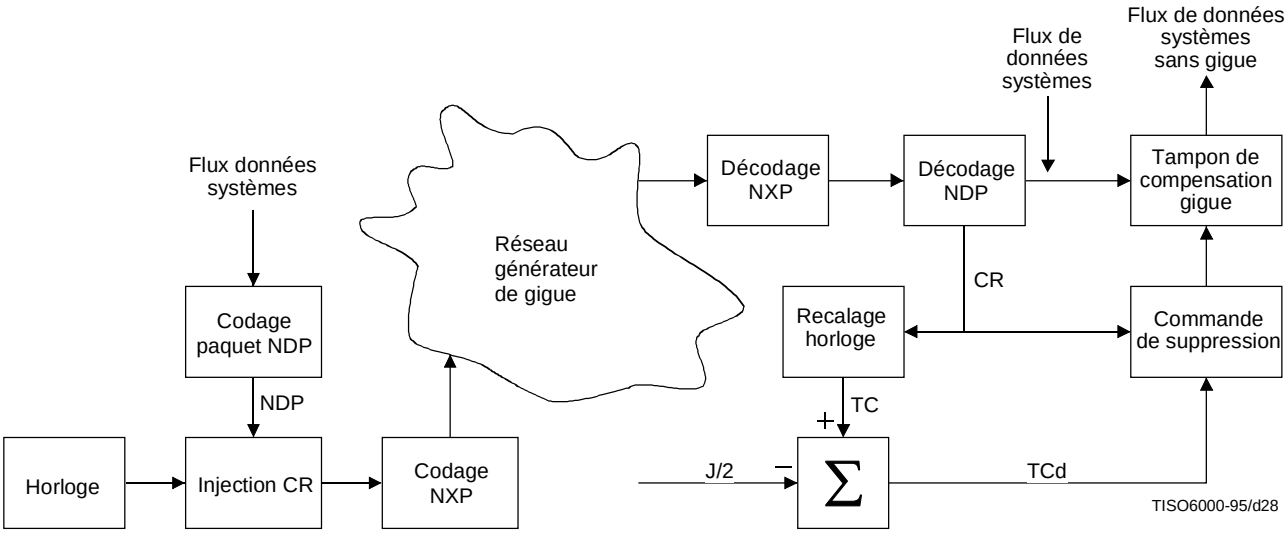


Figure J.2 – Lissage de gigue au moyen de pointeurs temporels issus de la couche Réseau

Supposons que l'adaptateur réseau soit conçu de façon à compenser une gigue de J secondes de crête à crête. Les instants théoriques d'acheminement des octets seront reconstitués au moyen des références temporelles (CR, *clock reference*) échantillonnées sur une horloge de base (TC, *time clock*). Les références temporelles et l'horloge de base sont des éléments analogues aux références PCR et STC. Le convertisseur de paquets de données réseau (NDP, *network data*

packet) convertit chaque paquet du flux système en paquet de données réseau (NDP). Les paquets NDP contiennent un champ acheminant les valeurs de référence temporelle et on y insère la valeur actuelle de l'horloge de base, au moment où le paquet NDP quitte le codeur correspondant. La fonction d'empaquetage pour transport par réseau (NXP, *network transport packetization*) encapsule les paquets NDP dans des paquets de transport par le réseau. Après transmission à l'autre bout du réseau, les références temporelles reconstruisent l'horloge de base, par exemple au moyen d'un circuit PLL. Le premier paquet MPEG-2 est extrait de la mémoire de lissage de gigue lorsque la valeur retardée d'horloge de base (TCd) est égale à la première référence temporelle de ce premier paquet MPEG-2. Les paquets MPEG-2 suivants seront extraits lorsque leur valeur de référence temporelle sera égale à celle de l'élément TCd.

Compte non tenu de détails d'implémentation tels que la vitesse de traitement par la boucle de récupération d'horloge de base (TC) et la pureté spectrale du signal d'horloge de base, la capacité de la mémoire de lissage de gigue ne dépend que de la valeur maximale de la gigue de crête à crête à lisser et du plus grand débit de transport intervenant dans le flux système. Cette capacité de mémoire de lissage de gigue, B_{dj}, est donnée par la relation suivante:

$$B_{dj} = JR_{max}$$

où R_{max} est le débit maximal du flux système en bits par seconde. Lorsque les paquets acheminés par le réseau subissent le délai nominal, la mémoire est à moitié remplie. Lorsqu'ils subissent un délai de J/2 secondes, la mémoire est vide et lorsqu'ils subissent un délai (en avance) de –J/2 secondes, la mémoire est pleine.

A titre d'exemple final, une horloge réseau commune sera parfois disponible de bout en bout et il sera alors possible de caler la fréquence d'horloge système sur celle de l'horloge commune. L'adaptateur réseau pourra lisser la gigue avec une file FIFO. L'adaptateur utilisera les références PCR ou SCR pour reconstituer les instants originaux d'acheminement des octets.

J.3 Exemples d'implémentations de décodeur

J.3.1 Adaptateur de réseau suivi d'un décodeur MPEG-2

Dans ce type d'implémentation, un adaptateur de réseau conforme à la spécification de conformité du réseau est connecté à un décodeur MPEG-2 conforme à la spécification d'interface en temps réel.

J.3.2 Décodeur intégré

L'exemple présenté au J.3.1 nécessite deux étapes de traitement du signal. La première est nécessaire pour lisser la gigue issue du réseau; la deuxième, qui reconstitue l'horloge STC par traitement des références PCR ou SCR, est nécessaire pour le décodage STD. L'exemple présenté dans le présent paragraphe est celui d'un décodeur qui intègre les fonctions de lissage de gigue et de décodage dans un seul système. L'horloge STC est reconstituée directement au moyen des valeurs des références PCR ou SCR affectées de gigue. Pour présenter cet exemple, on supposera un flux de transport de type MPEG-2.

La Figure J.3 illustre le fonctionnement de ce décodeur intégré. On suppose que le flux de paquets issus du réseau qui est injecté dans le décodeur est le même que celui de la Figure J.2.

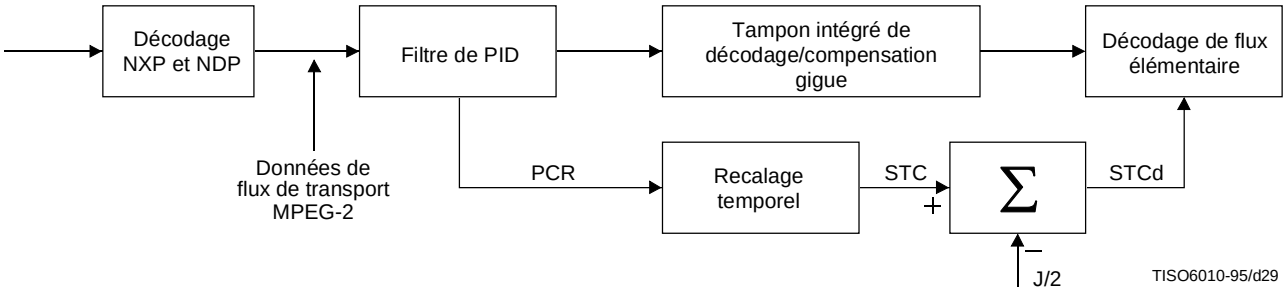


Figure J.3 – Fonctions intégrées de lissage de gigue et de décodage MPEG-2

Les paquets issus du réseau qui entrent dans le décodeur sont réassemblés en données de flux de transport MPEG-2 par les fonctions de décodage NXP et NDP. Les paquets de flux de transport de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 affectés de gigue sont ensuite filtrés pour en extraire les paquets possédant l'identificateur PID souhaité. Pour le cas illustré, l'identificateur PID en cours de décodage transporte également les références PCR. Celles-ci sont appliquées à un circuit PLL afin de reconstituer l'horloge STC. Les paquets entiers destinés à l'identificateur PID choisi sont placés

