



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

H.222.0

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(07/95)

**TRANSMISSION DE SIGNAUX
NON TÉLÉPHONIQUES**

**TECHNOLOGIES DE L'INFORMATION –
CODAGE GÉNÉRIQUE DES IMAGES
ANIMÉES ET DU SON ASSOCIÉ: SYSTÈMES**

Recommandation UIT-T H.222.0

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT) (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie 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. Le texte de la Recommandation H.222.0 de l'UIT-T a été approuvé le 10 juillet 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 13818-1.

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.

© UIT 1996

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
Résumé	vi
Introduction	vi
Intro. 1 Flux de transport	vii
Intro. 2 Flux de programme	ix
Intro. 3 Conversion entre flux de transport et flux de programme	x
Intro. 4 Flux élémentaire paqueté.....	xi
Intro. 5 Modèle de gestion du temps	xi
Intro. 6 Accès conditionnel	xi
Intro. 7 Opérations à l'échelle du multiplex	xi
Intro. 8 Opérations à l'échelle des flux individuels	xii
Intro. 8.1 Démultiplexage	xii
Intro. 8.2 Synchronisation	xii
Intro. 8.3 Relation avec la couche de compression	xiii
Intro. 9 Décodeur de référence du système	xiii
Intro. 10 Applications	xiii
SECTION 1 – CONSIDÉRATIONS GÉNÉRALES	1
1.1 Domaine d'application.....	1
1.2 Références normatives	1
1.3 Recommandations Normes internationales identiques.....	1
1.4 Autres références	2
SECTION 2 – ÉLÉMENTS TECHNIQUES	2
2.1 Définitions.....	2
2.2 Symboles et abréviations	5
2.2.1 Opérateurs arithmétiques	5
2.2.2 Opérateurs logiques	6
2.2.3 Opérateurs relationnels	6
2.2.4 Opérateurs au niveau des bits.....	6
2.2.5 Affectation	6
2.2.6 Mnémoniques.....	6
2.2.7 Constantes	7
2.3 Méthode de description de la syntaxe des flux binaires.....	7
2.4 Prescriptions applicables au flux binaire de transport.....	8
2.4.1 Structure de codage et paramètres du flux de transport	8
2.4.2 Décodeur modèle du système pour le flux de transport	9
2.4.2.1 Fréquence de l'horloge système	10
2.4.2.2 Entrée dans le décodeur modèle du système pour le flux de transport	11
2.4.2.3 Mise en mémoire tampon.....	12
2.4.2.4 Décodage	17
2.4.2.5 Présentation.....	17
2.4.2.6 Gestion de la mémoire tampon	17

	<i>Page</i>	
2.4.3	Spécification de la syntaxe et de la sémantique du flux de transport.....	18
2.4.3.1	Flux de transport	18
2.4.3.2	Couche paquet du flux de transport	18
2.4.3.3	Définition sémantique des champs dans la couche paquet du flux de transport	18
2.4.3.4	Champ d'adaptation.....	21
2.4.3.5	Définition sémantique des champs contenus dans la structure adaptation_field.....	21
2.4.3.6	Paquet de flux PES.....	30
2.4.3.7	Définition sémantique des champs contenus dans un paquet de flux PES	30
2.4.3.8	Acheminement de flux de programme et de flux de système ISO/CEI 11172-1 dans le flux de transport.....	40
2.4.4	Informations spécifiques du programme.....	41
2.4.4.1	Pointeur	43
2.4.4.2	Définition sémantique des champs dans la syntaxe du pointeur	43
2.4.4.3	Table d'association de programmes	43
2.4.4.4	Affectation des identificateurs de table.....	43
2.4.4.5	Définition sémantique des champs contenus dans une section d'association de programmes.....	43
2.4.4.6	Table d'accès conditionnel	45
2.4.4.7	Définition sémantique des champs dans une section d'accès conditionnel	45
2.4.4.8	Table de conditionnement de programme.....	46
2.4.4.9	Définition sémantique des champs contenus dans la section de conditionnement de flux de programme pour le flux de transport	46
2.4.4.10	Syntaxe de la section relative aux données privées	47
2.4.4.11	Définition sémantique des champs contenus dans une section de données privées	48
2.5	Prescriptions applicables au flux binaire de programme	48
2.5.1	Structure de codage et paramètres du flux de programme	48
2.5.2	Décodeur modèle du système pour le flux de programme.....	50
2.5.2.1	Fréquence de l'horloge système	51
2.5.2.2	Entrée dans le décodeur modèle du système pour le flux de programme	51
2.5.2.3	Mise en mémoire tampon.....	52
2.5.2.4	Flux de paquets PES	53
2.5.2.5	Décodage et présentation	54
2.5.3	Spécification de la syntaxe du flux de programme	54
2.5.3.1	Flux de programme	54
2.5.3.2	Définition sémantique des champs dans un flux de programme.....	54
2.5.3.3	Couche bloc de paquets dans un flux de programme.....	54
2.5.3.4	Définition sémantique des champs contenus dans un bloc de paquets pour flux de programme.....	54
2.5.3.5	En-tête de données systèmes.....	54
2.5.3.6	Définition sémantique des champs contenus dans l'en-tête système.....	56
2.5.3.7	Couche paquet du flux de programme	58
2.5.4	Conditionnement du flux de programme	58
2.5.4.1	Syntaxe de la structure program_stream_map	58
2.5.4.2	Définition sémantique des champs contenus dans la structure program_stream_map.....	58
2.5.5	Répertoire du flux de programme	59
2.5.5.1	Syntaxe du paquet de flux PES pour le répertoire de flux de programme	60
2.5.5.2	Définition sémantique des champs contenus dans un répertoire de flux de programme	60

	<i>Page</i>
2.6	Descripteurs de programme et d'élément de programme 62
2.6.1	Définition sémantique des champs des descripteurs de programme et d'élément de programme 62
2.6.2	Descripteur de flux vidéo 63
2.6.3	Définition sémantique des champs dans un descripteur de flux vidéo 63
2.6.4	Descripteur de flux audio 64
2.6.5	Définition sémantique des champs dans un descripteur de flux audio 64
2.6.6	Descripteur de hiérarchie 64
2.6.7	Définition sémantique des champs dans un descripteur de hiérarchie 65
2.6.8	Descripteur d'enregistrement 65
2.6.9	Définition sémantique des champs dans un descripteur d'enregistrement 66
2.6.10	Descripteur d'alignement de flux de données 66
2.6.11	Définition sémantique des champs dans un descripteur d'alignement de flux de données 66
2.6.12	Descripteur de grille de fond réceptrice 67
2.6.13	Définition sémantique des champs dans un descripteur de grille de fond réceptrice 67
2.6.14	Descripteur de fenêtre vidéo 68
2.6.15	Définition sémantique des champs dans un descripteur de fenêtre vidéo 68
2.6.16	Descripteur d'accès conditionnel 68
2.6.17	Définition sémantique des champs dans un descripteur d'accès conditionnel 69
2.6.18	Descripteur de code de langue selon ISO 639 69
2.6.19	Définition sémantique des champs dans un descripteur de code de langue selon l'ISO 639 69
2.6.20	Descripteur d'horloge système 70
2.6.21	Définition sémantique des champs dans un descripteur d'horloge système 70
2.6.22	Descripteur d'utilisation du tampon de multiplexage 70
2.6.23	Définition sémantique des champs dans un descripteur d'utilisation du tampon de multiplexage 71
2.6.24	Descripteur de droit d'auteur 71
2.6.25	Définition sémantique des champs dans un descripteur de droit d'auteur 71
2.6.26	Descripteur de débit binaire maximal 72
2.6.27	Définition sémantique des champs dans un descripteur de débit binaire maximal 72
2.6.28	Descripteur d'indicateur de données privées 72
2.6.29	Définition sémantique des champs dans un descripteur d'indicateur de données privées 72
2.6.30	Descripteur de tampon de lissage 72
2.6.31	Définition sémantique des champs dans un descripteur de mémoire de lissage 73
2.6.32	Descripteur STD 73
2.6.33	Définition sémantique des champs du descripteur STD 73
2.6.34	Descripteur IBP 74
2.6.35	Définition sémantique des champs du descripteur IBP 74
2.7	Restrictions appliquées à la sémantique des flux binaires multiplexés 74
2.7.1	Fréquence de codage de la référence temporelle du système 74
2.7.2	Fréquence de codage de la référence temporelle de programme 74
2.7.3	Fréquence de codage de la référence temporelle d'un flux élémentaire 75
2.7.4	Fréquence de codage de l'élément présentation timestamp 75
2.7.5	Codage conditionnel des éléments horodateurs 75
2.7.6	Contraintes de synchronisation applicables au décodage échelonné 75
2.7.7	Fréquence de codage de l'élément P-STD_buffer_size dans les en-têtes de paquet d'un flux PES 76
2.7.8	Codage d'en-tête système dans le flux de programme 76
2.7.9	Flux de programme de type flux paramétrique contraint du système 76
2.7.10	Verrouillage de la fréquence des échantillons dans les flux de transport 77
2.8	Compatibilité avec ISO/CEI 11172 78
Annexe A	– Modèle de décodeur pour le contrôle CRC 79
A.0	Décodeur modèle de contrôle CRC 79

	<i>Page</i>
Annexe B – Commande et contrôle des supports d'enregistrement numérique (DSM-CC).....	80
B.0 Introduction.....	80
B.0.1 Objet	80
B.0.2 Futures applications	80
B.0.3 Avantages.....	80
B.0.4 Fonctions fondamentales.....	81
B.0.4.1 Sélection du flux	81
B.0.4.2 Extraction	81
B.0.4.3 Enregistrement	81
B.1 Eléments généraux	81
B.1.1 Domaine d'application.....	81
B.1.2 Vue d'ensemble du mode d'application du protocole DSM-CC.....	81
B.1.3 Transmission des commandes et acquittements du protocole DSM-CC.....	82
B.2 Eléments techniques.....	83
B.2.1 Définitions.....	83
B.2.2 Spécification de la syntaxe du protocole DSM-CC	84
B.2.3 Sémantique des champs contenus dans la spécification syntaxique du protocole DSM-CC	84
B.2.4 Couche de commande	85
B.2.5 Sémantique des champs contenus dans la couche de commande	85
B.2.6 Couche d'acquiescement	87
B.2.7 Sémantique des champs contenus dans la couche d'acquiescement	87
B.2.8 Code temporel.....	88
B.2.9 Sémantique des champs contenus dans la structure de code temporel.....	88
Annexe C – Informations spécifiques du programme	89
C.0 Explications relatives aux informations spécifiques du programme (PSI) contenues dans un flux de transport	89
C.1 Introduction.....	89
C.2 Mécanisme fonctionnel.....	89
C.3 Insertion de sections dans des paquets du flux de transport.....	90
C.4 Fréquences de répétition et accès aléatoire	91
C.5 En quoi consiste un programme?.....	91
C.6 Affectation du numéro de programme	91
C.7 Utilisation des informations PSI dans un système type	92
C.8 Relations entre structures d'informations PSI	92
C.8.1 Table d'association de programmes (PAT)	92
C.8.2 Table de conditionnement de programme (PMT)	93
C.8.3 Table d'accès conditionnel (CAT).....	93
C.8.4 Table d'informations sur le réseau (NIT)	93
C.8.5 Sections privées	93
C.8.6 Descripteurs	93
C.9 Affectation spectrale et temps d'acquisition du signal	95
Annexe D – Implications de la présente Recommandation Norme internationale (systèmes) quant au modèle de synchronisation et à ses applications.....	98
D.0 Introduction.....	98
D.0.1 Modèle de gestion du temps.....	98
D.0.2 Synchronisation de la présentation des données audio et vidéo.....	100
D.0.3 Recalage sur l'horloge de base du système dans le décodeur.....	101
D.0.4 Gigue des références SCR et PCR	104
D.0.5 Récupération d'horloge en présence d'une gigue de réseau	105
D.0.6 Horloge système utilisée pour construire une sous-porteuse de chrominance	105
D.0.7 Reconstruction des données vidéo et audio en composantes	107
D.0.8 Glissement des trames.....	107
D.0.9 Lissage de la gigue due au réseau	107
Annexe E – Applications de transmission de données	109
E.0 Considérations générales.....	109
E.1 Suggestion.....	109

	<i>Page</i>
Annexe F – Présentation graphique de la syntaxe pour la présente Recommandation Norme internationale	110
F.0 Introduction	110
F.0.1 Syntaxe d'un flux de transport.....	110
F.0.2 Paquet de flux PES.....	111
F.0.3 Section de table d'association de programmes	112
F.0.4 Section de table d'accès conditionnel.....	112
F.0.5 Section de table de conditionnement de programme pour flux de transport.....	113
F.0.6 Section de table de données privées.....	113
F.0.7 Flux de programme	114
F.0.8 Conditionnement de flux de programme.....	115
Annexe G – Informations générales	116
G.0 Informations générales	116
G.0.1 Emulation d'octets de synchronisation	116
G.0.2 Etat d'image sautée et processus de décodage.....	116
G.0.3 Sélection de valeurs d'identificateur PID	116
G.0.4 Emulation d'un code de début de paquet PES.....	116
Annexe H – Données privées	117
H.0 Données privées	117
Annexe I – Conformité des systèmes et interface en temps réel	119
I.0 Conformité des systèmes et interface en temps réel.....	119
Annexe J – Interfaçage entre réseaux générateurs de gigue et décodeurs MPEG-2.....	120
J.0 Introduction.....	120
J.1 Modèle de conformité de réseau	120
J.2 Spécification de réseau pour le lissage de gigue	121
J.3 Exemples de mises en oeuvre de décodeur	122
J.3.1 Adaptateur de réseau suivi d'un décodeur MPEG-2	122
J.3.2 Décodeur intégré.....	122
Annexe K – Combinaison des flux de transport.....	124
K.0 Introduction.....	124
K.1 Les différents types de points de combinaison	124
K.1.1 Points de combinaison ordinaire	124
K.1.2 Points de combinaison transparente	125
K.2 Comportement des décodeurs en présence de combinaisons.....	125
K.2.1 Combinaisons non transparentes.....	125
K.2.2 Combinaisons transparentes.....	125
K.2.3 Surremplissage de la mémoire tampon	126

Résumé

La présente Recommandation | Norme internationale spécifie des méthodes génériques pour le multiplexage, la synchronisation et le recalage sur la base de temps dans les systèmes multimédias. Elle propose un multiplexage multimédia en mode paquet dans lequel chaque flux binaire élémentaire est segmenté en un flux élémentaire empaqueté (PES) (*packetized elementary stream*); puis les paquets correspondants sont multiplexés pour former un des deux flux suivants: le flux de programme (PS) (*program stream*), qui est un multiplex de paquets de flux PES de longueur variable, conçu pour les environnements sans erreur; et le flux de transport (TS) (*transport stream*), qui est formé de paquets de longueur fixe (188 octets) et qui a la fonction de multiplexage de plusieurs flux de programme ainsi que de divers paquets de flux PES, pour usage dans les environnements exposés aux erreurs. La synchronisation et le recalage sur base de temps des systèmes multimédias sont assurés par des éléments horodateurs, en termes d'horloge système et de présentation/décodage système.

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 recombinaés 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.