dans la mémoire du décodeur intégré. Une valeur positive de J/2 secondes est soustraite de l'horloge STC afin d'obtenir la valeur retardée d'horloge STC, STCd. De nouveau, J représente la gigue de crête à crête que le décodeur propre au réseau peut tolérer. Le retard est introduit pour garantir que toutes les données requises pour une unité d'accès seront arrivées dans la mémoire tampon au moment où le pointeur PTS/DTS de cette unité d'accès sera égal à la valeur actuelle de l'horloge STCd.

Compte non tenu de détails d'implémentation tels que la vitesse de la boucle de récupération de l'horloge STC et la pureté spectrale de ce signal, la capacité de la mémoire tampon du décodeur intégré, B_{size}, sera déterminée par la relation suivante:

$$\begin{aligned} B_{size} &= B_{dec} + B_{mux} + B_{OH} + 512 + B_j \\ &= B_n + 512 + B_j \end{aligned}$$

où $B_j = R_{max}\, J$, et R_{max} est le débit maximal d'entrée des données dans le filtre PID. Selon l'implémentation, la mémoire du décodeur intégré pourra être subdivisée en deux parties, comme dans le décodeur T-STD.

Annexe K

Combinaison des flux de transport

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

K.0 Introduction

Aux fins de la présente annexe, le terme "combinaison" désigne l'opération de concaténation effectuée sur le niveau transport de deux flux élémentaires différents, le flux de transport ainsi obtenu étant parfaitement conforme à la présente Recommandation | Norme internationale. Les deux flux élémentaires peuvent avoir été produits à deux emplacements distincts et à deux moments différents et n'ont pas été nécessairement destinés à être combinés lorsqu'ils ont été produits. On appellera ci-après "ancien" flux un flux élémentaire continu (audio ou vidéo), qui a été remplacé par un autre flux (le flux dit "nouveau"), à partir d'un certain instant. Cet instant est appelé "point de combinaison": il s'agit de la limite entre les données qui font partie de l'"ancien" flux et les données du "nouveau" flux.

Une combinaison peut être transparente ou non transparente:

- une combinaison transparente est une combinaison qui n'entraîne aucune discontinuité de décodage (voir 2.7.6). Autrement dit l'instant de décodage de la première unité d'accès du "nouveau" flux est "cohérent" par rapport à l'instant de décodage de l'unité d'accès de l'"ancien" flux, précédant le point de combinaison; il se confond par conséquent avec l'instant de décodage qui aurait été celui de l'unité d'accès suivante si l'"ancien" flux avait continué. Ci-dessous cet instant de décodage sera appelé "instant de décodage transparent";
- une combinaison non transparente est une combinaison qui entraîne l'apparition d'une discontinuité de décodage; autrement dit l'instant de décodage de la première unité d'accès du "nouveau" flux est postérieur à l'instant de décodage transparent.

NOTE – Un instant de décodage antérieur à l'instant de décodage transparent est interdit.

Une combinaison réalisée en une limite de paquet quelconque d'un flux de transport est autorisée, si le flux ainsi obtenu est légal. Toutefois, en règle générale, si l'on ne dispose d'aucune indication quant à l'emplacement des débuts de paquets PES et des débuts des unités d'accès, cette contrainte impose de faire l'analyse, non seulement de la couche de transport, mais aussi de la couche PES et de la couche du flux élémentaire et rend parfois indispensables certaines opérations sur la capacité utile des paquets du flux de transport. Si l'on souhaite éviter la réalisation d'opérations complexes de ce type, il convient de réaliser la combinaison aux emplacements caractérisés par des propriétés favorables du flux de transport, lesquelles propriétés sont indiquées par la présence d'un point de combinaison.

La présence d'un point de combinaison est indiquée par les champs de fanion de combinaison et de décompte avant combinaison (la sémantique de ces champs est décrite au 2.4.3.4). Ci-dessous le paquet de flux de transport dans lequel le champ splice_countdown prend la valeur '0' sera appelé "paquet de combinaison". Le point de combinaison est situé immédiatement après le dernier octet du paquet de combinaison.

K.1 Les différents types de points de combinaison

Un point de combinaison peut être un point de combinaison ordinaire ou un point de combinaison transparente.

K.1.1 Points de combinaison ordinaire

En l'absence du champ de fanion de combinaison transparente ou si la valeur de ce dernier est mise à zéro, il s'agit alors d'un point de combinaison ordinaire. La présence d'un point de combinaison ordinaire signale simplement les propriétés d'alignement du flux de transport; le paquet de combinaison se termine au niveau du dernier octet d'une unité d'accès et la capacité utile du paquet suivant du flux de transport de même identificateur PID commencera par l'en-tête d'un paquet PES dont la capacité utile commencera par un point d'accès de flux élémentaire [ou par un code de fin de séquence () suivi immédiatement d'un point d'accès de flux élémentaire, dans le cas de données vidéo]. Ces propriétés facilitent la réalisation d'opérations "couper"/"coller" sur le niveau transport, tout en respectant les contraintes syntaxiques et en garantissant la cohérence du flux binaire. Elles ne donnent cependant aucune indication quant aux propriétés de synchronisation ou de mise en mémoire tampon. Par conséquent, en cas d'utilisation de points de combinaison de ce type, la transparence des combinaisons réalisées exige des agencements particuliers de données privées ou une analyse de la capacité utile des paquets du flux de transport, ainsi que la surveillance de l'état des mémoires tampons et des valeurs des champs horodateurs.

K.1.2 Points de combinaison transparente

En présence du champ de fanion de combinaison transparente et si la valeur de ce dernier est mise à '1', le point de combinaison donne des indications quant à certaines propriétés de l'"ancien" flux. Les informations en question ne sont pas destinées aux décodeurs. Elles servent avant tout à faciliter la réalisation d'une combinaison transparente. Un point de combinaison de ce type est appelé point de combinaison transparente. Les informations fournies sont les suivantes:

- l'instant de décodage transparent, codé sous forme de valeur DTS dans le champ DTS_next_AU. Cette valeur DTS est exprimée dans la base de temps valide à l'intérieur du paquet de combinaison;
- dans le cas d'un flux élémentaire vidéo, les contraintes qui ont été appliquées à l'"ancien" flux, lorsqu'il a été produit pour faciliter la réalisation d'une combinaison transparente. Ces indications sont fournies par la valeur du champ splice_type dans le tableau correspondant au profil et au niveau du flux vidéo.

On notera qu'un point de combinaison transparente peut servir de point de combinaison ordinaire, si l'on ignore ces informations supplémentaires. Celles-ci peuvent néanmoins être utilisées par ailleurs, si l'on considère qu'elles peuvent servir à réaliser une combinaison non transparente ou à d'autres fins.

K.2 Comportement des décodeurs en présence de combinaisons

K.2.1 Combinaisons non transparentes

Comme indiqué plus haut, une combinaison non transparente est une combinaison qui entraîne l'apparition d'une discontinuité de décodage.

Il faut signaler que la présence d'une combinaison de ce type exige l'observation des contraintes liées à la discontinuité de décodage (voir 2.7.6), en particulier:

- une valeur PTS doit être codée pour la première unité d'accès du "nouveau" flux (sauf en période d'activation du mode enrichissement ou lorsque le fanion low_delay est mis à la valeur '1');
- l'instant de décodage calculé à partir de cette valeur PTS (ou à partir de la valeur DTS associée) ne doit pas précéder l'instant de décodage transparent;
- dans le cas d'un flux élémentaire vidéo, si le paquet de combinaison ne se termine pas par un code de fin de séquence (), le "nouveau" flux doit commencer par un code de fin de séquence () suivi immédiatement d'un en-tête de séquence ().

Théoriquement, et du fait qu'elles introduisent des discontinuités de décodage, les combinaisons de ce type se traduisent par la présentation non continue des unités de présentation (c'est-à-dire avec un délai de longueur variable entre l'affichage de deux images consécutives ou entre deux trames audio consécutives). Dans la pratique, l'effet observé dépendra des conditions d'implémentation du décodeur, en particulier pour les flux élémentaires vidéo. Dans le cas de certains décodeurs de données vidéo, l'arrêt sur une ou plusieurs images constitue parfois la solution retenue de préférence. Voir partie 4 de l'ISO/CEI 13818.

K.2.2 Combinaisons transparentes

L'élimination des discontinuités de décodage permet d'éliminer des discontinuités de présentation. Dans le cas des flux élémentaires audio cela est toujours possible. Par contre, il y a lieu de noter qu'en ce qui concerne les flux vidéo, la continuité de présentation n'est théoriquement possible que dans les cas 1) et 2) décrits ci-dessous:

1) l'"ancien" flux se termine par la fin d'une séquence low-delay, et le "nouveau" flux commence par le début d'une séquence dite "non-low-delay";
2) l'"ancien" flux se termine à la fin d'une séquence "non-low-delay", et le "nouveau" flux commence au début d'une séquence "low-delay".

Les effets induits par des situations de ce type sont tributaires des conditions d'implémentation. Par exemple, dans le cas 1), une image doit parfois être présentée pendant deux périodes de trame, tandis que dans le cas 2) il peut s'avérer nécessaire de sauter une image. Il est néanmoins techniquement possible que certaines implémentations autorisent des situations de ce type sans que des effets indésirables en résultent.

En outre, compte tenu du 6.1.1.6 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, un code de fin de séquence () doit précéder le premier en-tête de séquence du "nouveau" flux, si au moins un paramètre de séquence (c'est-à-dire un paramètre défini dans l'en-tête de séquence ou dans une extension d'en-tête de séquence) ne prend pas la même valeur dans les deux flux, excepté les paramètres de définition de la matrice de quantification. Par exemple, si le champ débit ne prend pas la même valeur dans le "nouveau" flux et dans l'"ancien" flux, la présence d'un code de fin de séquence () est obligatoire. Par conséquent, si le paquet de combinaison ne se termine pas par un code de fin de séquence, le "nouveau" flux doit commencer par un code de fin de séquence, suivi d'un en-tête de séquence.

En vertu du paragraphe précédent, la présence d'un code de fin de séquence sera obligatoire dans la plupart des combinaisons, même les combinaisons transparentes. Il y a lieu de noter que la Rec. UIT-T H.262 | ISO/CEI 13818-2 spécifie le processus de décodage des séquences vidéo [c'est-à-dire des données comprises entre un en-tête de séquence () et un code de fin de séquence ()], et ne spécifie rien quant à la façon de réaliser un changement de séquence. En ce qui concerne le comportement des décodeurs en présence de combinaisons de ce type, il y a lieu de consulter la partie 4 de l'ISO/CEI 13818.

K.2.3 Surremplissage de la mémoire tampon

Même si les deux flux élémentaires sont conformes au modèle T-STD avant d'être combinés, il n'est pas forcément certain qu'il n'y aura pas de surremplissage des mémoires tampons STD en présence du flux combiné, pendant la durée de l'intervalle de temps au cours duquel les bits des deux flux se trouvent dans ces mémoires.

Dans le cas d'un débit vidéo constant, si l'"ancien" flux n'a été soumis à aucune condition particulière, et si aucune précaution spéciale n'a été prise pendant la combinaison, ce surremplissage est possible lorsque le débit vidéo du "nouveau" flux est supérieur à celui de l'"ancien". De fait, il est certainement exact qu'il n'y aura pas de surremplissage des mémoires tampons MB_n et EB_n du décodeur T-STD, si les bits sont injectés dans le décodeur T-STD au débit "ancien". Par contre, si le débit d'injection passe à une valeur plus élevée à l'entrée de la mémoire TB_n, avant que les bits "anciens" soient complètement extraits du décodeur T-STD, le niveau de remplissage des mémoires tampons STD deviendra plus important que si l'"ancien" flux avait continué, sans combinaison, et risque alors de provoquer le surremplissage des mémoires EB_n et MB_n. Dans le cas d'un débit vidéo variable, le même problème risque de se poser, si le débit d'injection du "nouveau" flux dépasse le débit pour lequel des dispositions ont été prises lors de la création de l'"ancien" flux. Une telle situation est proscrite.

Toutefois, le dispositif d'encodage qui crée l'"ancien" flux peut imposer des conditions supplémentaires à la gestion du vérificateur de mémoire vidéo VBV au voisinage des points de combinaison, de telle sorte que des dispositions appropriées soient prises dans l'éventualité d'un "nouveau" débit vidéo inférieur à une valeur déterminée. Par exemple, dans le cas d'un point de combinaison transparente, des conditions supplémentaires de ce type peuvent être indiquées par le choix d'une valeur 'splice_type' à laquelle correspondent des valeurs attribuées dans les Tableaux 2-7 à 2-20 aux paramètres 'splice_decoding_delay' et 'max_splice_rate'. Si le débit vidéo du "nouveau" flux est inférieur à la valeur 'max_splice_rate' ainsi déterminée, il est alors certain que le flux combiné ne provoquera pas de surremplissage pendant le laps de temps au cours duquel les éléments binaires des deux flux se trouvent dans la mémoire du décodeur T-STD.

Lorsque aucune des contraintes de ce type n'a été imposée, il est possible d'éviter l'apparition de ce problème en adoptant un délai d'introduction des bits entre l'"ancien" flux et le "nouveau", afin de permettre aux mémoires tampons du décodeur T-STD de se vider suffisamment avant l'introduction des bits du "nouveau" flux. Si l'on appelle t_{in} l'instant auquel le dernier octet de la dernière unité d'accès de l'"ancien" flux est introduit dans le décodeur STD, et t_{out} l'instant auquel il en sort, il suffit de veiller à ce que, pendant l'intervalle de temps $[t_{in}, t_{out}]$, le flux combiné ne donne pas lieu à l'introduction d'un plus grand nombre de bits que si l'"ancien" flux avait continué, sans combinaison de flux. A titre d'exemple, lorsque l'"ancien" flux se caractérise par un débit constant R_{old}, et le "nouveau" par un débit constant R_{new}, il suffit d'introduire un délai T_d vérifiant les relations ci-dessous pour éviter ce risque de surremplissage:

$$T_d \geq 0 \text{ et } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old}/R_{new})$$

Annexe L

Procédure d'enregistrement (voir 2.9)

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

L.1 Procédure de demande d'un identificateur enregistré (RID, *registered identifier*)

Les demandeurs d'identificateur RID doivent s'adresser à l'organisme d'enregistrement, qui leur remettra des formulaires d'enregistrement. Les renseignements que le demandeur doit fournir sont indiqués en L.3. Les sociétés et organisations sont autorisées à déposer une demande.