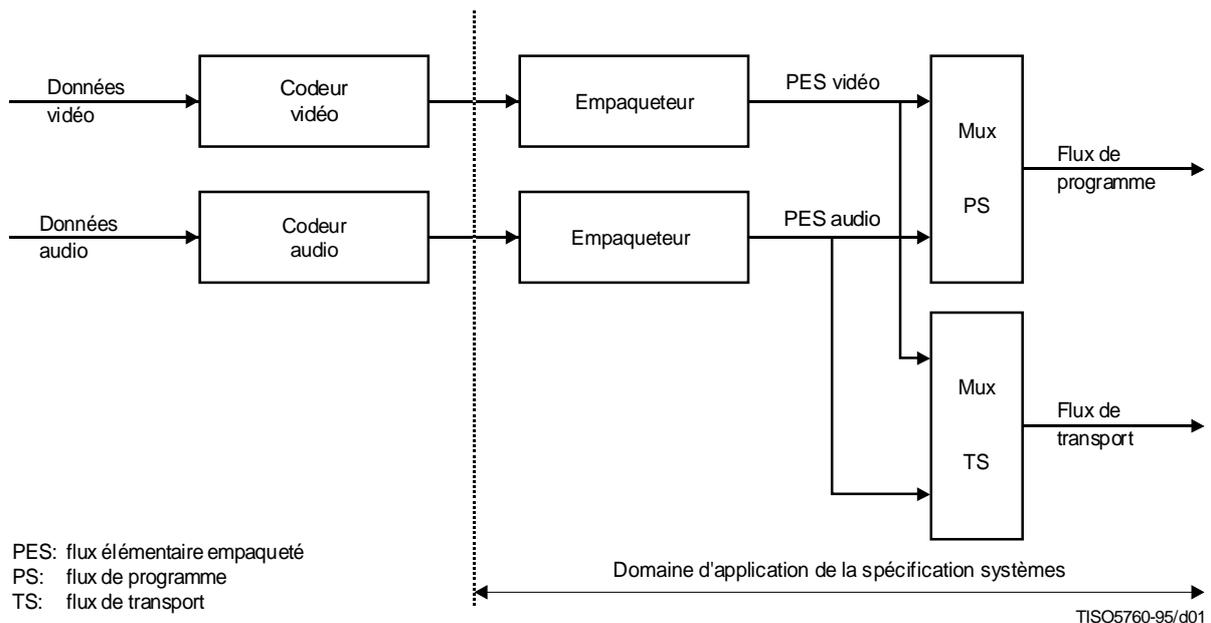


Figure Intro. 1 – Synoptique simplifié du domaine d'application de la présente Recommandation | Norme internationale

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) 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 de mise en œuvre 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 la mise en œuvre 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 la mise en œuvre 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). 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.

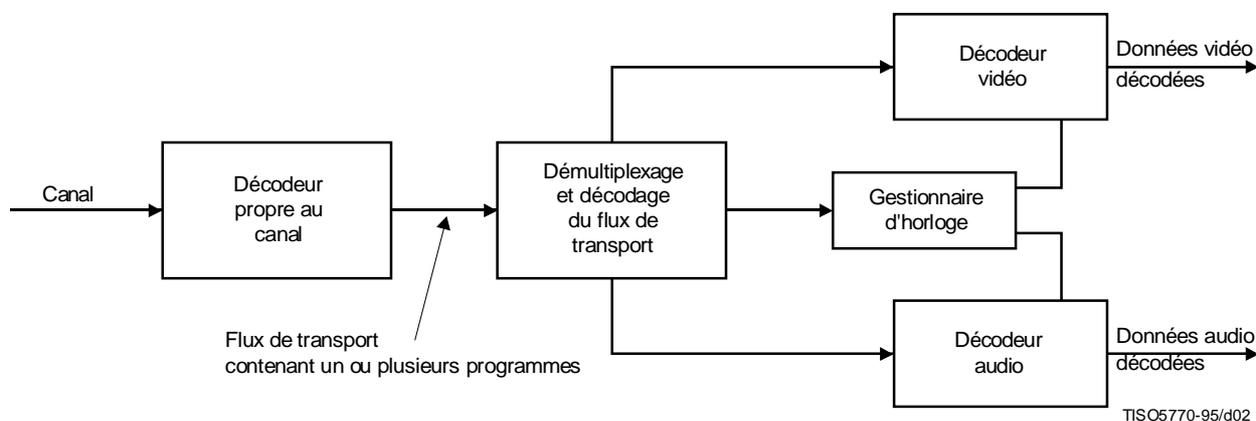


Figure Intro. 2 – Prototype (à titre d'exemple) de multiplexage et de décodage pour le transport

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.

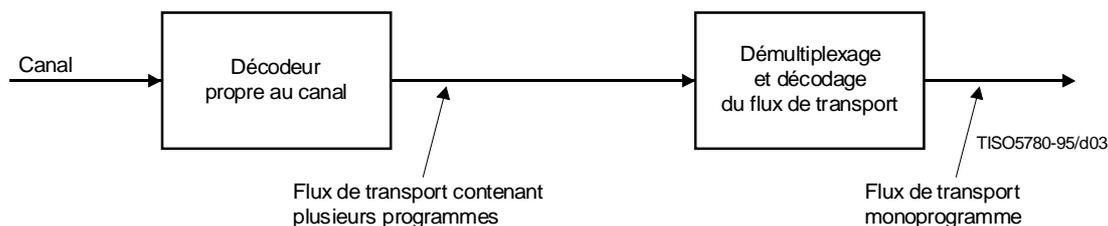


Figure Intro. 3 – Prototype (à titre d'exemple) de multiplexage de flux de transport

La Figure Intro. 4 illustre le cas d'un flux de transport 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 mises en œuvre spécifiques de démultiplexeurs ou de décodeurs comportent toutes ces fonctions.

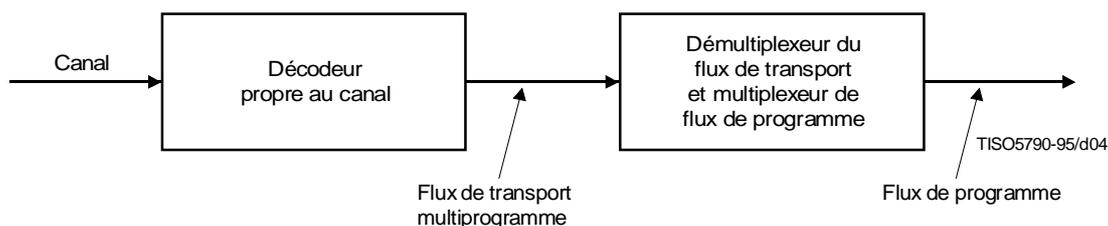


Figure Intro. 4 – Prototype de conversion de flux de transport à flux de programme

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) 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.

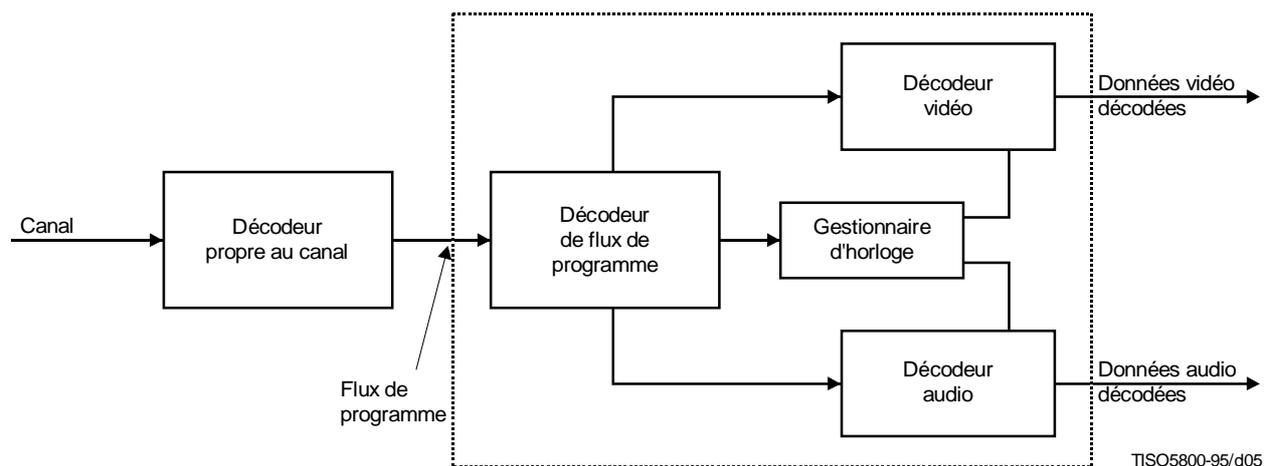


Figure Intro. 5 – Prototype de décodeur pour flux de programme

Le prototype de décodeur pour flux de programme, représenté sur la Figure Intro. 5, se compose de décodeur 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/ou 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 appliqués 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 mises en œuvre 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 de mettre en œuvre des systèmes impliquant un retard aux extrémités constant. Il est possible d'appliquer 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 la mise en œuvre 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) 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) (voir 2.4.2) et en «décodeur modèle du système pour les flux de programme» (P-STD) (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 conformes à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 et des données audio conformes à l'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). Un protocole de commande et contrôle de support d'enregistrement numérique (DSM CC) 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 – CONSIDÉRATIONS GÉNÉRALES

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 qui sont définies dans les Parties 2 et 3 de ISO/CEI 13818. La couche système assure cinq 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; et
- 5) le repérage temporel.

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 les 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. Toute Recommandation ou Norme internationale est sujette 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 internationales 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 UIT-T en vigueur.

1.3 Recommandations | Normes internationales identiques

- Recommandation UIT-T H.262 (1995) | ISO/CEI 13818-2:…¹⁾, *Technologies de l'information – Codage des images animées et du son associé: Données vidéo.*

¹⁾ A publier.

1.4 Autres références

- ISO 639-2:…²⁾, *Terminologie – Code pour la représentation des noms de langue. Partie 2: Alpha-3 code.*
- ISO 8859-1:1987, *Traitement de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 1: Alphabet latin n° 1.*
- 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 13522-1:…²⁾, *Technologies de l'information – Codage de l'information multimédia et hypermédia – Partie 1: Représentation d'objet MHEG – Notation de base (ASN.1).*
- ISO/CEI 13818-3:1995, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 3: son.*
- Recommandation UIT-R BT.601-3, *Paramètres de codage de télévision numérique pour studios.*
- Recommandation UIT-R BT.470-2, *Systèmes de télévision.*
- Recommandation UIT-R BR.648, *Enregistrement numérique des signaux audio.*
- Rapport UIT-R BO.955-2, *Radiodiffusion sonore par satellite pour récepteurs portatifs et récepteurs dans les véhicules automobiles.*
- Recommandation J.17 du CCITT (1988), *Préaccentuation utilisée sur les circuits pour transmissions radiophoniques.*
- IEEE Standard 1180:1990, *Standard Specification for the Implementations of 8 by 8 Inverse Discrete Cosine Transform.*
- Publication CEI 908:1987, *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.

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

2.1.3 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.

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

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

²⁾ A publier.

- 2.1.6 compression:** réduction du nombre de bits utilisés pour représenter une donnée élémentaire.
- 2.1.7 débit constant:** exploitation dans laquelle le débit est constant du début à la fin du flux binaire comprimé.
- 2.1.8 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.9 CRC:** contrôle de redondance cyclique destiné à vérifier que les données sont correctes.
- 2.1.10 élément de données:** donnée élémentaire telle qu'elle est représentée avant codage et après décodage.
- 2.1.11 flux décodé:** reconstitution décodée d'un flux binaire comprimé.
- 2.1.12 décodeur:** matérialisation d'un processus de décodage.
- 2.1.13 décodage (processus):** processus défini dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, consistant à lire un flux binaire codé entrant et à produire, après décodage, des images ou des échantillons audio.
- 2.1.14 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.
- 2.1.15 support d'enregistrement numérique (DSM) (*digital storage media*):** dispositif ou système d'enregistrement ou de transmission numérique.
- 2.1.16 DSM-CC (*digital storage media command and control*):** commande et contrôle de support d'enregistrement numérique.
- 2.1.17 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/ou de commande qui, en principe, relèvent spécifiquement d'un flux binaire.
- 2.1.18 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.
- 2.1.19 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.
- 2.1.20 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.
- 2.1.21 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.
- 2.1.22 codeur:** matérialisation d'un processus de codage.
- 2.1.23 codage (processus):** processus, non spécifié dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, 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 Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.24 codage entropique (ou statistique):** codage sans perte à longueur variable de la représentation numérique d'un signal afin de réduire la redondance.
- 2.1.25 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.
- 2.1.26 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.**
- 2.1.27 interdite (valeur):** qualificatif qui, lorsqu'il est utilisé dans les articles de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 définissant le flux binaire codé, indique que la valeur spécifiée ne doit jamais être utilisée.
- 2.1.28 flux (multiplexé) (système):** flux binaire composé de 0 ou de plus de 0 flux élémentaire(s) combiné(s) de la manière décrite dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

- 2.1.29 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 Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.30 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.31 données en paquet (système):** octets de données contigus d'un flux élémentaire présents dans un paquet.
- 2.1.32 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.33 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.
- 2.1.34 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 charge 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.
- 2.1.35 PES (système):** abréviation du terme Packetized Elementary Stream (flux élémentaire empaqueté).
- 2.1.36 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.37 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.
- 2.1.38 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.39 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.
- 2.1.40 unité de présentation; PU (système) (*presentation unit*):** unité d'accès audio décodée ou image décodée.
- 2.1.41 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.
- 2.1.42 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.
- 2.1.43 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.
- 2.1.44 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Un exemple de PSI privées est le tableau d'information de réseau non obligatoire.
- 2.1.45 accès aléatoire:** processus consistant à commencer la lecture et le décodage du flux binaire codé, à un point arbitraire.
- 2.1.46 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 Rec. UIT-T H.222.0 | ISO/CEI 13818-1, tous les bits de valeur réservée doivent être mis à '1'.

2.1.47 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.

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

2.1.49 combinaison (système): concaténation réalisée au niveau du système de deux flux élémentaires distincts. Le flux système résultant est entièrement conforme aux dispositions de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. La combinaison peut créer des discontinuités de la base de temps du compteur de continuité, des informations spécifiques du programme, et du décodage.

2.1.50 codes de déclenchement (système): mots codés à 32 bits insérés 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 2-18 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

2.1.51 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.

2.1.52 image fixe: image codée qui se compose d'une séquence vidéo contenant exactement une seule image à codage intratrame. Cette image est assortie d'un horodatage de présentation et l'instant de présentation des images suivantes (si elles existent) est d'au moins deux périodes d'image après celui de l'image fixe.

2.1.53 en-tête de système (système): structure de données qui est définie au 2.5.3.5 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 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.

2.1.54 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.

2.1.55 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.

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

2.1.57 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.

2.1.58 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 arrondissements 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)
++	Incrémementation
--	Décrémementation
* 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'intervalle de temps effectifs dans le cas où block_type == 2, avec $0 \geq \text{window} \geq 2$.

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 mnémorique 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^{ième} élément d'une table de données.

data_element [m][n] L'élément de données [m][n] est le m+1, n+1^{ième} 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^{ième} 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 met en œuvre 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 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. 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 quatre tables PSI:

- table d'association de programmes;
- table de conditionnement de programme;
- table d'accès conditionnel;
- table d'informations sur le réseau.

Ces tables contiennent les informations nécessaires et suffisantes pour démultiplexer et présenter les programmes. La table de conditionnement de programme présentée dans le Tableau 2-28 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.

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 le Tableau 2-27 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.

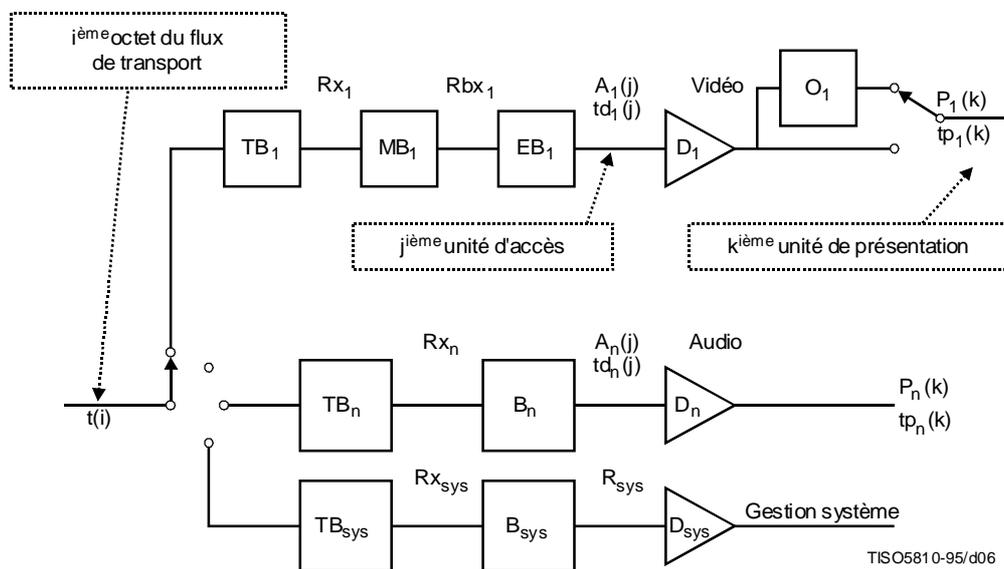


Figure 2-1 – Notation d'un décodeur modèle de système pour un flux de transport

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 ^{ième} 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, où i est l'indice désignant l'octet final du champ program_clock_reference_base.
A _n (j)	est la j ^{ième} 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 ^{ième} unité d'accès dans le flux élémentaire n.
P _n (k)	est la k ^{ième} 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 ^{ième} 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:

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

dérive de la fréquence d'horloge système: $\leq 75 \times 10^{-3}$ 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 binaires définis dans la présente Spécification sont mesurés en fonction de la fréquence d'horloge système. Par exemple, un débit binaire 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 ($300 \times 2^{33}/\text{system_clock_frequency}$). Ce rapport est dû au codage sur 33 bits des informations de calage de la référence PCR sur $1/300^{\text{e}}$ 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 i^{e} 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 décodage, et par comptage des octets du flux de transport complet intercalés entre les champs PCR successifs du programme à décodage. Le champ PCR est codé, selon l'équation 2-1, en deux parties: d'une part, en unités de période correspondant à $1/300^{\text{e}}$ de la fréquence d'horloge système, dite PCR_base (équation 2-2) et, d'autre part, en unités de période correspondant à la fréquence d'horloge système dite PCR_extension (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:

$$PCR(i) = PCR_base(i) \times 300 + PCR_ext(i) \quad (2-1)$$

où:

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

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

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.