L.2 Responsabilités de l'organisme d'enregistrement

Les principales responsabilités de l'organisme d'enregistrement, qui administre l'enregistrement des identificateurs de droits d'auteur, sont décrites dans le présent paragraphe. Les Directives du JTC 1 contiennent quelques autres responsabilités. L'organisme d'enregistrement doit:

a) implémenter une procédure d'enregistrement à appliquer à un identificateur RID unique, conformément à l'Annexe H des Directives du JTC 1;

b) recevoir et traiter les demandes d'attribution d'identificateur de code de type d'œuvre, reçues de l'organisme d'enregistrement des droits d'auteur;

c) vérifier que les demandes reçues sont conformes à la présente procédure d'enregistrement et indiquer au demandeur, dans les 30 jours de la réception de sa demande, l'identificateur RID qui lui a été attribué;

d) informer les fournisseurs d'applications, dont la demande est rejetée par écrit dans les 30 jours de la réception de la demande et informer le demandeur des processus d'appel;

e) tenir à jour un registre exact des identificateurs RID attribués. Les révisions apportées aux renseignements de contact et aux spécifications techniques doivent être acceptées et conservées par l'organisme d'enregistrement;

f) communiquer le contenu de ce registre sur demande de tout tiers intéressé;

g) conserver une base de données contenant les formulaires des demandes d'identificateur RID octroyées et refusées. Les parties recherchant des renseignements techniques sur le format de données privées faisant l'objet d'un identificateur de droits d'auteur doivent avoir accès aux informations faisant partie de la base de données conservée par l'organisme d'enregistrement;

h) rendre compte annuellement de ses activités au JTC 1, à l'ITTF et au Secrétariat du JTC 1/SC 29 ou à leurs représentants respectifs, selon un calendrier mutuellement convenu.

L.2.1 Renseignements de contact de l'organisme d'enregistrement

Nom de l'organisation:

Adresse:

Téléphone:

Télécopie:

L.3 Responsabilités des parties demandant un identificateur RID

La partie demandant un identificateur RID en vue de l'identification de droits d'auteur doit:

a) déposer sa demande au moyen du formulaire et des procédures fournis par l'organisme d'enregistrement;

b) fournir des renseignements de contact décrivant comment l'on peut obtenir, à titre non discriminatoire, une description complète de l'organisation de protection des droits d'auteur;

c) donner des détails techniques sur la syntaxe et la sémantique du format de données utilisé dans le champ **additional_copyright_info** pour décrire les œuvres audiovisuelles ou d'autres travaux protégés par droits d'auteur. Une fois enregistrée, la syntaxe utilisée pour l'information supplémentaire sur les droits d'auteur ne doit pas être modifiée;

d) convenir d'instituer dans un délai raisonnable l'usage prévu de l'identificateur RID octroyé;

e) conserver une trace permanente du formulaire de demande et de la notification reçue de l'organisme d'enregistrement pour chaque identificateur de droits d'auteur octroyé.

L.4 Procédure d'appel en cas de refus de demande

Le comité de gestion pour l'enregistrement (RMG, *registration management group*) a été formé afin de pouvoir être saisi d'appels relatifs aux refus de demande d'identificateur RID. Le comité RMG doit être composé d'experts désignés par les membres P et L du comité technique ISO chargé de la présente Recommandation | Norme internationale. Ce comité doit avoir un rapporteur et un secrétariat, désignés par ses membres. L'organisme d'enregistrement est habilité à désigner un seul membre observateur, sans droit de vote.

Les responsabilités du comité RMG sont les suivantes:

a) examiner et juger tous les appels dans un délai raisonnable;

b) informer, par écrit, les organisations qui font appel en vue d'un réexamen de leur position au sujet du jugement du RMG en la matière;

c) examiner le rapport annuel résumant les activités de l'organisme d'enregistrement;

d) fournir aux comités Membres ISO des renseignements sur le domaine d'activités de l'organisme d'enregistrement.

Annexe M

Formulaire de demande d'enregistrement (voir 2.9)

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

M.1 Renseignements de contact sur l'organisation demandant un identificateur enregistré (RID)

Nom de l'organisation:

Adresse:

Téléphone:

Télécopie:

Courrier électronique:

M.2 Déclaration d'intention d'appliquer l'identificateur RID assigné

Domaine d'application de l'identificateur RID: selon les directives fournies par l'organisme d'enregistrement.

M.3 Date d'implémentation prévue de l'identificateur RID

M.4 Représentant autorisé

Nom:

Titre:

Adresse:

Signature: _____

M.5 Cadre réservé à l'usage officiel de l'organisme d'enregistrement

Enregistrement refusé: _____

Motif du refus de la demande:

Enregistrement octroyé: _____ Valeur d'enregistrement: _____

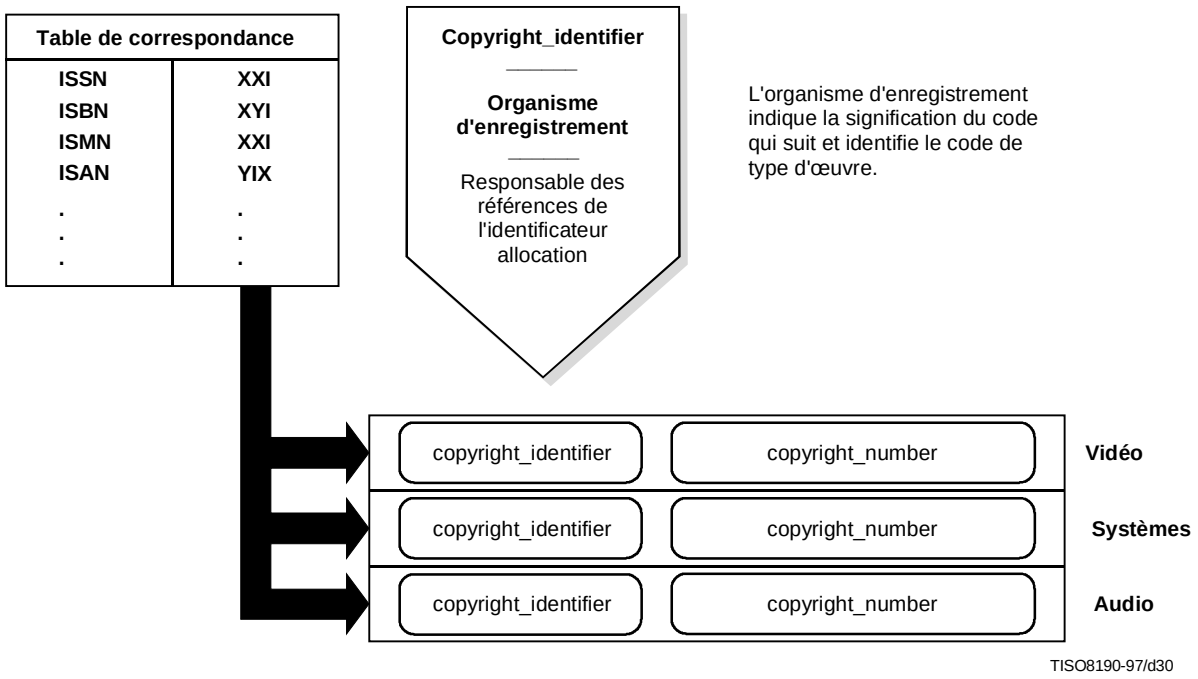
Pièce jointe 1 – Détails techniques sur le format de données enregistré.