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

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:

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

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. Puisqu'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 binaire 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 données issues du flux élémentaire n comme indiqué par leurs identificateurs PID, sont injectés dans la mémoire de transport TB_n , pour le flux n . Cela vise également les paquets de transport en double et les paquets sans capacité utile. Le transfert à la mémoire TB_n du i^e octet issu de l'entrée du décodeur STD est instantané, de manière que ce i^e octet pénètre au temps $t(i)$ dans la mémoire tampon pour le flux n , de capacité TBS_n .

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 R_{x_n} est égal à zéro;
- sinon, pour des données vidéo:

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

où

$R_{max}[\text{profil, niveau}]$ 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;

R_{x_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 binaire maximal indiqué dans ISO/CEI 11172-2 pour un flux binaire à paramètres contraints;

- pour des données audio:

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

- pour des données système:

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

Le débit R_{x_n} est mesuré par rapport à la fréquence d'horloge du système.

Les paquets de transport complets contenant des informations systèmes pour le programme dont le décodage a été sélectionné pénètrent, au débit du flux de transport, dans la mémoire tampon TB_{sys} . Il s'agira notamment des paquets de transport dont les identificateurs ont la valeur 0 ou 1, ainsi que de tous les paquets de transport identifiés au moyen de la table d'association de programmes (Tableau 2-25) comme ayant, pour le programme choisi, la valeur `program_map_PID`. Les données issues de la table d'informations sur le réseau (NIT), telles que spécifiées par l'identificateur PID de table NIT, ne sont pas transférées dans la mémoire TB_{sys} .

Les octets sont extraits de TB_{sys} au débit $R_{x_{sys}}$ 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 telle qu'elle est acheminée dans l'en-tête de séquence. (Voir le résumé «paramètres contraints» dans 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

$$MBS_n = BS_{mux} + BS_{oh} + VBV_{max}[\text{profil, niveau}] - \text{capacité tampon VBV}$$

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

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

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

$$BS_{mux} = 4 \text{ ms} \times R_{max}[\text{profil, niveau}]$$

tandis que $VBV_{max}[\text{profil, niveau}]$ est défini au Tableau 8-14 de la Rec. UIT-T H.262 | ISO/CEI 13818-2 et $R_{max}[\text{profil, niveau}]$ 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.)

ISO/CEI 13818-1 : 1996 (F)

Pour le niveau supérieur 1440 et le niveau supérieur

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

où BS_{oh} est défini par:

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

et BS_{mux} est défini par:

$$BS_{mux} = 4 \text{ ms} \times R_{max}[\text{profil, niveau}]$$

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

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

$$MBS_n = BS_{mux} + BS_{oh} + VB_{V_{max}} - \text{capacité tampon VB}_{V}$$

où BS_{oh} est défini par:

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

et BS_{mux} est défini par:

$$BS_{mux} = 4 \text{ ms} \times R_{max}$$

tandis que R_{max} et $VB_{V_{max}}$ désignent respectivement le débit binaire maximal et la capacité maximale de tampon VB_V pour un flux paramétrique binaire contraint de ISO/CEI 11172-2.

Une portion $BS_{mux} = 4 \text{ ms} \times R_{max}[\text{profil, niveau}]$ 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 1 – 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

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

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:

$$BS_{dec} + BS_{oh} = 2848 \text{ octets}$$

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.

Systemes

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 2.6.3.2) 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; ou encore
- le mode enrichissement est à «vrai» (voir 2.4.3.6).

Pour le niveau inférieur et le niveau principal

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

Pour le niveau 1440 supérieur et le niveau supérieur

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

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

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

où R_{\max} désigne le débit binaire maximal pour un flux paramétrique binaire contraint de 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 R_{bx} ; 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 éliminés. 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 2.6.3.2) 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 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:

$$R_{bx}(j) = NB(j) / (\text{délai VBV}(j) - \text{délai VBV}(j + 1) + td_n(j + 1) - td_n(j)) \quad (2-6)$$

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 2 – 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 binaire $R_{bx}(j)$ 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 2-29), dont la définition figure dans la Rec. UIT-T H.262 | ISO/CEI 13818-2, et inférieure ou égale au débit binaire maximal admis dans le jeu de paramètres soumis à des contraintes, défini pour les flux élémentaires de type 0x01 (voir 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 éliminés. 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 imbriqués 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} .

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

NOTE 3 – 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 réordonnées 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.

ISO/CEI 13818-1 : 1996 (F)

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. 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 $A_n(j)$.

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 :

$$0 \leq F_n(t)$$

2.4.3 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 le Tableau 2-1.

Tableau 2-1 – Flux de transport

Syntaxe	Nombre de bits	Mnémonique
<pre>MPEG_transport_stream() { do { transport_packet() } while (nextbits() == sync_byte) }</pre>		

2.4.3.2 Couche paquet du flux de transport

Voir le Tableau 2-2.

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).

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

Syntaxe	Nombre de bits	Mnémonique
<pre> transport_packet() { sync_byte transport_error_indicator payload_unit_start_indicator transport_priority PID transport_scrambling_control adaptation_field_control continuity_counter 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 } } } </pre>	<p>8</p> <p>1</p> <p>1</p> <p>1</p> <p>13</p> <p>2</p> <p>2</p> <p>4</p> <p>8</p>	<p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p>

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 le Tableau 2-29).

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 le Tableau 2-29).

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 éléments binaires. Il indique le type de données enregistrées dans la capacité utile du paquet. La valeur 0x0000 d'un PID est réservée à la table d'association de programmes (voir le Tableau 2-25). La valeur de PID 0x0001 est réservée à la table d'accès conditionnel (voir le Tableau 2-27). Les valeurs de PID 0x0002 à 0x000F sont réservées. La valeur de PID 0x1FFF est réservée aux paquets vides (voir le Tableau 2-3).

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 le Tableau 2-4).

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/ou d'une capacité utile (voir le Tableau 2-5).

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

Valeur	Description
0x0000	Table d'association de programmes
0x0001	Table d'accès conditionnel
0x0002 - 0x000F	Valeur réservée
0x00010 ... 0x1FFE	Susceptible d'être affectée comme identificateur PID de réseau, Program_map_PID, identificateur PID élémentaire ou autre
0x1FFF	Paquet vide
NOTE – Les paquets de transport avec identificateurs PID 0x0000, 0x0001 et 0x0010 - 0x1FFE peuvent contenir un champ PCR.	

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

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 rejeter 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 le Tableau 2-6.

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.

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

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

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 «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 – Premier octet d'un en-tête de séquence vidéo;
- données audio – Premier octet d'une trame audio.

A la suite d'une discontinuité du compteur de continuité d'un paquet de transport identifié comme contenant des données de flux élémentaire, le premier octet de données de flux élémentaire d'un paquet de transport ayant le même identificateur PID doit être le premier octet du point d'accès du flux élémentaire ou, dans le cas des données vidéo, le premier octet d'un point d'accès du flux élémentaire ou encore d'un code de fin de séquence suivi d'un point d'accès. 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 rejeté. Un flux de transport ne doit pas être construit de telle sorte que le rejet 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, lorsqu'il est mis à '1', le prochain paquet PES du début de capacité utile des paquets de transport (voir le Tableau 2-29) de même identificateur PID doit contenir le premier octet d'un en-tête de séquence vidéo, s'il s'agit d'un flux de transport PES de type '1' ou '2' (données vidéo) ou doit contenir le premier octet d'une trame audio s'il s'agit d'un flux de transport de type '3' ou '4' (données audio). 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. En cas de données vidéo, ce fanion ne peut être mis à '1' que si la capacité utile contient un ou plusieurs octets issus d'une tranche à codage intratrame. 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:

$$OPCR(i) = OPCR_base(i) \times 300 + OPCR_ext(i) \quad (2-8)$$

où

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

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

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 nième paquet qui suit le point de combinaison (à l'exclusion des paquets dupliqués et des paquets sans capacité utile).

Aux fins du présent paragraphe, la définition d'un point d'accès est la suivante:

- données vidéo – Le premier octet d'un en-tête de séquence vidéo;
- données audio – Le premier octet d'une trame audio.

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 la valeur de ce fanion est fixée, si le flux élémentaire contenu dans cet identificateur PID est un flux audio, le champ réservé au type de combinaison doit être mis à '0000'; si le flux élémentaire est un flux vidéo, il doit vérifier les contraintes correspondant à la valeur du type de combinaison.

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:

$$\text{offset} = t_1(i) - t(i)$$

$$\text{ltw_offset} = \text{offset}/1$$

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 $BS_{dec} + 1$ octets 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 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 $A_i, A_{i+1}, \dots, A_{i+N}$, 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:

$$t_1(A_{i+j}) = t_1(A_i) + j * 188 * 8 \text{ bits/octet} / R$$

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:

$$\text{offset} = t_1(A_i) - t(A_i)$$

correspondant à la valeur $t_1(.)$ calculée d'après la relation ci-dessus = $t_1(A_i)$, 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 est un flux de données audio, le champ `splice_type` doit prendre la valeur '0000'. Si le flux élémentaire transporté sous cet identificateur PID est un flux de données vidéo, ce champ indique les conditions de combinaison que doit vérifier ce flux élémentaire. Les Tableaux 2-7 à 2-16 des pages suivantes 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)**

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) 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) 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) 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) 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) 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

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 rejetée par le décodeur.

2.4.3.6 Paquet de flux PES

Voir le Tableau 2-17.

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 2-18. 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 2-18. 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 le Tableau 2-19).

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 du code de déclenchement de données vidéo ou du mot de synchronisation audio défini par le descripteur d'alignement de flux de données figurant 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 indiqué par `alignment_type` '01' figurant dans les Tableaux 2-47 et 2-48. La valeur '0' indique que l'apparition d'un tel alignement est indéfinie.

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.

Tableau 2-17 – 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 != DSM-CC_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 2-17 – Paquet de flux PES (*fin*)

Syntaxe	Nombre de bits	Mnémonique
<pre> 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 == DSM-CC_stream stream_id == ITU-T Rec. H.222.1 type E stream) { for (i = 0; i < PES_packet_length; i++) { PES_packet_data_byte } } else if (stream_id == padding_stream) { for (i = 0; i < PES_packet_length; i++) { padding_byte } } </pre>	8	bslbf
	8	bslbf

Tableau 2-18 – 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		ISO/IEC 13818-3 or ISO/IEC 11172-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 ISO/IEC 13818-2 or ISO/IEC 11172-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 ISO/IEC 13818-1 Annex B or ISO/IEC 13818-6_DSM-CC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010 ... 1111 1110		Reserved data stream
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.

NOTES

- Les paquets PES de type program_stream_map ont une syntaxe unique, qui est spécifiée au 2.5.4.1.
- Les paquets de type private_stream_1 et ISO/CEI_13552_stream suivent la même syntaxe de paquet PES que pour les flux vidéo selon la Rec. UIT-T H.262 | ISO/CEI 13818-2 et que pour les flux audio selon ISO/CEI 13818-3.
- Les paquets PES de type private_stream_2, ECM_stream et EMM_stream sont similaires au champ private_stream_1 sauf qu'aucune syntaxe n'est spécifiée après le champ PES_packet_length.
- Les paquets PES de type program_stream_directory ont une syntaxe unique, spécifiée au 2.5.5.
- Les paquets PES de type DSM-CC_stream ont une syntaxe unique, qui est spécifiée dans ISO/CEI 13818-6.
- Cet identificateur stream_id est associé au type de flux 0x09 indiqué au Tableau 2-29.
- Cet identificateur stream_id est utilisé uniquement dans les paquets de flux 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 2.4.3.7).

Tableau 2-19 – 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

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 la relation 2-11 ci-dessous. Voir 2.7.4 pour les contraintes applicables à la fréquence de codage des horodateurs de présentation.

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

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.

Dans le cas de données vidéo, si l'en-tête de paquet PES contient un horodateur PTS, celui-ci doit se rapporter à l'unité d'accès contenant le premier code de déclenchement d'image qui commence dans ledit paquet. Un code de déclenchement d'image commence dans un paquet PES si le premier octet de ce code est présent dans les données du paquet.

Pour les unités de présentation de données audio (PU), pour les unités de présentation de données vidéo en séquences à faible délai et pour les images à codage de type B, l'instant de présentation $tp_n(k)$ doit être égal à l'instant de décodage $td_n(k)$.

Pour les images à codage de type I ou P en séquences sans faible délai et s'il n'y a pas d'indication de discontinuité du décodage entre les unités d'accès (AU) k et k' , l'instant de présentation $tp_n(k)$ doit être égal à l'instant de décodage $td_n(k')$ 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 $tp_n(k)$ et $td_n(k)$ doit être la même que si le flux initial avait continué sans discontinuité, ni terminaison.

NOTE 1 – Une séquence à faible délai est une suite de données vidéo dans laquelle le fanion `low_delay` est activé (voir 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2).

S'il y a un 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 échelonné, 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:

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

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

ISO/CEI 13818-1 : 1996 (F)

Dans le cas de données vidéo, si l'en-tête de paquet PES contient un horodateur DTS, celui-ci doit se rapporter à l'unité d'accès contenant le premier code de déclenchement d'image qui commence dans ledit paquet. Un code de déclenchement d'image commence dans un paquet PES si le premier octet de ce code est présent dans les données du paquet.

Dans le cas d'un codage échelonné, 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:

$$ESCR(i) = ESCR_base(i) \times 300 + ESCR_ext(i) \quad (2-13)$$

où

$$ESCR_base(i) = ((system_clock_frequency * t(i)) DIV 300) \% 2^{33} \quad (2-14)$$

$$ESCR_ext(i) = ((system_clock_frequency * t(i)) DIV 1) \% 300 \quad (2-15)$$

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 5 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, le nombre de fois N qu'une image est présentée à la sortie du processus de décodage, est spécifié pour chaque image par la valeur des champs repeat_first_field et trame à balayage progressif, 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 2-20 – 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 2-21, indique le(s) champ(s) qu'il y a lieu d'afficher. Le champ `frequency_truncation` indique qu'un ensemble limité de coefficients peut être présent. Le Tableau 2-22 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 $N * \text{rep_cntrl}$ 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 $N * \text{rep_cntrl}$ 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 $N * \text{rep_cntrl}$ 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 $N * \text{rep_cntrl} / 2$ field. L'autre trame de l'image est ensuite affichée pendant un temps égal à $N - N * \text{rep_cntrl} / 2$ 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 2-21 – 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 2-21, indique le(s) champ(s) 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 2-22, «Valeurs de sélection des coefficients».

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 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.

field_id (identificateur de trame) – Ce champ, codé sur 2 éléments binaires, indique quelle(s) trame(s) il faut afficher conformément au Tableau 2-21.

Tableau 2-21 – 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 la 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 2-22.

Tableau 2-22 – 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 2 – 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_identifieur (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 paramètre 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:

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

autre

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

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.

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.

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 rejeté 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 rejeté 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 le Tableau 2-18). 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 13818-1, 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 13818-1.

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) 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 quatre structures tabulaires, comme indiqué dans le Tableau 2-23. 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 PID prédéfinis, d'autres avec des PID sélectionnables par l'utilisateur.

Tableau 2-23 – Informations spécifiques du programme

Nom de la structure	Type de flux	Identificateur PID réservé	Description
Table d'association de programmes	Rec. UIT-T H.222.0 ISO/CEI 13818-1	0x00	Associe le numéro de programme et la table de conditionnement de programme PID
Table de conditionnement de programme	Rec. UIT-T H.222.0 ISO/CEI 13818-1	Valeur assignée dans la table d'association de programmes	Spécifie les valeurs 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 de PID

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 mettre en correspondance 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 d'application de ces données privées soit la même que pour l'application 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 conditionnement 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, l'application des données privées sur les paquets du flux de transport n'est pas définie 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 2-24.

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 après le dernier octet d'une section; dans ce cas, tous les octets suivants doivent, jusqu'à la fin du paquet de transport, être aussi des octets de bourrage de valeur 0xFF. Ces octets pourront être rejetés par un décodeur. En pareille circonstance, la capacité utile du prochain paquet de flux de transport ayant la même valeur d'identification PID devra commencer par un champ de pointeur de valeur 0x00, indiquant que la prochaine section 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 invalide 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 conditionnement 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 conditionnement 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 conditionnement 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 conditionnement de programme. La version transmise le plus récemment de la section de conditionnement 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 conditionnement de programme acheminée dans les paquets de transport dont la valeur d'identificateur PID est définie comme étant l'identificateur de conditionnement pour ce programme spécifique. Tous les paquets de transport qui acheminent une section donnée de conditionnement de programme de flux de transport doivent avoir la même valeur d'identificateur PID. L'identificateur de paquet pour conditionnement 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 conditionnement de programme de flux de transport. Pour qu'une nouvelle version d'une section de conditionnement 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 conditionnement 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.

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 2-24 ci-dessous.

Tableau 2-24 – 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) – Codé 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 le Tableau 2-25).

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 2-26 ci-dessous.

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 2-26.

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.

Tableau 2-25 – Section relative aux associations de programmes

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

Tableau 2-26 – Valeurs d'affectation des identificateurs table_id

Valeur	Description
0x00	program_association_section
0x01	conditional_access_section(CA_section)
0x02	TS_program_map_section
0x03 - 0x3F	Rec. UIT-T H.222.0 ISO/CEI 13818-1 (valeur réservée)
0x40 - 0xFE	Valeur privée d'utilisateur
0xFF	Valeur interdite

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 doit augmenter d'une unité à chaque modification de la définition de la table d'association de programmes. Dès qu'il atteint la valeur 31, ce compteur revient à zéro. 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, 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 conditionnement 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 conditionnement 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 conditionnement 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) 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 2-27.

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

Tableau 2-27 – 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 2-26.

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 conditionnement de programme

La table de conditionnement de programme fournit les correspondances entre les numéros de programme et les éléments de programme qui les acheminent. Une seule occurrence d'une telle correspondance est appelée «définition de programme». La table de conditionnement 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 le Tableau 2-28).

Tableau 2-28 – Section relative au conditionnement 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 conditionnement 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 conditionnement de programme pour le flux de transport – être mis à la valeur 0x02, comme indiqué dans le Tableau 2-26.

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 conditionnement de flux de programme est applicable. Une seule définition de programme doit être acheminée dans chaque section de conditionnement 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 conditionnement 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 conditionnement 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 conditionnement 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 conditionnement 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 conditionnement 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 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 2-29.

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 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 conditionnement 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 conditionnement 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 le Tableau 2-30).

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 le Tableau 2-26).

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.

2.5 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.

Tableau 2-29 – Assignment des types de flux

Valeur	Description
0x00	UIT-T ISO/CEI réservé
0x01	ISO/CEI 11172-2 Vidéo
0x02	Rec. UIT-T H.262 ISO/CEI 13818-2 Vidéo ou ISO/CEI 11172-2 flux vidéo à paramètres contraints
0x03	ISO/CEI 11172-3 Audio
0x04	ISO/CEI 13818-3 Audio
0x05	Rec. UIT-T H.222.0 ISO/CEI 13818-1 réservé
0x06	Rec. UIT-T H.222.0 ISO/CEI 13818-1 PES paquets contenant des données privées
0x07	ISO/CEI 13522 MHEG
0x08	Rec. UIT-T H.222.0 ISO/CEI 13818-1 Annexe A DSM-CC
0x09	Rec. UIT-T H.222.1
0x0A	ISO/CEI 13818-6 type A
0x0B	ISO/CEI 13818-6 type B
0x0C	ISO/CEI 13818-6 type C
0x0D	ISO/CEI 13818-6 type D
0x0E	ISO/CEI 13818-1 auxiliaire
0x0F - 0x7F	Rec. UIT-T H.222.0 ISO/CEI 13818-1 réservé
0x80 - 0xFF	Utilisateur privé

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 2-30 – 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
}		
}		

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 le Tableau 2-18). Cet en-tête de paquet PES peut contenir soit simplement un horodateur de présentation, soit des horodateurs de décodage 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 binaire.

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. 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 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.

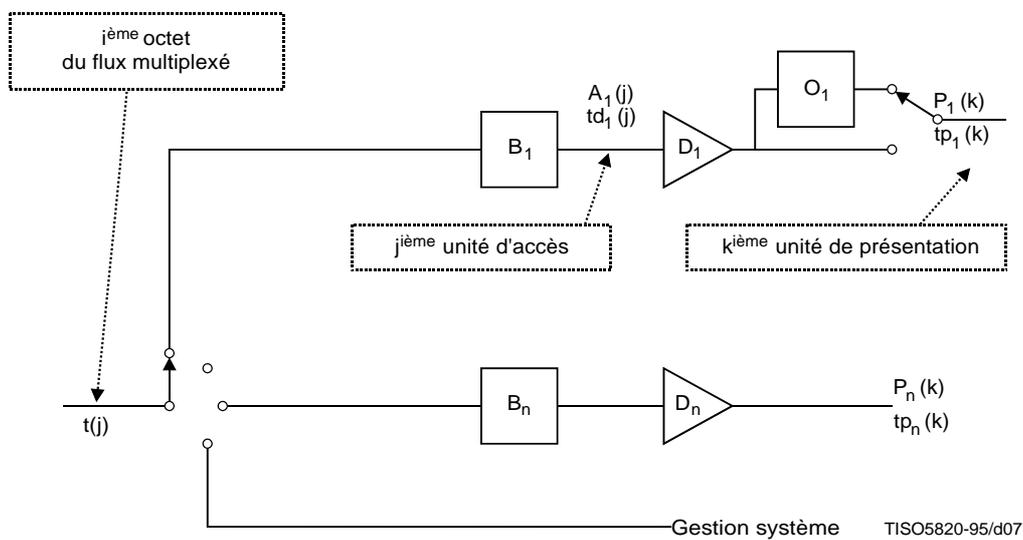


Figure 2-2 – Notation d'un décodeur modèle de système pour un flux de programme

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 $i^{\text{ème}}$ 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, où i est l'indice désignant l'octet final du champ de la base de référence temporelle du système.
A _n (j)	j ^{ème} 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 ^{ème} unité d'accès contenue dans le flux élémentaire n.
P _n (k)	k ^{ème} 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 ^{ème} 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 réordonnement 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:

- $27\ 000\ 000 - 810 \leq \text{system_clock_frequency} \leq 27\ 000\ 000 + 810$;
- dérive de la fréquence d'horloge système: $\leq 75 \times 10^{-3} \text{ Hz/s}$.

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 ($300 \times 2^{33} / \text{system_clock_frequency}$). Ce rapport est dû au codage sur 33 bits des informations de calage de la référence SCR sur $1/300^e$ 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^{ème} 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:

$$SCR(i) = SCR_base(i) \times 300 + SCR_ext(i) \quad (2-18)$$

où

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

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

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.

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

où

i'	est l'indice de l'octet contenant le dernier bit du champ de base <code>system_clock_reference</code> 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 <code>system_clock_reference</code> , en unités d'horloge système
<code>program_mux_rate</code>	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. 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 assujéti à la relation suivante:

$$td_n(j) - t(i) \leq 1 \text{ seconde}$$

sinon, en présence de données vidéo d'image fixe, le délai doit vérifier la relation suivante:

$$td_n(j) - t(i) \leq 60 \text{ secondes}$$

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:

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

$BS_{\text{oh}} = (1/750) \text{ secondes} * R_{\max}[\text{profil, niveau}]$, avec $VBV_{\max}[\text{profil, niveau}]$ et $R_{\max}[\text{profil, niveau}]$ étant respectivement la capacité maximale de la mémoire VBV et le débit binaire 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:

$$BS_n = VBV_{\max} + BS_{\text{oh}}$$

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

$$BS_n = 2848 \text{ octets}$$

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).

2.5.3 Spécification de la syntaxe du flux de programme

La syntaxe suivante décrit un flux d'octets.

2.5.3.1 Flux de programme

Voir le Tableau 2-31.

Tableau 2-31 – Flux de programme

Syntaxe	Nombre de bits	Mnémonique
<pre> MPEG2_program_stream() { do { pack() } while (nextbits() == pack_start_code) MPEG_program_end_code } </pre>	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 0001 1011 1001' (0x000001B9). Il termine le flux de programme.

2.5.3.3 Couche bloc de paquets dans un flux de programme

Voir les Tableaux 2-32 et 2-33.

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(i) 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 rejeté 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 le Tableau 2-34.

Tableau 2-32 – Bloc de paquets du flux de programme

Syntaxe	Nombre de bits	Mnémonique
<pre>pack() { pack_header() while (nextbits() == packet_start_code_prefix) { PES_packet() } }</pre>		

Tableau 2-33 – En-tête de bloc de paquets de flux de programme

Syntaxe	Nombre de bits	Mnémonique
<pre>pack_header() { pack_start_code '01' system_clock_reference_base [32..30] marker_bit system_clock_reference_base [29..15] marker_bit system_clock_reference_base [14..0] marker_bit system_clock_reference_extension marker_bit program_mux_rate marker_bit marker_bit reserved pack_stuffing_length for (i = 0; i < pack_stuffing_length; i++) { stuffing_byte } if (nextbits() == system_header_start_code) { system_header() } }</pre>	<p>32</p> <p>2</p> <p>3</p> <p>1</p> <p>15</p> <p>1</p> <p>15</p> <p>1</p> <p>9</p> <p>1</p> <p>22</p> <p>1</p> <p>1</p> <p>5</p> <p>3</p> <p>8</p>	<p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p>

Tableau 2-34 – En-tête système de flux de programme

Syntaxe	Nombre de bits	Mnémonique
<pre>system_header() { system_header_start_code header_length marker_bit rate_bound marker_bit audio_bound fixed_flag CSPS_flag system_audio_lock_flag system_video_lock_flag marker_bit video_bound packet_rate_restriction_flag reserved_bits while (nextbits() == '1') { stream_id '11' P-STD_buffer_bound_scale P-STD_buffer_size_bound } }</pre>	<p>32</p> <p>16</p> <p>1</p> <p>22</p> <p>1</p> <p>6</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>5</p> <p>1</p> <p>7</p> <p>8</p> <p>2</p> <p>1</p> <p>13</p>	<p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p>

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 de ce 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 binaire est constant. Si elle est mise à '0', elle indique que le débit binaire est variable. En cas de fonctionnement à débit binaire 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:

$$SCR_{base}(i) = ((c1 * i + c2) DIV 300) \% 2^{33} \tag{2-22}$$

$$SCR_{ext}(i) = ((c1 * i + c2) DIV 300) \% 300 \tag{2-23}$$

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.

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_P-STD} \tag{2-24}$$

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 fréquence d'image et fréquence système) – Ce fanion indique s'il existe une relation spécifiée de rapport constant entre la fréquence d'image 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 la Rec. UIT-T H.262 | ISO/CEI 13818-2 spécifie la fréquence de trame vidéo. Le fanion de verrouillage du rapport entre fréquence d'image 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 vidéo d'un flux de programme conforme à la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, le rapport entre la fréquence système et la fréquence de trame réelle, SCFR, est constant et égal à la valeur indiquée dans le tableau suivant, à la fréquence nominale de trame qui est indiquée dans le flux de données vidéo.

$$SCFR = \frac{\text{system_clock_frequency}}{\text{rame_rate_in_the_P-STD}} \quad (2-25)$$

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) – Codé sur 5 éléments binaires, ce champ indique une valeur entière comprise entre 0 et 16 et qui est supérieure ou égale au nombre maximal de flux vidéo conformes à la Rec. UIT-T H.262 | ISO/CEI 13818-2 et ISO/CEI 11172-2, et multiplexés 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 conforme à la Rec. UIT-T H.262 | ISO/CEI 13818-2 et ISO/CEI 11172-2 est actif si la mémoire tampon du décodeur P-STD n'est pas vide ou si l'unité de présentation est en cours de présentation dans le modèle P-STD ou si le tampon de réordonnement des images n'est pas vide.

packet_rate_restriction_flag (fanion de restriction de débit de paquet) – Le fanion de restriction de débit binaire 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) – Cet octet est réservé pour usage futur par l'ISO/CEI. Sauf spécification contraire par l'UIT-T | ISO/CEI, cet octet doit avoir la valeur '1111 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 2-18.

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:

$$\text{si } (P\text{-STD_buffer_bound_scale} == 0), BS_n \leq P\text{-STD_buffer_size_bound} \times 128$$

autre

$$BS_n \leq P\text{-STD_buffer_size_bound} \times 1024$$

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 Conditionnement du flux de programme

Le conditionnement du flux de programme (PSM) 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 conditionnement de flux de programme (PSM) est présent sous forme de paquet PES lorsque la valeur d'identification du flux est 0xBC (voir le Tableau 2-18).

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 le Tableau 2-35.

Tableau 2-35 – Conditionnement du flux de programme

Syntaxe	Nombre de bits	Mnémonique
<pre> program_stream_map() { packet_start_code_prefix map_stream_id program_stream_map_length current_next_indicator reserved program_stream_map_version reserved marker_bit program_stream_info_length for (i = 0; i < N; i++) { descriptor() } elementary_stream_map_length for (i = 0; i < N1; i++) { stream_type elementary_stream_id elementary_stream_info_length for (i = 0; i < N2; i++) { descriptor() } } CRC_32 } </pre>	<p>24</p> <p>8</p> <p>16</p> <p>1</p> <p>2</p> <p>5</p> <p>7</p> <p>1</p> <p>16</p> <p>16</p> <p>8</p> <p>8</p> <p>16</p> <p>32</p>	<p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>rpchof</p>

2.5.4.2 Définition sémantique des champs contenus dans la structure program_stream_map

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 conditionner) – 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 conditionnement 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 conditionnement de flux de programme) – Codé sur 5 éléments binaires, ce champ indique le numéro de version du conditionnement complet du flux de programme. Ce numéro de version doit être augmenté de 1 modulo 32 chaque fois que la définition du conditionnement de flux de programme change. Lorsque le champ current_next_indicator est mis à '1', la version de conditionnement 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 conditionnement 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 conditionnement 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 2-29. 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 conditionnement 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 2-36.

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.

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 le Tableau 2-36.

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 le Tableau 2-18).

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 indiquer 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 le Tableau 2-37).

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 2-36 – 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
}		
}		

Tableau 2-37 – Indicateur de codage non prédictif

Valeur	Signification
0	Codage prédictif
1	Codage non prédictif

Tableau 2-38 – 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 2-39 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.

Tableau 2-39 – 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	X	maximum bitrate descriptor
15	X	X	private data indicator descriptor
16	X	X	smoothing buffer descriptor
17	X	X	STD_descriptor
18	X	X	IBP descriptor
19-63	n/a	n/a	Rec. UIT-T H.222.0 ISO/CEI 13818-1 Réservé
64-255	n/a	n/a	Utilisateur privé

descriptor_length (longueur de descripteur) – Codé sur 1 octet, ce champ spécifie le nombre d'octets du descripteur qui le suit immédiatement.

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 le Tableau 2-40).

Tableau 2-40 – 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 2-41.

Tableau 2-41 – 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 ISO/CEI 13818-2 que des données de flux paramétriques contraints conformes à 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 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 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 éléments binaires, 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.

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 ISO/CEI 13818-3 ou dans ISO/CEI 11172-3 (voir le Tableau 2-42).

Tableau 2-42 – Descripteur de flux de données audio

Syntaxe	Nombre de bits	Mnémonique
<pre>audio_stream_descriptor() { descriptor_tag descriptor_length free_format_flag ID layer variable_rate_audio_indicator reserved }</pre>	<p>8 8 1 1 2 1 3</p>	<p>uimsbf uimsbf bslbf bslbf bslbf bslbf bslbf</p>

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 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 ISO/CEI 13818-3).

layer (couche) – Codé sur deux éléments binaires, ce champ est mis à la même valeur que les champs indicateurs de couche du flux audio (voir 2.4.2.3 de 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 le débit du flux audio associé peut varier entre trames consécutives. Les flux audio de même codage doivent être présentés 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 données vidéo et audio à codage hiérarchique, ainsi que des flux privés multiplexés en flux complexes comme décrit dans la présente Recommandation | Norme internationale, dans la Rec. UIT-T H.262 | ISO/CEI 13818-2 et dans ISO/CEI 13818-3 (voir le Tableau 2-43).