Pièce jointe 2 – Notification de la procédure d'appel en cas de refus d'une demande.

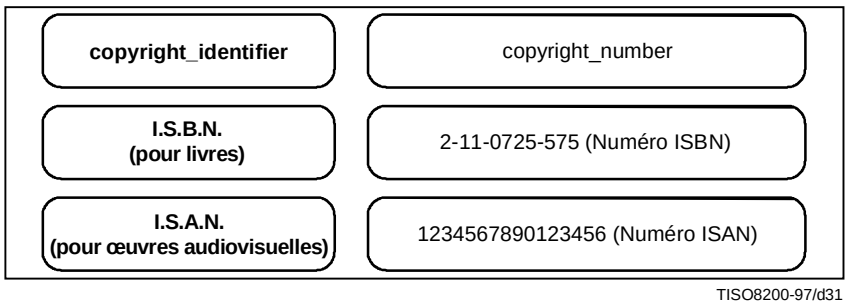
Annexe N

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Organisme d'enregistrement
Diagramme de structure administrative (voir 2.9)



Exemples



Tous les identificateurs de droits d'auteur sont enregistrés par l'organisme d'enregistrement, à titre unique pour les numéros de droits d'auteur normalisés par l'ISO. Chaque organisation qui attribue des numéros de droits d'auteur demande à l'organisme d'enregistrement un identificateur de droits d'auteur spécifique, par exemple la Bibliothèque d'Etat des Biens culturels prussiens, désignée par l'ISO pour gérer les numéros I.S.B.N., demande à l'organisme d'enregistrement un identificateur spécifique pour les livres.

Annexe O

Procédure d'enregistrement (voir 2.10)

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

O.1 Procédure de demande d'un identificateur enregistré (RID)

Les demandeurs d'identificateur RID doivent s'adresser à l'organisme d'enregistrement, qui leur remettra des formulaires d'enregistrement. Les renseignements que le demandeur doit fournir sont indiqués dans O.4. Les sociétés et organisations sont autorisées à déposer une demande.

O.2 Responsabilités de l'organisme d'enregistrement

Les principales responsabilités de l'organisme d'enregistrement, qui administre l'enregistrement des identificateurs de format de données privées, sont décrites dans cette annexe. Les Directives du JTC 1 contiennent quelques autres responsabilités. L'organisme d'enregistrement doit:

a) implémenter une procédure d'enregistrement à appliquer à un identificateur RID unique, conformément aux Directives du JTC 1;

b) recevoir des fournisseurs d'applications les demandes d'attribution d'identificateur et les traiter;

c) vérifier que les demandes reçues sont conformes à la présente procédure d'enregistrement et indiquer au demandeur, dans les 30 jours de la réception de sa demande, l'identificateur RID qui lui a été attribué;

d) informer les fournisseurs d'applications dont la demande est rejetée par écrit dans les 30 jours de la réception de la demande et examiner en temps utile les resoumissions de la demande;

e) tenir à jour un registre exact des identificateurs RID attribués. Les révisions apportées aux spécifications de format doivent être acceptées et conservées par l'Organisme d'enregistrement;

f) communiquer le contenu de ce registre, sur demande, aux comités Membres de l'ISO ou de la CEI participant au JTC 1, aux organisations avec lesquelles l'ISO ou la CEI sont en liaison et à toute partie intéressée;

g) conserver une base de données contenant les formulaires des demandes d'identificateur RID octroyées et refusées. Les parties recherchant des renseignements techniques sur le format de données privées faisant l'objet d'un identificateur RID doivent avoir accès aux informations faisant partie de la base de données conservée par l'organisme d'enregistrement;

h) rendre compte annuellement de ses activités au JTC 1, à l'ITTF et au Secrétariat du JTC 1/SC 29 ou à leurs représentants respectifs;

i) prévoir d'utiliser, chaque fois que possible, les identificateurs RID existants.

O.3 Renseignements de contact de l'organisme d'enregistrement

O.4 Responsabilités des parties demandant un identificateur RID

La partie demandant un identificateur de format doit:

a) déposer sa demande au moyen du formulaire et des procédures fournis par l'organisme d'enregistrement;

b) donner une description de l'objet du flux binaire enregistré ainsi que les détails techniques nécessaires, comme spécifié dans le formulaire de demande;

c) fournir des renseignements de contact décrivant comment l'on peut obtenir, à titre non discriminatoire, une description complète;

d) convenir d'instituer dans un délai raisonnable l'usage prévu de l'identificateur RID octroyé;

e) conserver une trace permanente du formulaire de demande et de la notification reçue de l'organisme d'enregistrement pour chaque identificateur de format de données privées octroyé.

O.5 Procédure d'appel en cas de refus de demande

Le comité de gestion pour l'enregistrement (RMG) a été formé afin de pouvoir être saisi d'appels relatifs aux refus de demande d'identificateur RID. Le comité RMG doit être composé d'experts désignés par les membres P et L du comité

technique ISO chargé de la présente Spécification. Ce comité doit avoir un rapporteur et un secrétariat, désignés par ses membres. L'organisme d'enregistrement est habilité à désigner un seul membre observateur, sans droit de vote.

Les responsabilités du comité RMG sont les suivantes:

a) examiner et juger tous les appels dans un délai raisonnable;

b) informer, par écrit, les organisations qui font appel en vue d'un réexamen de leur position au sujet du jugement du RMG en la matière;

c) examiner le rapport annuel résumant les activités de l'organisme d'enregistrement;

d) fournir aux organismes Membres de l'ISO et aux comités nationaux de la CEI des renseignements sur le domaine d'activités de l'organisme d'enregistrement.

Annexe P

Formulaire de demande d'enregistrement

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

P.1 Renseignements de contact sur l'organisation demandant un identificateur enregistré (RID)

Nom de l'organisation:

Adresse:

Téléphone:

Télécopie:

Courrier électronique:

Télex:

P.2 Demande d'identificateur RID spécifique

NOTE – Si le système a déjà été implémenté et est en cours d'utilisation, remplir cette rubrique et la rubrique P.3 puis passer à P.6; sinon, laisser cet espace en blanc et passer à P.4.

P.3 Brève description de l'identificateur RID qui est en usage et date de l'implémentation du système

P.4 Déclaration d'intention d'appliquer l'identificateur RID assigné

P.5 Date d'implémentation prévue de l'identificateur RID

P.6 Représentant autorisé

Nom:

Titre:

Adresse:

Signature: _____

P.7 Cadre réservé à l'usage officiel de l'organisme d'enregistrement

Enregistrement refusé:_____

Motif du refus de la demande:

Enregistrement octroyé: _____ Valeur d'enregistrement: _____

Pièce jointe 1 – Détails techniques sur le format de données enregistré.