Tableau 2-43 – Descripteur de hiérarchie

Syntaxe	Nombre de bits	Mnémonique
<pre> hierarchy_descriptor() { descriptor_tag descriptor_length reserved hierarchy_type reserved hierarchy_layer_index reserved hierarchy_embedded_layer_index reserved hierarchy_channel } </pre>	<p>8 8 4 4 2 6 2 6 2 6</p>	<p>uimsbf uimsbf bslbf uimsbf bslbf uimsbf bslbf uimsbf bslbf uimsbf</p>

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 2-44.

Tableau 2-44 – Valeurs du descripteur de hiérarchie

Valeur	Description
0	Valeur réservée
1	Rec. UIT-T H.262 ISO/CEI 13818-2 (échelonnabilité spatiale)
2	Rec. UIT-T H.262 ISO/CEI 13818-2 (échelonnabilité SNR)
3	Rec. UIT-T H.262 ISO/CEI 13818-2 (échelonnabilité temporelle)
4	Rec. UIT-T H.262 ISO/CEI 13818-2 (subdivision des données)
5	ISO/CEI 13818-3 (flux binaire audio d'extension)
6	Rec. UIT-T H.222.0 ISO/CEI 13818-1 (flux de données privées)
7-14	Valeur réservée
15	Couche de base

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 encastrée) – 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.

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 le Tableau 2-45).

Tableau 2-45 – Descripteur d'enregistrement

Syntaxe	Nombre de bits	Identificateur
<pre> registration_descriptor() { descriptor_tag descriptor_length format_identifieur for (i = 0; i < N; i++) { additional_identification_info } } </pre>	<p>8</p> <p>8</p> <p>32</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p>

2.6.9 Définition sémantique des champs dans un descripteur d'enregistrement

format_identifieur (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_identifieur** 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 le Tableau 2-46).

Tableau 2-46 – Descripteur d'alignement du flux de données

Syntaxe	Nombre de bits	Mnémonique
<pre> data_stream_alignment_descriptor() { descriptor_tag descriptor_length alignment_type } </pre>	<p>8</p> <p>8</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>

2.6.11 Définition sémantique des champs dans un descripteur d'alignement de flux de données

alignment_type (type d'alignement) – Le Tableau 2-47 décrit le type d'alignement de données vidéo lorsque l'indicateur d'alignement de données a la valeur '1' dans l'en-tête de paquet PES. Quelle que soit la valeur de codage du champ **alignment_type**, le premier octet de données de paquet PES suivant l'en-tête de paquet PES, est le premier octet d'un code de déclenchement du type indiqué au Tableau 2-47. Au début d'une séquence vidéo l'alignement doit être observé au niveau du code de déclenchement du premier en-tête de séquence.

NOTE – Un type d'alignement '01' dans le Tableau 2-47 n'exclut pas un alignement de départ GOP ou SEQ.

La définition de l'unité d'accès pour données vidéo figure au 2.1.1.

Tableau 2-47 – 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 2-48 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 2-48 – 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.

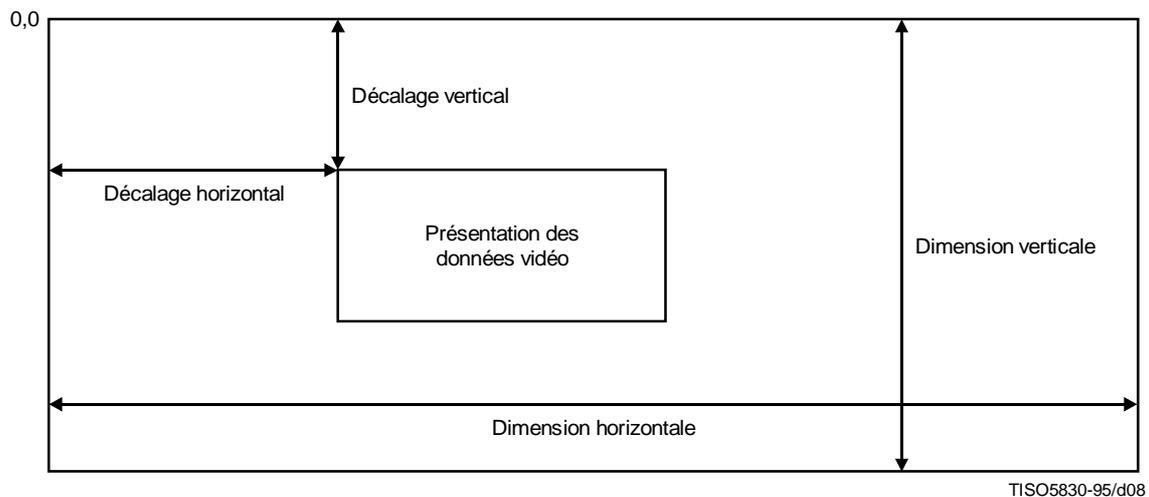


Figure 2-3 – Zone d'affichage par descripteur de la grille de fond réceptrice

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 le Tableau 2-49).

Tableau 2-49 – Descripteur de grille sur fond réceptrice

Syntaxe	Nombre de bits	Mnémonique
<pre>target_background_grid_descriptor() { descriptor_tag descriptor_length horizontal_size vertical_size aspect_ratio_information }</pre>	<p style="text-align: center;">8 8 14 14 4</p>	<p style="text-align: center;">uimsbf uimsbf uimsbf uimsbf uimsbf</p>

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 le Tableau 2-50).

Tableau 2-50 – Descripteur de fenêtre vidéo

Syntaxe	Nombre de bits	Mnémonique
<pre>video_window_descriptor() { descriptor_tag descriptor_length horizontal_offset vertical_offset window_priority }</pre>	<p style="text-align: center;">8 8 14 14 4</p>	<p style="text-align: center;">uimsbf uimsbf uimsbf uimsbf uimsbf</p>

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 la 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 la 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/ou 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 le Tableau 2-51).

Tableau 2-51 – Descripteur d'accès conditionnel

Syntaxe	Nombre de bits	Mnémonique
<pre>CA_descriptor() { descriptor_tag descriptor_length CA_system_ID reserved CA_PID for (i = 0; i < N; i++) { private_data_byte } }</pre>	<p>8</p> <p>8</p> <p>16</p> <p>3</p> <p>13</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p>

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/ou 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 conditionnement 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.

2.6.18 Descripteur de code de langue selon ISO 639

Le descripteur de langue sert à spécifier la langue utilisée dans le flux élémentaire associé (voir le Tableau 2-52).

Tableau 2-52 – Descripteur de code de langue selon ISO 639

Syntaxe	Nombre de bits	Mnémonique
<pre>ISO_639_language_descriptor() { descriptor_tag descriptor_length for (i = 0; i < N; i++) { ISO_639_language_code audio_type } }</pre>	<p>8</p> <p>8</p> <p>24</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p>

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) langue(s) utilisée(s) 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 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 2-53.

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.

Tableau 2-53 – Valeurs de type de données audio

Valeur	Description
0x00	Non défini
0x01	Effets d'écoute
0x02	Malentendants
0x03	Commentaire pour malvoyants
0x04 - 0xFF	Valeur réservée

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:

$$\text{clock_accuracy_integer} \times 10^{-\text{clock_accuracy_exponent}} \text{ ppm} \tag{2-26}$$

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 le Tableau 2-54).

Tableau 2-54 – Descripteur d'horloge de base du système

Syntaxe	Nombre de bits	Mnémonique
<code>system_clock_descriptor() {</code>		
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
<code>}</code>		

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 le Tableau 2-55).

Tableau 2-55 – Descripteur d'utilisation du tampon de multiplexage

Syntaxe	Nombre de bits	Mnémonique
<pre> multiplex_buffer_utilization_descriptor() { descriptor_tag descriptor_length bound_valid_flag LTW_offset_lower_bound reserved LTW_offset_upper_bound } </pre>	<p>8</p> <p>8</p> <p>1</p> <p>15</p> <p>1</p> <p>14</p>	<p>uimbsf</p> <p>uimbsf</p> <p>bslbf</p> <p>uimbsf</p> <p>bslbf</p> <p>uimbsf</p>

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 le Tableau 2-56).

Tableau 2-56 – Descripteur de droit d'auteur

Syntaxe	Nombre de bits	Identificateur
<pre> copyright_descriptor() { descriptor_tag descriptor_length copyright_identifier for (i = 0; i < N; i++) { additional_copyright_info } } </pre>	<p>8</p> <p>8</p> <p>32</p> <p>8</p>	<p>uimbsf</p> <p>uimbsf</p> <p>uimbsf</p> <p>bslbf</p>

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 binaire maximal

Voir le Tableau 2-57.

Tableau 2-57 – Descripteur de débit binaire maximal

Syntaxe	Nombre de bits	Identificateur
<pre> maximum_bitrate_descriptor() { descriptor_tag descriptor_length reserved maximum_bitrate } </pre>	<p>8</p> <p>8</p> <p>2</p> <p>22</p>	<p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p>

2.6.27 Définition sémantique des champs dans un descripteur de débit binaire maximal

maximum_bitrate (débit binaire maximal) – Codé sur 22 éléments binaires, ce champ contient un entier positif qui indique la limite supérieure du débit binaire 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 binaire maximal est donné dans la table de conditionnement de programme (PMT). 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 le Tableau 2-58.

Tableau 2-58 – Descripteur d'indication de données privées

Syntaxe	Nombre de bits	Identificateur
<pre> private_data_indicator_descriptor() { descriptor_tag descriptor_length private_data_indicator } </pre>	<p>8</p> <p>8</p> <p>32</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>

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 correspondant(s).

Dans le cas de flux de transport, les octets des paquets de transport de l'élément ou des éléments de programme associé(s) présent(s) 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 conditionnement de programme ou dans la table de conditionnement 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 conditionnement 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 conditionnement 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 2-59 – Descripteur de mémoire de lissage

Syntaxe	Nombre de bits	Mnémonique
smoothing_buffer_descriptor() { descriptor_tag descriptor_length reserved sb_leak_rate reserved sb_size }	8 8 2 22 2 22	uimsbf uimsbf bslbf uimsbf bslbf 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 le Tableau 2-59).

2.6.32 Descripteur STD

Ce descripteur facultatif s'applique uniquement au modèle théorique T-STD et aux flux élémentaires vidéo; ses modalités d'utilisation sont spécifiées au 2.4.2. Ce descripteur ne s'applique pas aux flux de programme (voir le Tableau 2-60).

Tableau 2-60 – Descripteur de modèle STD

Syntaxe	Nombre de bits	Mnémonique
STD_descriptor() { descriptor_tag descriptor_length reserved leak_valid_flag }	8 8 7 1	uimsbf uimsbf bslbf 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 dans la séquence vidéo (voir le Tableau 2-61).

Tableau 2-61 – Descripteur IBP

Syntaxe	Nombre de bits	Mnémonique
<pre> ibp_descriptor() { descriptor_tag descriptor_length closed_gop_flag identical_gop_flag max_gop_length } </pre>	<p style="text-align: center;"> 8 8 1 1 14 </p>	<p style="text-align: center;"> uimsbf uimsbf uimsbf uimsbf uimsbf </p>

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.

2.7 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:

$$|t(i) - t(i')| \leq 0,7 \text{ s}$$

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:

$$|t(i) - t(i')| \leq 0,1 \text{ s}$$

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:

$$|t(i) - t(i')| \leq 0,7 \text{ s}$$

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:

$$|tp_n(k) - tp_n(k'')| \leq 0,7 \text{ s}$$

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 $k < k' < k''$; et
- aucune discontinuité de décodage n'existe dans le flux élémentaire n entre $P_n(k)$ et $P_n(k'')$.

Dans le cas des images fixes, la contrainte liée à l'observation de 0,7 s n'est pas applicable.

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) 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) 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.

2.7.6 Contraintes de synchronisation applicables au décodage échelonné

Si une séquence audio est codée au moyen d'un flux binaire d'extension selon ISO/CEI 13818-3, les unités correspondantes de décodage/présentation dans les deux couches doivent avoir des valeurs identiques d'horodateur PTS.

Si une séquence vidéo est codée en tant qu'amélioration SNR d'une autre séquence, comme indiqué au 7.8 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les deux séquences doivent comporter deux séries identiques d'instant de présentation.

Si une séquence vidéo est codée comme deux partitions, comme indiqué au 7.10 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les deux séquences doivent comporter deux séries identiques d'instant de présentation.

Si une séquence vidéo est codée en tant qu'amélioration spatiale échelonnée d'une autre séquence, comme indiqué 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'instant 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'instant de présentation doivent comporter un nombre aussi grand que possible d'instant 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 simultanée; ou
 - 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 spatiale échelonnée d'une autre séquence, comme indiqué au 7.9 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, les images désignées ci-dessous de la couche inférieure peuvent servir de référence. On notera que le temps est mesuré par rapport aux instant de présentation:

- image simultanée ou présentée le plus récemment de la couche inférieure;
- 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) 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 QUIPOS, 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 binaires plus élevés, le débit de paquets QUIPOS 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',

$$NP \leq (t(i') - t(i)) \times 300 \times \max \left[1, \frac{R_{\max}}{2,0 \times 10^6} \right] \quad (2-27)$$

et si le fanion de restriction de débit de paquets est mis à la valeur '0',

$$NP \leq (t(i') - t(i)) \times 300 \times \max \left[1, \frac{R_{\max}}{2,0 \times 10^6} \right] \quad (2-28)$$

où

$$R_{\max} = 8 \times 50 \times \text{rate_bound} \quad \text{bit/s} \quad (2-29)$$

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 2-31 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 CSPTS, 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 d'un CSPTS, ce qui suit s'applique.

La mémoire BS_n possède une capacité égale à la somme de la capacité du vérificateur de mémoire vidéo spécifié dans la Rec. UIT-T H.262 | ISO/CEI 13818-2 et d'une capacité mémoire supplémentaire, BS_{add} . Celle-ci est spécifiée comme suit:

$$BS_{add} \leq \text{MAX} [6 \times 1024, R_{v\max} \times 0,001] \text{ en octets}$$

où $R_{v\max}$ est le débit binaire crête des données vidéo.

Dans le cas d'un flux élémentaire de données audio intégré dans un flux CSPTS, la relation suivante est applicable:

$$BS_n \leq 4096 \text{ octets}$$

2.7.10 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 ISO/CEI 13818-3 ou dans 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.

$$SCASR = \frac{\text{system_clock_frequency}}{\text{audio_sample_rate_in_the_T-STD}} \quad (2-30)$$

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	$\frac{27\ 000\ 000}{16\ 000}$	$\frac{27\ 000\ 000}{32\ 000}$	$\frac{27\ 000\ 000}{22\ 050}$	$\frac{27\ 000\ 000}{44\ 100}$	$\frac{27\ 000\ 000}{24\ 000}$	$\frac{27\ 000\ 000}{48\ 000}$

Pour toutes les unités de présentation contenues dans tous les flux élémentaires de données vidéo d'un flux de transport, le rapport SCFR entre la fréquence d'horloge système et la fréquence de trame réelle est constant et égal à la valeur indiquée dans le tableau suivant, à la fréquence de trame nominale qui est indiquée dans le flux de données vidéo.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_T-STD} \tag{2-31}$$

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.

2.8 Compatibilité avec ISO/CEI 11172

Le flux de programme de la présente Recommandation | Norme internationale est défini de manière à être compatible par anticipation avec 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.

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 éléments binaires est spécifié dans la Figure A.1.

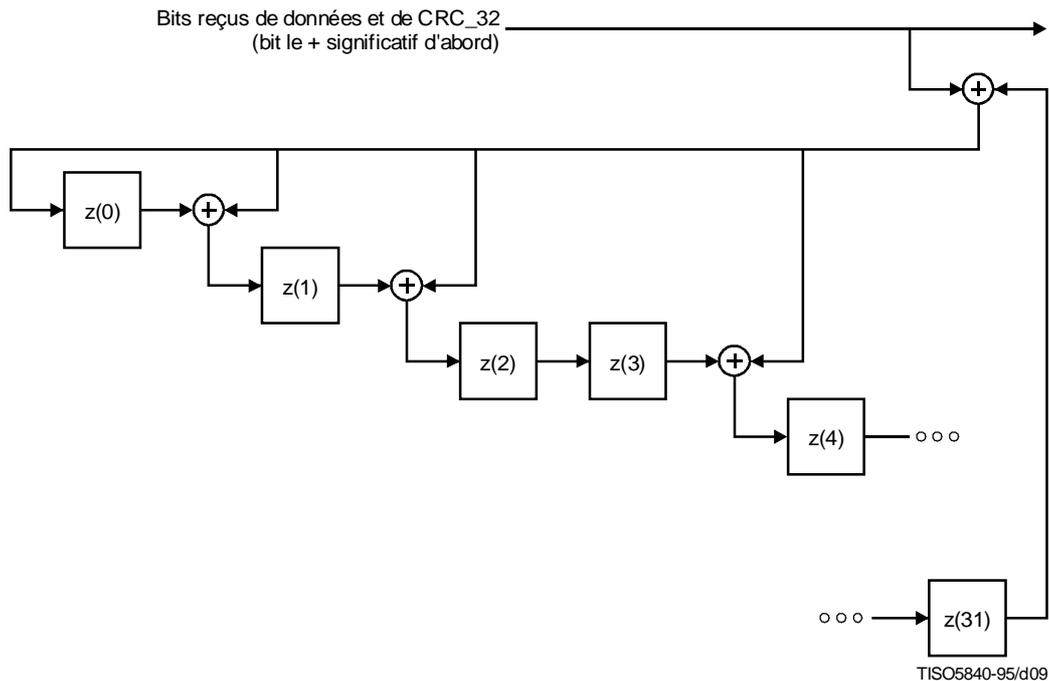


Figure A.1 – Décodeur modèle du contrôle CRC sur 32 bits

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:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (\text{A-1})$$

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 fort (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 basses 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 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 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 usagers 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 usagers la garantie que les flux binaires conformes à 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 (la correspondance 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 cette 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) 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 Vue d'ensemble 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.

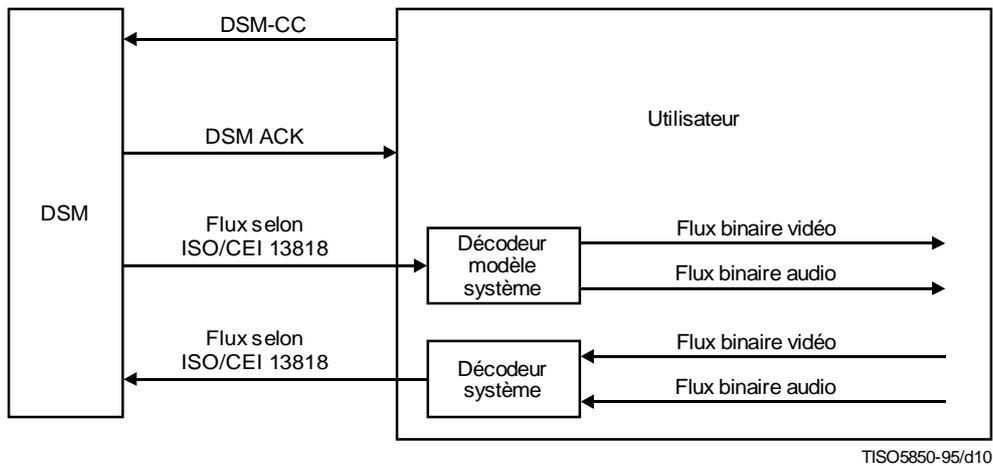


Figure B.1 – Configuration d'une application de commande et DSM-CC

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 encadré 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.

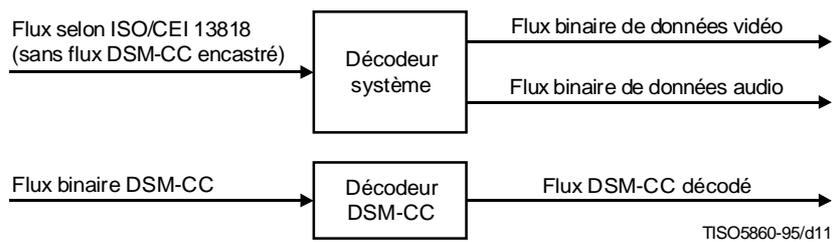


Figure B.2 – Flux de commande DSM-CC décodé sous forme de flux binaire autonome

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 encadré 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/ou 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 2-18. Une fois le flux DSM-CC identifié, il suit les règles spécifiées par T-STD ou P-STD.

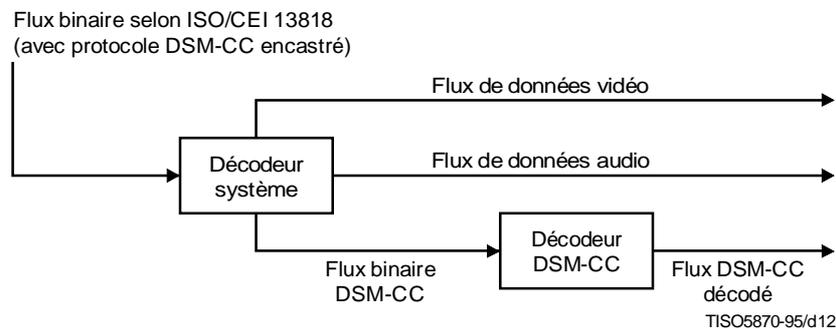


Figure B.3 – Flux de commande DSM-CC décodé sous forme d'élément du flux système

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'acquiescement (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 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/ou 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-CCC.
- 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'acquiescement proprement dite doit suivre le champ packet_length.
- Un flux de données d'acquiescement 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
<pre> DSM_CC() { packet_start_code_prefix stream_id packet_length command_id if (command_id == '01') { control() } else if (command_id == '02') { ack() } } </pre>	<p>24</p> <p>8</p> <p>16</p> <p>8</p>	<p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>

B.2.3 Sémantique des champs contenus dans la spécification syntaxique du protocole DSM-CC

packet_start_code_prefix – Champ codé sur 24 éléments binaires; 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 éléments binaires spécifiant l'identification du flux binaire; il prend la valeur '1111 0010' dans le cas du flux binaire DSM-CC.

packet_length – Champ codé sur 16 éléments binaires 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 acquiescement. 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	Acquiescement
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), 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 le 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 éléments binaires, 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 éléments binaires, 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 mettre en correspondance 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 éléments binaires, 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 usager 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'acquiescement

Contraintes sur l'activation des fanions dans une commande DSM-CC

Pour chaque flux binaire d'acquiescement de commande DSM, un seul des fanions de sélection, d'extraction, d'enregistrement et d'erreur doit être sélectionné (mis à '1') (voir le Tableau B.5).

Tableau B.5 – Acquiescement de commande DSM-CC

Syntaxe	Nombre de bits	Mnémonique
<pre>ack() { select_ack retrieval_ack storage_ack error_ack reserved marker_bit cmd_status if (cmd_status == '1' && (retrieval_ack == '1' storage_ack == '1')) { time_code() } }</pre>	<p>1 1 1 1 10 1 1</p>	<p>bslbf bslbf bslbf bslbf bslbf bslbf bslbf</p>

B.2.7 Sémantique des champs contenus dans la couche d'acquiescement

select_ack (acquiescement 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 (acquiescement 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 (acquiescement 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 (acquiescement 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; sinon, 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 acquiescement 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 acquiescement d'extraction, l'opération a échoué. Les raisons possibles de cet échec sont par exemple un identificateur de flux invalide, 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 invalide 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 le Tableau B.6.

Tableau B.6 – Code temporel

Syntaxe	Nombre de bits	Mnémonique
<code>time_code() {</code>		
reserved	7	bslbf
infinite_time_flag	1	bslbf
if (<code>infinite_time_flag == '0'</code>) {		
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). Cette annexe en développe l'utilisation.

On peut considérer les informations PSI comme appartenant à quatre tables:

- 1) la table d'association de programmes (PAT);
- 2) la table de conditionnement de programme (PMT) pour le flux de transport;
- 3) la table d'informations sur le réseau (NIT);
- 4) la table d'accès conditionnel (CAT).

Le contenu des tables PAT, PMT et CAT est spécifié dans la présente Recommandation | Norme internationale. La table NIT est privée mais la valeur d'identification des paquets de transport qui l'acheminement est spécifiée dans la table PAT. Ce contenu doit toutefois être conforme à la structure définie dans la présente Recommandation | Norme internationale pour les sections.

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 conditionnement de programme pour un flux de transport.

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 conditionnement 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 réordonner 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 rejetée) 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 Insertion 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 de rejeter 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 binaire des données PSI. Si des correspondances 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 2.1.42)]. 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).

L'insertion dans le flux de transport peut être effectuée 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 binaires 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 conditionnement 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). La présence d'une table d'informations sur le réseau (NIT) est entièrement facultative.

Les tables d'informations PSI sont insérées 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 insérées dans des paquets de table PAT ou CAT.

Il est requis que toutes les sections de table PAT soient insérées dans des paquets du flux de transport avec `PID = 0x0000` et que toutes les sections de table CAT soient insérées dans des paquets avec `PID = 0x0001`. Les sections de table PMT peuvent être insérées 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.

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 insérée 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 conditionnement de programme (PMT)

La table de conditionnement de programme donne la correspondance 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.

NOTES

- 1 Les paquets de flux de transport contenant la table PMT sont transmis sans embrouillage.
- 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 correspondances 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.

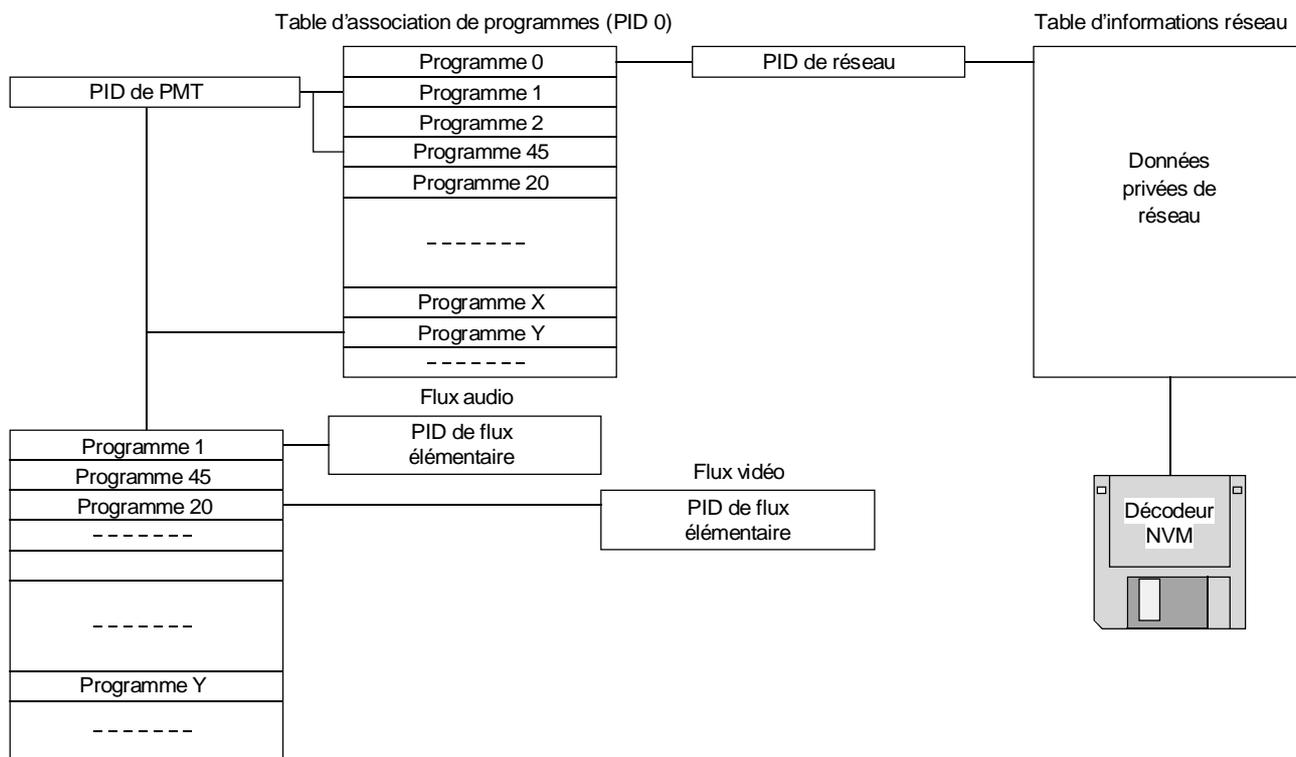


Table de conditionnement de programme (PMT)

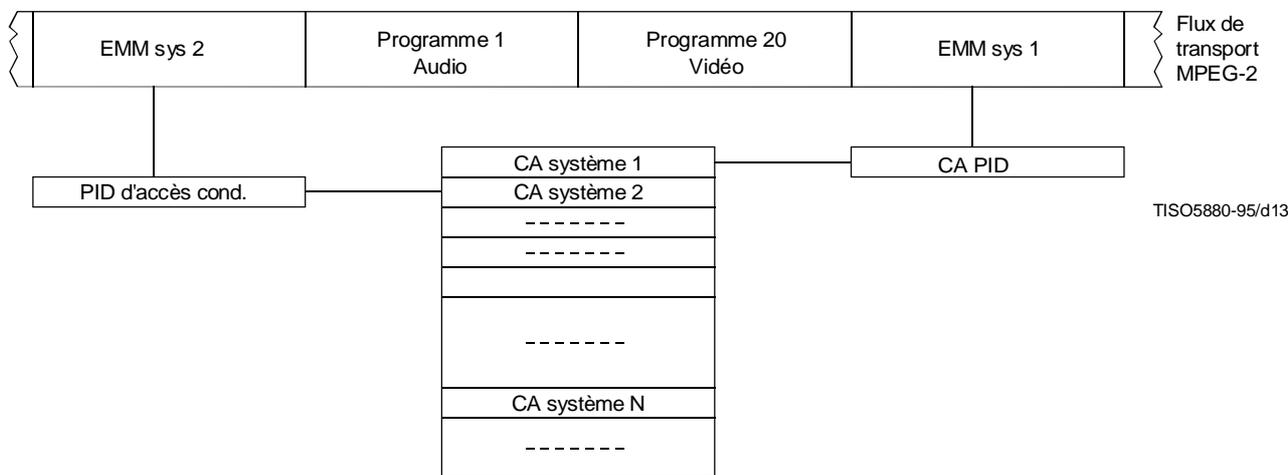


Table d'accès conditionnel (PID 1)

Figure C.1 – Relations bijectives entre programme et réseau

Tableau C.1 – Descripteur composite (élément composite_descriptor)

Syntaxe	Nombre de bits	Identificateur
<pre>Composite_descriptor() { descriptor_tag(privately defined) descriptor_length for (i = 0; i < N; i++) { sub_descriptor() } }</pre>	<p>8</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p>

Tableau C.2 – Descripteur secondaire

Syntaxe	Nombre de bits	Identificateur
<pre>sub_descriptor() { sub_descriptor_tag sub_descriptor_length for (i = 0; i < N; i++) { private_data_byte } }</pre>	<p>8</p> <p>8</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>

C.9 Affectation spectrale et temps d'acquisition du signal

Toute mise en œuvre 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. Ce 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 la répartition en fréquence (MRF), 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.

Ce paragraphe traite des incidences, en termes de débit binaire comme de temps d'acquisition du signal, des paragraphes 2.4.4.4 et 2.4.4.9 concernant la syntaxe des informations PSI. 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 ci-dessous 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.

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 conditionnement 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 conditionnement 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.

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 conditionnement 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

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

- 5 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 binaire 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 binaires 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 de piloter 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.