Pièce jointe 2 – Notification de la procédure d'appel en cas de refus d'une demande.

Annexe Q

Modèles de mémoires tampons T-STD et P-STD pour données audio ADTS ISO/CEI 13818-7

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Q.1 Introduction

Le décodeur modèle du système pour le flux de transport applicable aux flux audio est défini au 2.4.2. On trouvera dans la présente annexe le modèle de mémoire tampon de l'ADTS de l'ISO/CEI 13818-7.

Dans un multiplex conforme à UIT-T H.222.0 | ISO/CEI 13818-1, on reconnaît les flux audio ADTS ISO/CEI 13818-7 à la présence de l'identificateur de flux stream_id=0x110yyyyy ["y" = "don't care" (ne pas tenir compte)] et du type de flux stream_type=0x0F définis dans les Tableaux 2-22 et 2-23.

Q.2 Débit de débordement de la mémoire tampon de transport

Pour les données audio, à l'exception des données audio ADTS ISO/CEI 13818-7, le débit de débordement de la mémoire tampon de transport est de 2 Mbit/s. Ce débit est toutefois inférieur au débit maximal relatif aux données audio ADTS ISO/CEI 13818-7. Pour le flux de données audio ADTS ISO/CEI 13818-7, on attribue donc au débit de débordement une valeur différente de celle des flux de données audio ISO/CEI 11172-3 et ISO/CEI 13818-3.

Le flux élémentaire ADTS ISO/CEI 13818-7 comprend un ou plusieurs canaux. Le débit maximal de chaque canal est de 576 kbit/s pour une fréquence d'échantillonnage de 96 kHz. Pour calculer le débit de débordement à retenir pour les données audio ADTS ISO/CEI 13818-7, on applique donc la formule suivante:

$$Rx_n = 1{,}2 \times R_{max} \times N \text{ bit/s}$$

où:

R_{max} est un débit constant à 576 kbit/s tel que défini au 3.2.2 de l'ISO/CEI 13818-7. Il s'agit d'une limite supérieure du débit par canal du flux ADTS AAC correspondant à la valeur maximale de la fréquence d'échantillonnage (c'est-à-dire Fs = 96 kHz);

et où:

N est le nombre de canaux audio qui doivent être assortis de leurs propres mémoires tampons dans le décodeur dans ce flux élémentaire (flux de canaux individuels dans un même élément de canal ou dans un élément de paires de canaux et dans des éléments de canaux de couplage commutés indépendamment).

Q.3 Capacité de mémoire tampon

Pour les données audio, à l'exception des données ADTS ISO/CEI 13818-7, la capacité de la mémoire tampon principale est de 3584 octets. Cette capacité est toutefois inférieure à la capacité maximale de la mémoire tampon d'entrée dans le décodeur pour les données audio ADTS ISO/CEI 13818-7. Pour la capacité de la mémoire tampon principale relative au flux de données audio ADTS ISO/CEI 13818-7, on attribue donc une valeur différente de celle des flux de données audio ISO/CEI 11172-3 et ISO/CEI 13818-3.

La capacité de la mémoire tampon principale pour les données audio ADTS ISO/CEI 13818-7 est calculée comme suit:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh}$$

où BS_{oh}, tampon de surdébit de paquets, est défini par:

$$BS_{oh} = 528 \text{ octets}$$

et BS_{mux}, tampon de multiplexage supplémentaire, est défini par:

$$BS_{mux} = 0{,}004 \text{ seconde} \times R_{max} \times N$$

et BS_{dec}, tampon d'unité d'accès, est défini par:

$$BS_{dec} = 6144 \text{ bits} \times N$$

où:

R_{max} est un débit constant à 576 kbit/s tel que défini au 3.2.2 de l'ISO/CEI 13818-7. Il s'agit d'une limite supérieure du débit par canal du flux ADTS AAC correspondant à la valeur maximale de la fréquence d'échantillonnage (c'est-à-dire F_s = 96 kHz);

et où:

N est le nombre de canaux audio qui doivent être assortis de leurs propres mémoires tampons dans le décodeur dans ce flux élémentaire (flux de canaux individuels dans un même élément de canal ou dans un élément de paires de canaux et dans des éléments de canaux de couplage commutés indépendamment).

Q.3.1 TBS_n: même valeur que pour les autres données audio

En ce qui concerne la mémoire tampon de lissage, la capacité de la mémoire TB_n ne diffère pas entre les flux audio ADTS ISO/CEI 13818-7 et les autres flux audio. Il n'est donc pas nécessaire de modifier la valeur de TBS_n, qui correspond à la capacité de TB_n.

Q.3.2 BS_{mux}: valeur différente des autres données audio

Il convient de modifier la capacité du tampon de multiplexage supplémentaire BS_{mux} afin de l'adapter à des gigues de retard pouvant atteindre 4 ms. Cette approche est analogue à celle retenue pour d'autres flux dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Q.3.3 BS_{dec}: valeur différente des autres données audio

Le tampon d'unité d'accès BS_{dec} est fondé sur la capacité de la mémoire tampon d'entrée dans le décodeur du flux élémentaire. Comme indiqué au 3.2.2 de l'ISO/CEI 13818-7, la capacité totale des mémoires tampons d'entrée de décodeur est de 6144 bits multipliée par le nombre de canaux qui doivent être assortis de leurs mémoires tampons d'entrée respectives dans le décodeur.

Q.3.4 BS_{oh}: valeur différente des autres données audio

BS_{oh} correspond au surdébit de l'en-tête de paquet PES.

Comme indiqué au 2.4.2.6,

> *le temps de passage des données dans les mémoires tampons de décodeurs modèles de systèmes doit demeurer inférieur ou égal à 1 seconde, sauf en ce qui concerne les données d'images vidéo fixes.*

Comme indiqué en outre au 2.7.4,

> *le flux de programme et le flux de transport doivent être construits de manière que l'intervalle de temps maximal entre les horodateurs de présentation codés pour chaque flux élémentaire de données vidéo ou audio soit inférieur ou égal à 0,7 s.*

Il convient de fixer la capacité de BS_{oh} à une valeur appropriée correspondant au surdébit de l'en-tête de paquet PES lorsque le flux AAC est empaqueté selon les règles ci-dessus. La taille maximale de l'en-tête de paquet PES est de 264 octets. Par conséquent, la valeur BS_{oh} = 528 octets, c'est-à-dire deux fois la taille maximale de l'en-tête de paquet PES, garantit qu'au moins deux en-têtes de paquet PES peuvent entrer dans la mémoire tampon principale quelle que soit la taille de l'en-tête du paquet PES. Autrement dit, l'en-tête de paquet PES avec horodateur de présentation (PTS) peut être inséré à des intervalles inférieurs à 0,7 seconde même lorsque les données d'une seconde se trouvent dans la mémoire tampon principale.