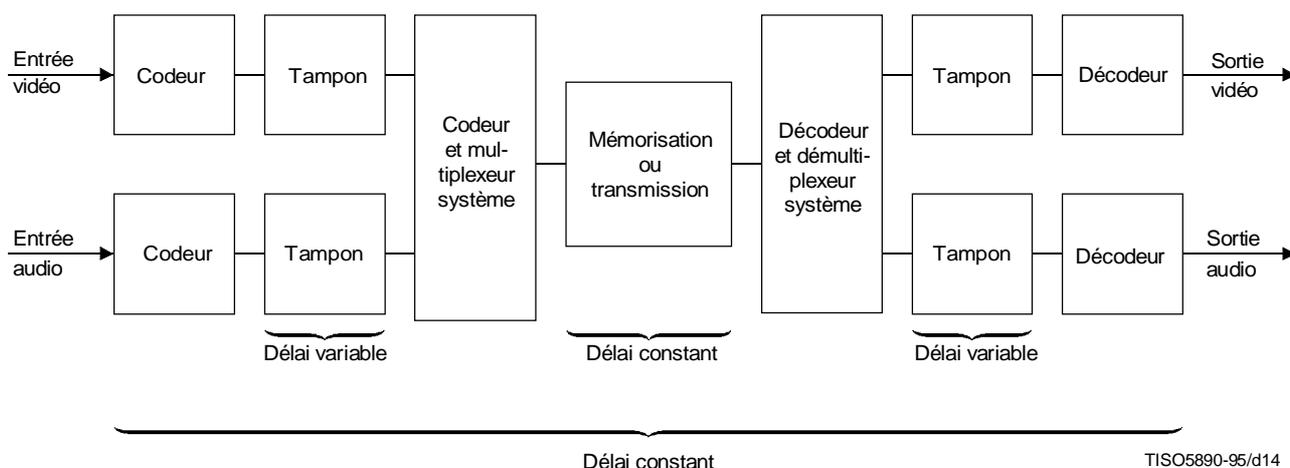


Figure D.1 – Modèle à délai constant

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 modèle²⁾, 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 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 applications 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 la mettre en œuvre.

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 codeur. 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 mis en œuvre 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 mis en œuvre 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 mises en œuvre, 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 mise en œuvre.

²⁾ 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.

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 AAU (unité d'accès audio) ou qu'une APU (unité de présentation audio), selon le contexte. Une unité de présentation vidéo (VPU) est une image décodée et une unité d'accès vidéo (VAU) 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 réalisations concrètes de décodeurs, afin de rendre sa spécification plus claire et d'en faciliter une large application 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 la mise en œuvre. 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 binaire 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 mise en œuvre 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 cette 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 réordonnement 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 réordonnement 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 réordonnement 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 réaliser 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 de ce 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 recalibrer 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 mettre en œuvre 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). 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 recalcule 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.

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.

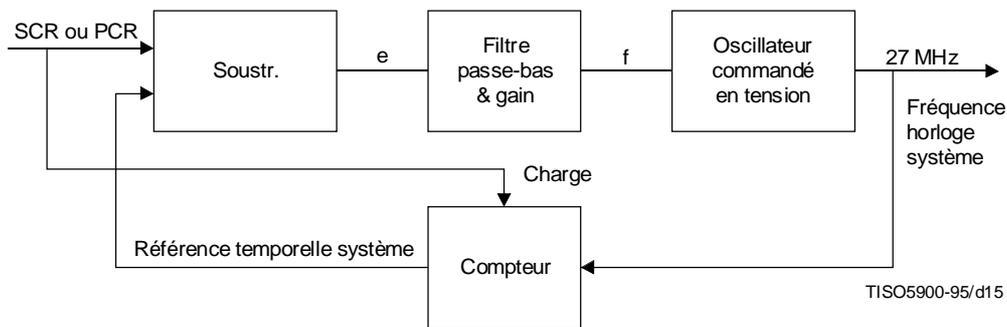


Figure D.2 – Recalage d'horloge STC par verrouillage de phase (PLL)

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). 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 + tôt	del_{min}
au + tard	del_{max}
au + 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 millièmes, 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 millionnièmes, ainsi que du pire cas de pente de stabilisation en fréquence, mesurée en millionnièmes 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), 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 applications 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) 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 de mise en œuvre de systèmes de décodage et de présentation.

Dans un certain type de mise en œuvre, 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 Considérations générales

- 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

Cette 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 la Figure F.1.

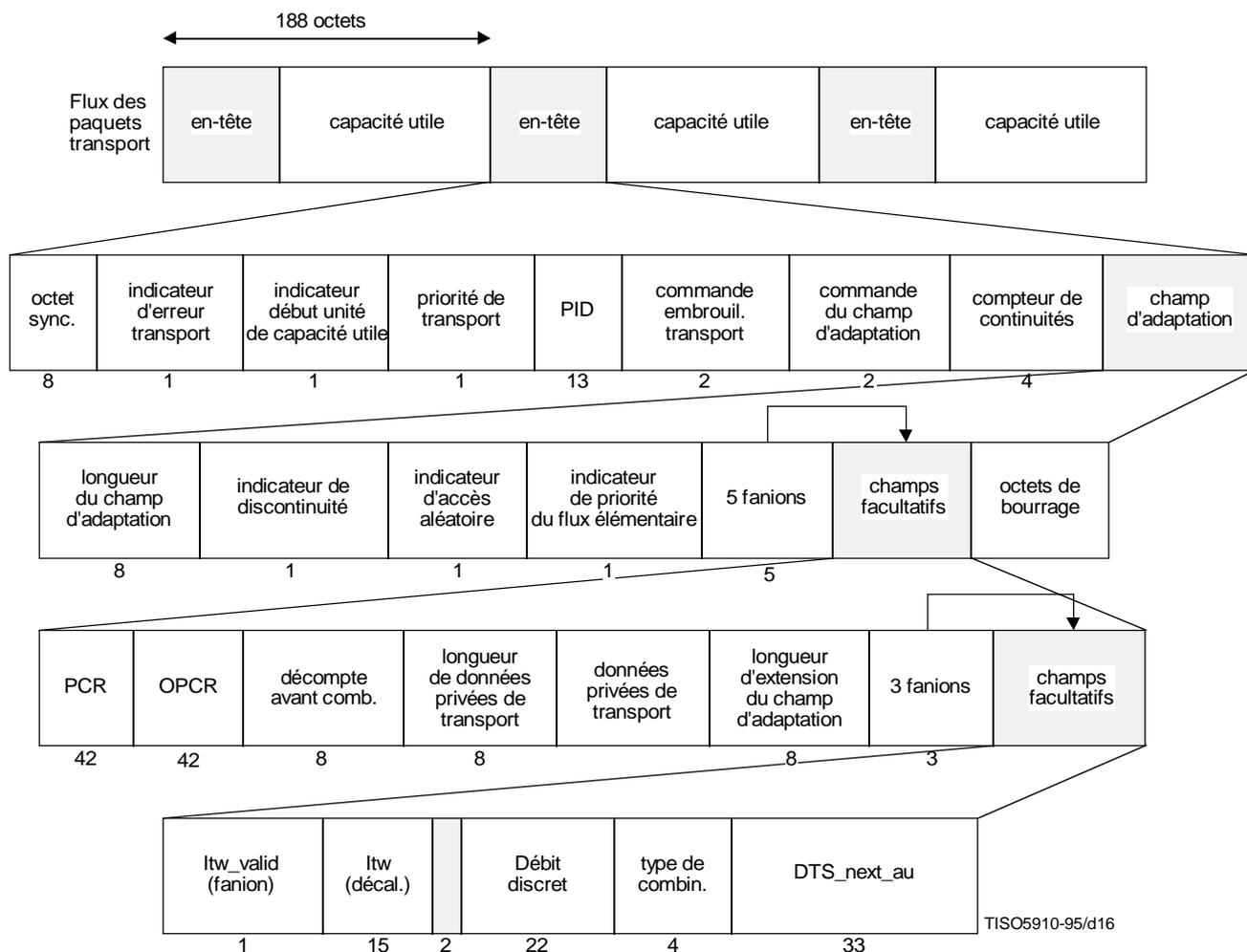


Figure F.1 – Diagramme syntaxique du flux de transport

F.0.2 Paquet de flux PES

Voir la Figure F.2.

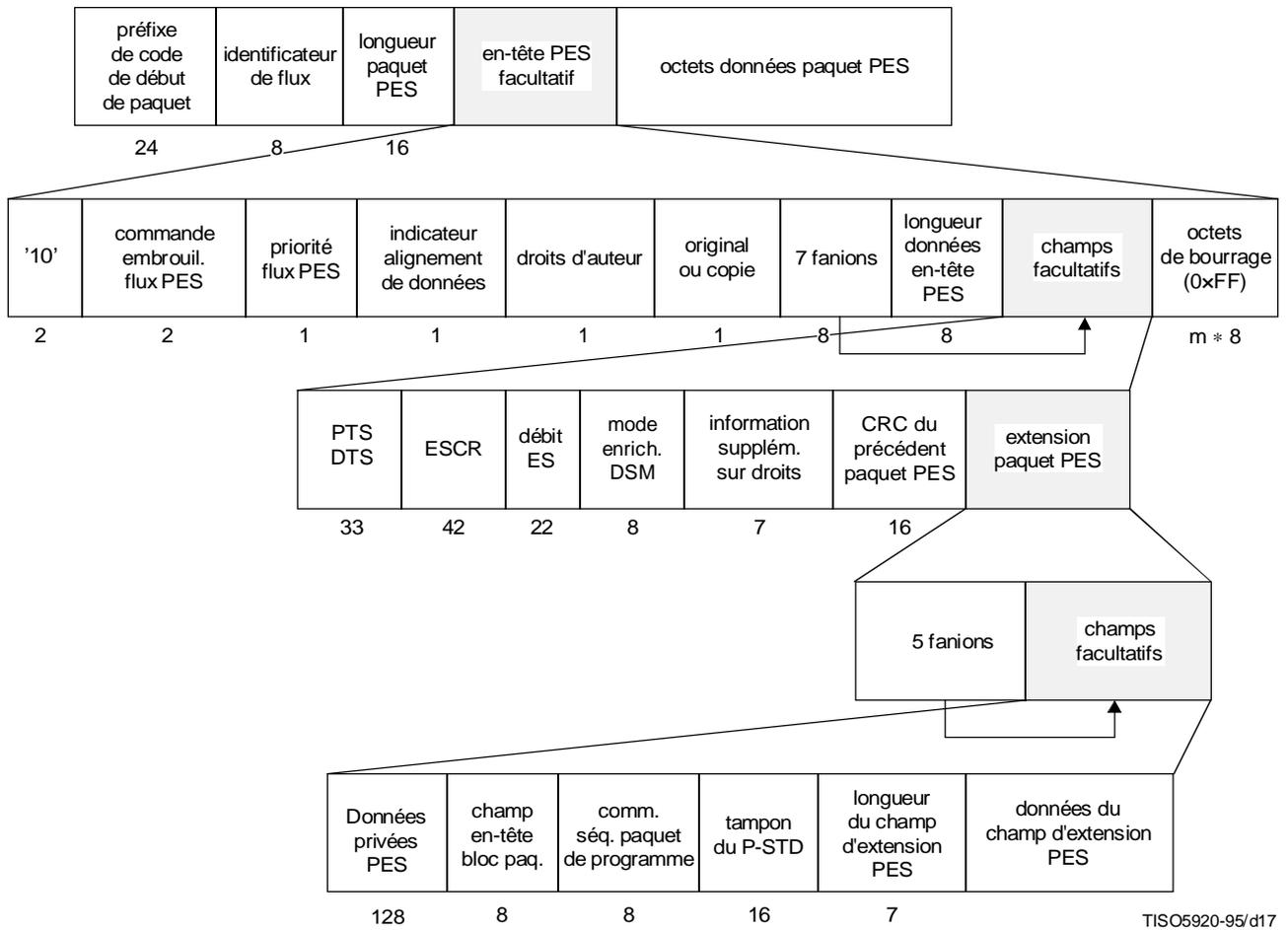


Figure F.2 – Diagramme syntaxique des paquets de flux PES

F.0.3 Section de table d'association de programmes

Voir la Figure F.3.

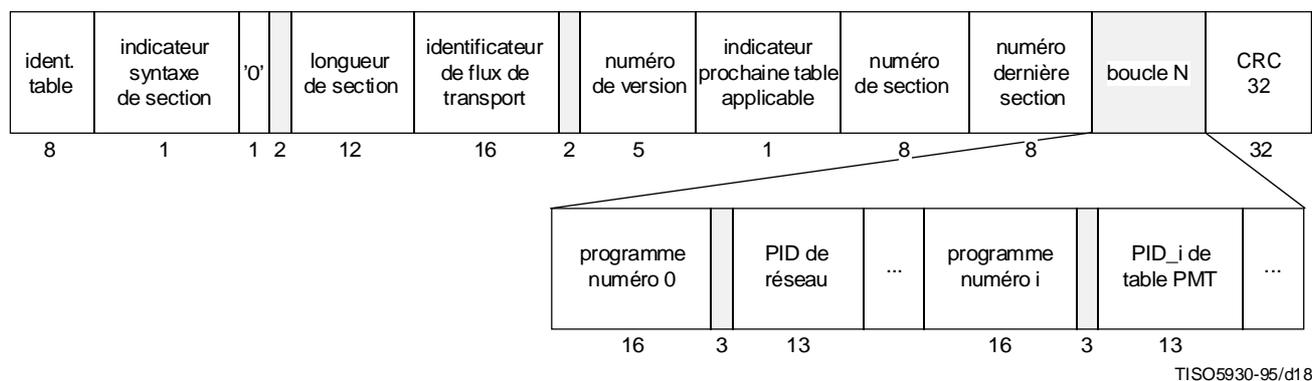


Figure F.3 – Diagramme de section de table d'association de programmes

F.0.4 Section de table d'accès conditionnel

Voir la Figure F.4.

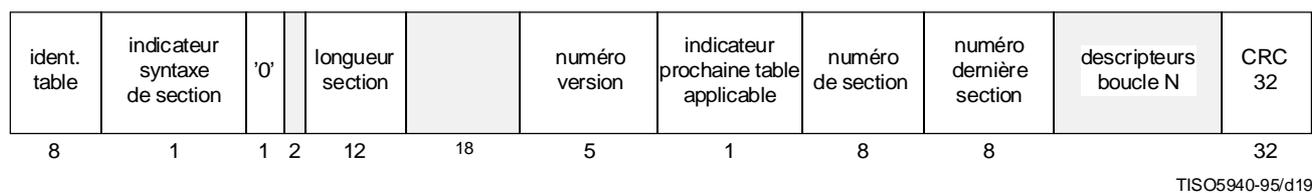


Figure F.4 – Diagramme de section de table d'accès conditionnel

F.0.5 Section de table de conditionnement de programme pour flux de transport

Voir la Figure F.5.

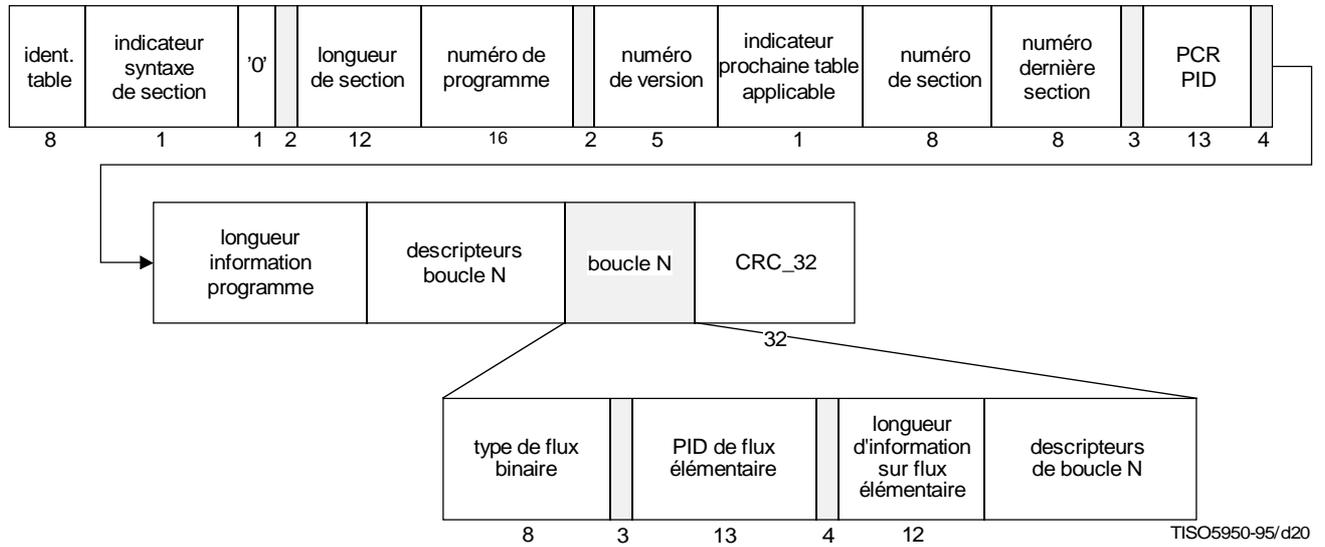


Figure F.5 – Diagramme de section de table de conditionnement de programme pour flux de transport

F.0.6 Section de table de données privées

Voir la Figure F.6.

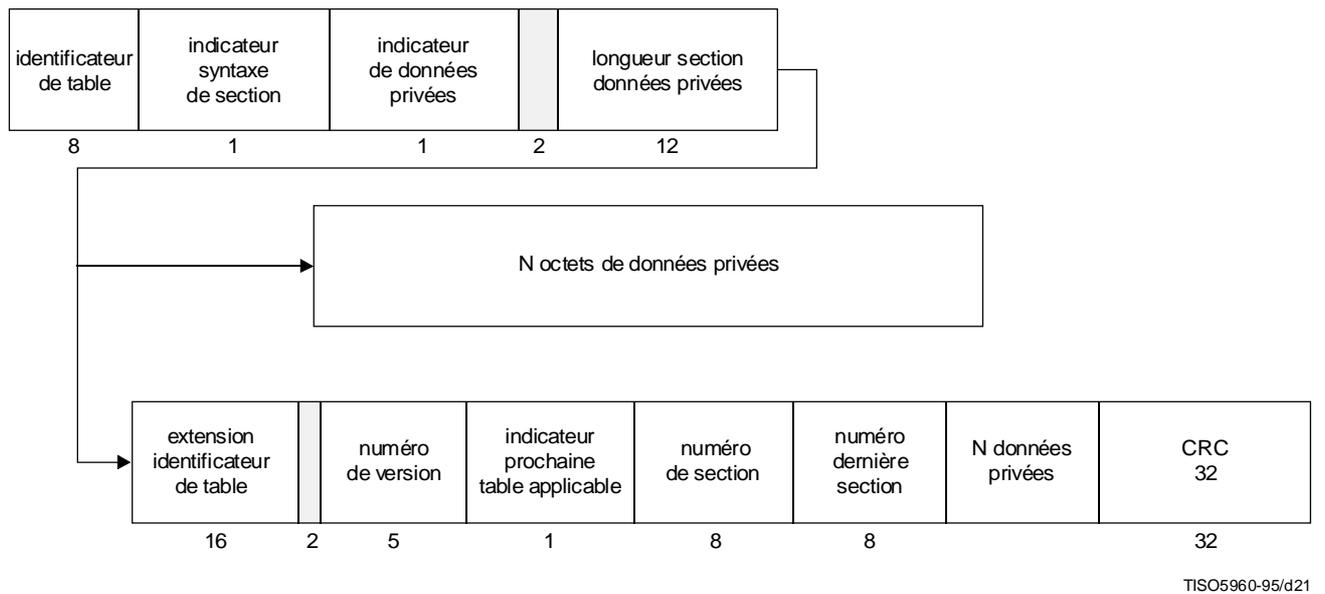
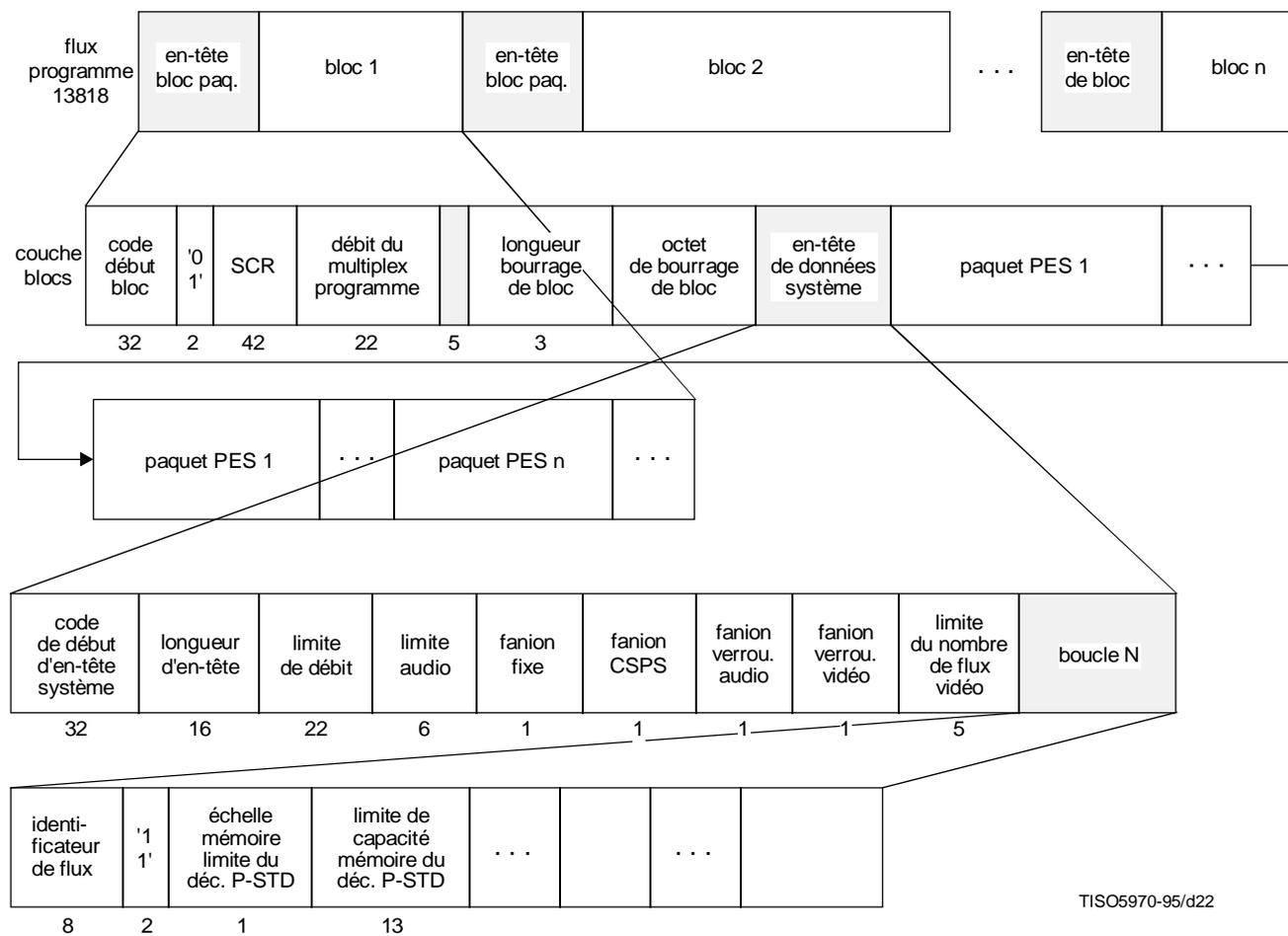


Figure F.6 – Diagramme de section de table de données privées

F.0.7 Flux de programme

Voir la Figure F.7.



TISO5970-95/d22

Figure F.7 – Diagramme du flux de programme

F.0.8 Conditionnement de flux de programme

Voir la Figure F.8.

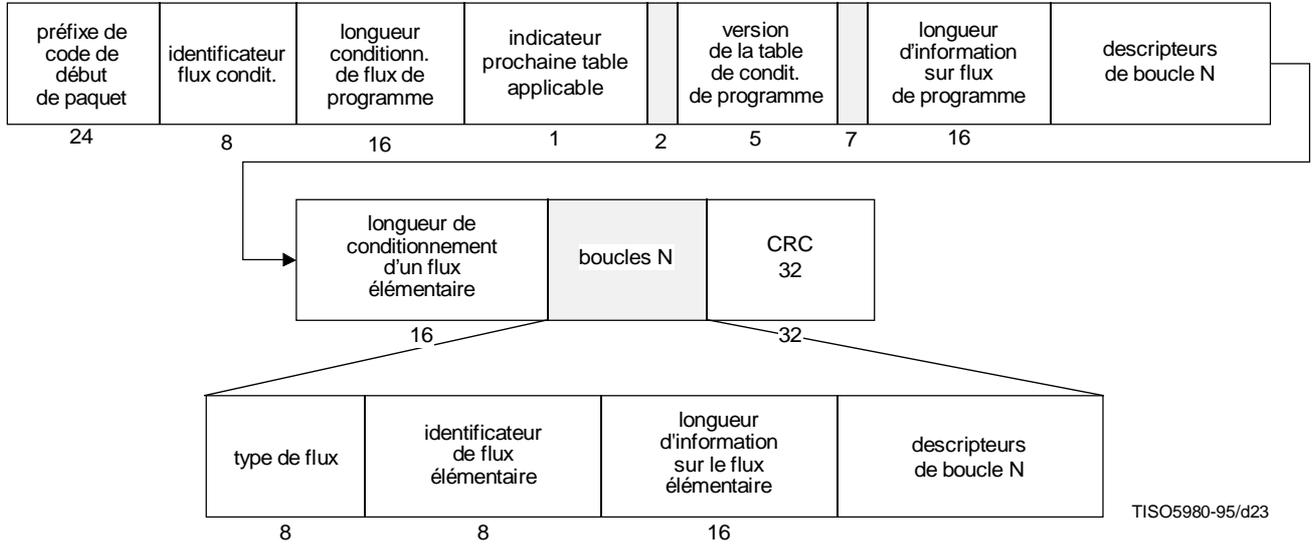


Figure F.8 – Diagramme du conditionnement de flux de programme

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 2-29 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 2-17 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 2-29 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 2-39. Ces descripteurs peuvent être placés à l'intérieur d'une table de conditionnement de flux de programme (Tableau 2-29), d'une section de table d'accès conditionnel (Tableau 2-27), d'une section de table de conditionnement de programme pour transport (Tableau 2-28) et dans toute section de table de données privées (Tableau 2-30).

Le descripteur d'accès conditionnel désigne également, de manière spécifique, des octets de données privées.

5) *Section de table de données privées*

Le Tableau 2-30 sur la section de données privées offre un moyen supplémentaire pour acheminer des données privées, de nouveau sous deux formes. Selon le Tableau 2-29 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 mises en œuvre 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.2 présente un cadre général pour la mise au point de modèles de conformité de réseau.

Le paragraphe J.3 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 binaire 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.4 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. 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.4.1. 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.4.2.

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), 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 décodé 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.

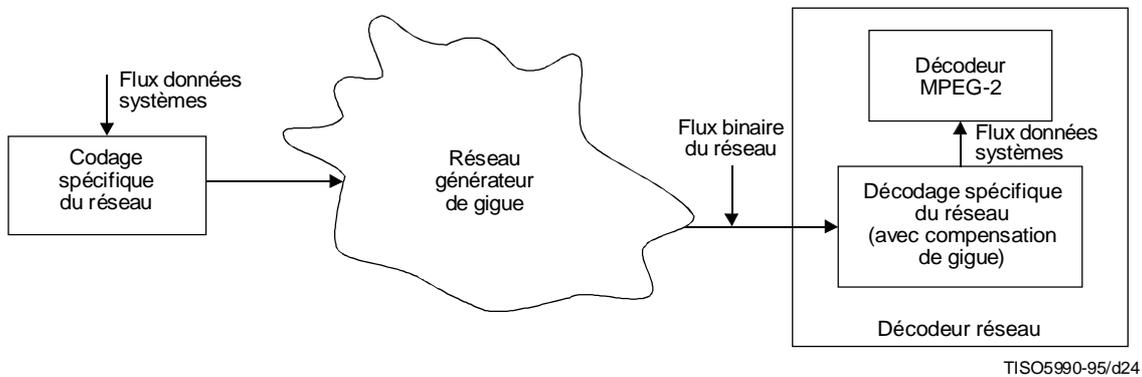


Figure J.1 – Envoi de flux systèmes sur un réseau générateur de gigue

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.

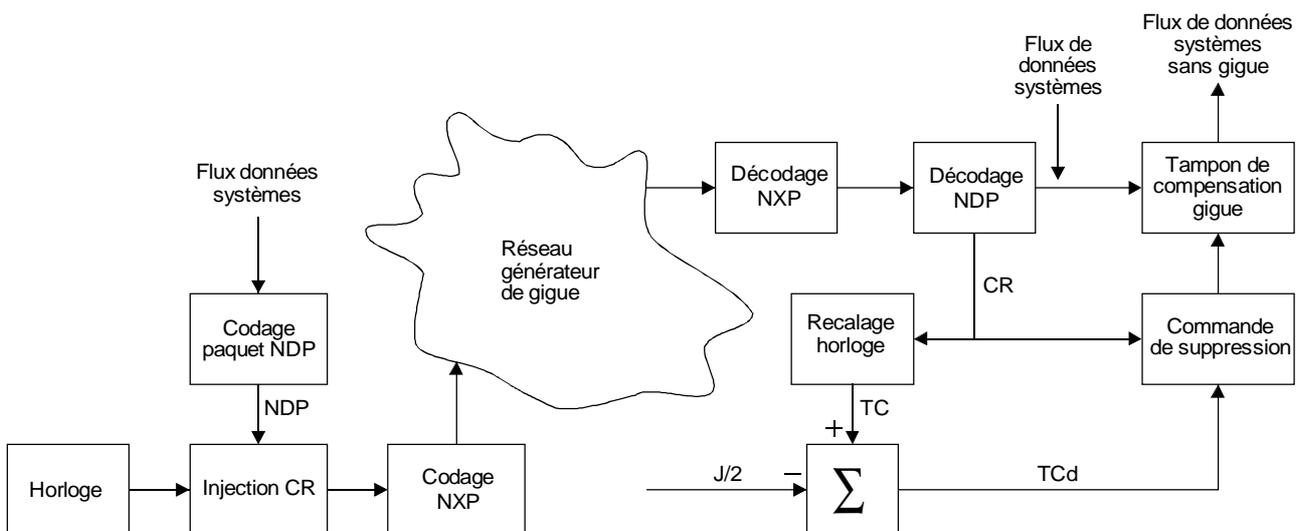


Figure J.2 – Lissage de gigue au moyen de pointeurs temporels issus de la couche réseau

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) échantillonnées sur une horloge de base (TC). 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) 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) 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 de mise en œuvre 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 binaire 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 de mises en œuvre de décodeur

J.3.1 Adaptateur de réseau suivi d'un décodeur MPEG-2

Dans ce type de mise en œuvre, 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 dans J.4.1 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 ce 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 issu du réseau qui est injecté dans le décodeur est le même que celui de la Figure J.2.

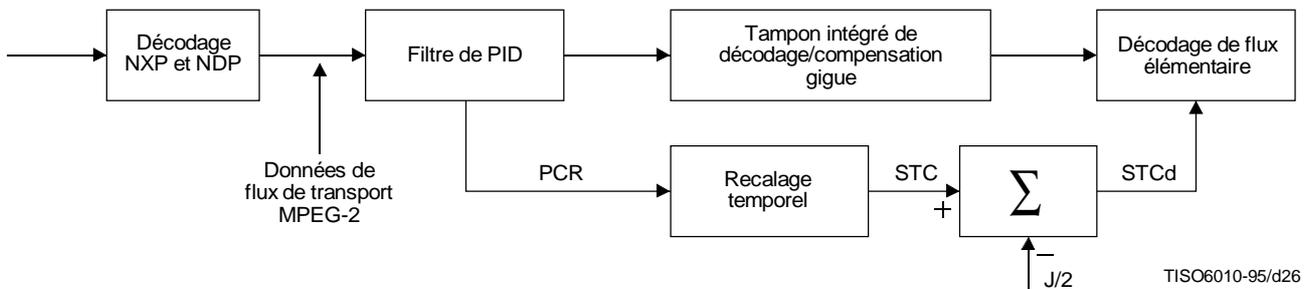


Figure J.3 – Fonctions intégrées de lissage de gigue et de décodage MPEG_2

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 de mise en œuvre 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:

$$B_{size} = B_{dec} + B_{mux} + B_{OH} + 512 + B_j, \quad = B_n + 512 + B_j$$

où $B_j = R_{max} J$, et R_{max} est le débit maximal d'entrée des données dans le filtre PID. Selon la mise en œuvre, 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/ou 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_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 fait abstraction de 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 de mise en œuvre 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 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 de mise en œuvre. 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 binaire 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 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 binaire 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 binaire 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/ou MB_n . Dans le cas d'un débit binaire 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 binaire 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 2-16 aux paramètres 'splice_decoding_delay' et 'max_splice_rate'. Si le débit binaire 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 flux binaire constant R_{old} , et le «nouveau» par un débit binaire constant R_{new} , il suffit d'introduire un délai T_d vérifiant les relations ci-dessous pour éviter ce risque de surremplissage:

$$T_d \geq 0 \text{ et } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old}/R_{new})$$