Exemple: fréquence d'échantillonnage de 48 kHz

La taille de l'en-tête de paquet PES en l'absence de tout champ facultatif, à l'exception de l'horodateur PTS, est de 18 octets. Les unités d'accès d'une seconde sont au nombre de 47. Lorsque les données d'une seconde sont dans la mémoire tampon principale (cas le plus défavorable), on peut faire correspondre le surdébit de l'en-tête de paquet PES au surdébit BS_{oh} en empaquetant deux unités d'accès ou plus dans un paquet.

number_of_AU = 48 kHz/1024 = 46,875 secondes

(number_of_AU/2) × 18[octet] = 421,875 octets < BS_{oh}

On peut faire correspondre des en-têtes de paquet PES plus fréquents au surdébit BS_{oh}, si le temps de passage des données dans la mémoire tampon principale est inférieur à une seconde.

Q.4 Conclusion

Le modèle de mémoire tampon dans le décodeur devrait s'appliquer à la capacité maximale de la mémoire tampon, bien que le codage AAC puisse assurer jusqu'à 48 canaux et un débit très élevé. Par conséquent, les 3 niveaux de nombre de canaux, 2, 8 et 48, sont utilisés pour définir le débit de débordement et la capacité de la mémoire tampon principale. En cas d'utilisation de deux canaux, on utilise comme valeurs conventionnelles les mêmes valeurs de débit de débordement et de capacité de la mémoire tampon principale pour assurer la compatibilité. Dans les autres cas (8 et 48 canaux), les formules proposées sont appliquées.

Débit de débordement T-STD pour les données audio ADTS ISO/CEI 13818-7

Nombre de canaux	Rx_n [bit/s]
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Canaux: il s'agit du nombre de canaux de sortie audio pleine bande plus le nombre d'éléments de canaux de couplage commutés indépendamment dans le même flux audio élémentaire. Par exemple, dans le cas type où il n'y a pas d'élément de canaux de couplage commutés indépendamment, le son monophonique correspond à 1 canal, le son stéréophonique à 2 canaux et le son ambiophonique 5.1 à 5 canaux (le canal pour les effets à basses fréquences (LFE) n'est pas compté).

Capacité de la mémoire tampon principale T-STD pour les données audio ADTS ISO/CEI 13818-7

Nombre de canaux	BS_n [octets]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Canaux: il s'agit du nombre de canaux de sortie audio pleine bande plus le nombre d'éléments de canaux de couplage commutés indépendamment dans le même flux audio élémentaire. Par exemple, dans le cas type où il n'y a pas d'élément de canaux de couplage commutés indépendamment, le son monophonique correspond à 1 canal, le son stéréophonique à 2 canaux et le son ambiophonique 5.1 à 5 canaux (le canal pour les effets à basses fréquences (LFE) n'est pas compté).

Pour le flux de programme, la capacité de la mémoire tampon principale susmentionnée devrait être choisie dans la P-STD_buffer_scale et dans la P-STD_buffer_size de la façon suivante:

Nombre de canaux	P-STD_buffer_scale	P-STD_buffer_size
1-2	0	28
3-8	0	71
9-48	0	401

Annexe R

Acheminement de scènes ISO/CEI 14496 dans un flux selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

R.1 Procédure d'accès au contenu pour des éléments de programme ISO/CEI 14496 dans un flux de programme

On trouvera ci-dessous une procédure d'acquisition de récepteur de référence pour accéder à des éléments de programme ISO/CEI 14496 contenus dans un flux de programme selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. On suppose ici que le flux de programme inclut un mappage de flux de programme (acheminé dans un paquet PES ayant un identificateur de flux stream_id égal à 0xBC):

- acquisition du mappage de flux de programme;
- identification du descripteur IOD dans la première boucle de descripteur;
- identification des ES_ID du descripteur d'objet, de la description de scène et d'autres flux décrits dans le descripteur d'objet initial;
- acquisition du descripteur SL et du descripteur FMC dans la deuxième boucle de descripteur pour les identificateurs de flux élémentaire 0xFA et 0xFB, si applicable;
- production, sur la base de ces descripteurs, d'une table de mappage de flux entre les identificateurs ES_ID et l'identificateur de flux élémentaire associé plus la voie FlexMux, si applicable;
- localisation du flux de descripteur d'objet au moyen de son identificateur ES_ID et de la table de mappage de flux;
- localisation d'autres flux décrits dans le descripteur d'objet initial au moyen de leurs identificateurs ES_ID et de la table de mappage de flux;
- surveillance en continu du flux de descripteur d'objet et identification des ES_ID de flux additionnels;
- localisation des flux additionnels au moyen de leur identificateur ES_ID et de la table de mappage de flux.

La Figure R.1 donne un exemple de contenu ISO/CEI 14496 dans un flux de programme composé d'un flux descripteur d'objet, d'un flux descripteur de scène (commande BIFS), d'un flux BIFS-Anim et d'un flux IPMP. Tous les flux ISO/CEI 14496 sont multiplexés en un unique flux FlexMux.

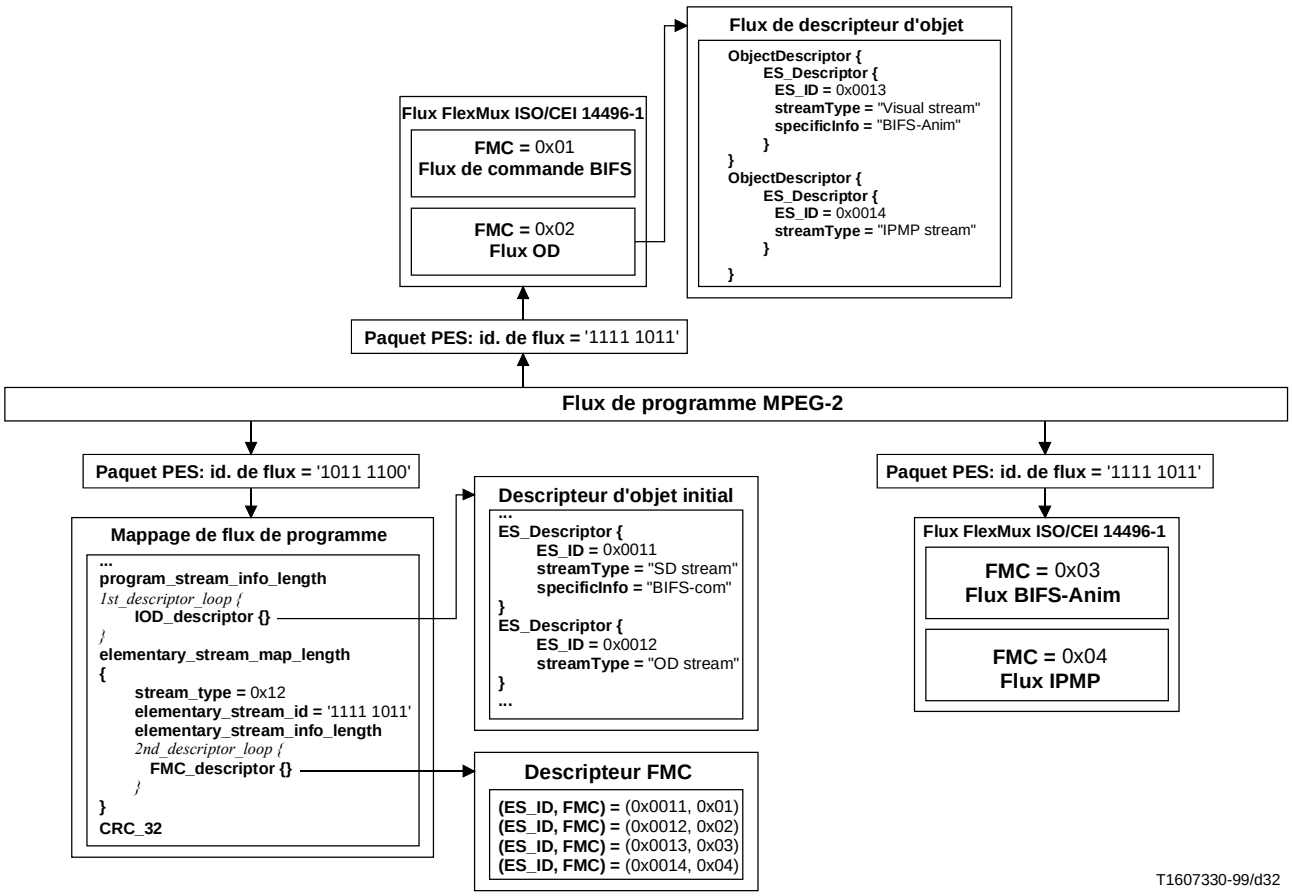


Figure R.1 – Exemple de contenu ISO/CEI 14496 dans un flux de programme

R.2 Procédure d'accès au contenu pour des éléments de programme ISO/CEI 14496 dans un flux de transport

On trouvera ci-dessous une procédure d'acquisition de récepteur de référence pour accéder à des éléments de programme ISO/CEI 14496 contenus dans un flux de transport selon la Rec. UIT-T H.222.0 | ISO/CEI 13818-1:

- acquisition de la table de mappage de programme pour le programme désiré;
- identification du descripteur IOD dans la première boucle de descripteur;
- identification des ES_ID du descripteur d'objet, de la description de scène et d'autres flux décrits dans le descripteur d'objet initial;
- acquisition de l'ensemble de tous les descripteurs SL et FMC présents dans la deuxième boucle de descripteur pour l'un quelconque des identificateurs de flux élémentaire;
- production, sur la base de ces descripteurs, d'une table de mappage de flux entre les identificateurs ES_ID et l'identificateur de flux élémentaire associé plus la voie FlexMux, si applicable;
- localisation du flux de descripteur d'objet au moyen de son identificateur ES_ID et de la table de mappage de flux;
- localisation d'autres flux décrits dans le descripteur d'objet initial au moyen de leurs identificateurs ES_ID et de la table de mappage de flux;
- surveillance en continu du flux de descripteur d'objet et identification des ES_ID de flux additionnels;
- localisation des flux additionnels au moyen de leur identificateur ES_ID et de la table de mappage de flux.

La Figure R.2 donne un exemple d'éléments de programme ISO/CEI 14496 dans un flux de transport composé d'un flux descripteur d'objet, d'un flux descripteur de scène (commande BIFS), d'un flux BIFS-Anim et d'un flux IPMP. Les flux de commande BIFS et OD sont acheminés au moyen de sections ISO/CEI 14496, tandis que les flux élémentaires BIFS-Anim et IPMP sont acheminés dans des paquets de flux PES référencés par deux valeurs distinctes d'identificateur elementary_PID, sans utilisation de l'outil FlexMux ISO/CEI 14496-1.

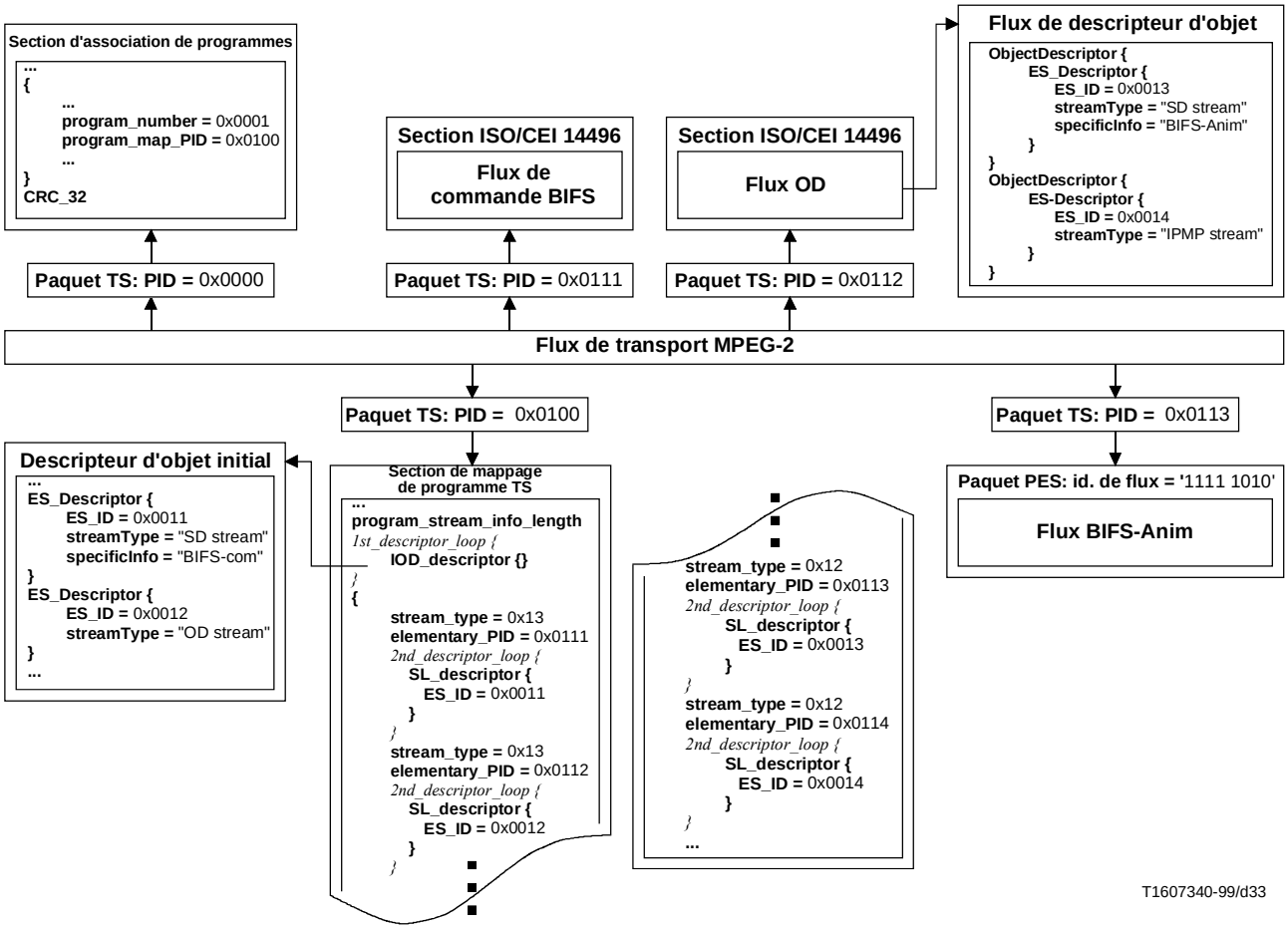


Figure R.2 – Exemple de contenu ISO/CEI 14496 dans un flux de transport

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	**Systèmes audiovisuels et multimédias**
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Gestion des télécommunications y compris le RGT et maintenance des réseaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Imprimé en Suisse
Genève, 2008