



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**J.181**

(06/2004)

SÉRIE J: RÉSEAUX CÂBLÉS ET TRANSMISSION DES  
SIGNAUX RADIOPHONIQUES, TÉLÉVISUELS ET  
AUTRES SIGNAUX MULTIMÉDIAS

Transmission numérique des signaux de télévision

---

**Message de repérage d'insertion de programme  
numérique pour systèmes de télévision par  
câble**

Recommandation UIT-T J.181

---



## **Recommandation UIT-T J.181**

### **Message de repérage d'insertion de programme numérique pour systèmes de télévision par câble**

#### **Résumé**

La présente Recommandation s'applique au raccordement de flux de transport MPEG-2 pour l'insertion de programmes numériques, notamment des annonces publicitaires. Elle définit un mécanisme de repérage dans le flux qui signale les occasions de raccordement et d'insertion ainsi qu'une technique pour acheminer la notification des prochains points de raccordement dans le flux de transport.

#### **Source**

La Recommandation UIT-T J.181 a été approuvée le 29 juin 2004 par la Commission d'études 9 (2001-2004) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

## 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. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

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

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

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

## NOTE

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

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

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2005

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

## TABLE DES MATIÈRES

|     | <b>Page</b>  |
|-----|--|
| 1   | Domaine d'application ..... 1                            |
| 2   | Références..... 1  |
| 2.1 | Références normatives..... 1                             |
| 2.2 | Références informatives ..... 2                          |
| 3   | Définition de termes ..... 2                             |
| 4   | Abréviations..... 4                                      |
| 5   | Introduction ..... 4                                     |
| 5.1 | Points de raccordement (informatif)..... 4               |
| 5.2 | Points de raccordement de programme (informatif) ..... 5 |
| 5.3 | Événements de raccordement (informatif) ..... 5          |
| 5.4 | Sélection de l'identificateur PID ..... 6                |
| 5.5 | Flux de message (informatif)..... 7                      |
| 6   | Descripteurs de table PMT ..... 8                        |
| 6.1 | Descripteur d'enregistrement ..... 8                     |
| 6.2 | Descripteur d'identificateur de repérage..... 9          |
| 6.3 | Descripteur d'identificateur de flux ..... 10            |
| 7   | Table d'informations de raccordement ..... 11            |
| 7.1 | Aperçu général..... 11                                   |
| 7.2 | Section des informations de raccordement..... 12         |
| 7.3 | Commandes de raccordement..... 15                        |
| 7.4 | Instant ..... 21   |
| 7.5 | Contraintes..... 22                                      |
| 8   | Descripteurs de raccordement..... 24                     |
| 8.1 | Aperçu général..... 24                                   |
| 8.2 | Descripteur de raccordement..... 25                      |
| 8.3 | Descripteurs de raccordement spécifiques ..... 26        |
| 9   | Cryptage..... 31   |
| 9.1 | Aperçu général..... 31                                   |
| 9.2 | Cryptage par clé fixe ..... 31                           |
| 9.3 | Algorithmes de cryptage..... 32                          |



# Recommandation UIT-T J.181

## Message de repérage d'insertion de programme numérique pour systèmes de télévision par câble

### 1 Domaine d'application

La présente Recommandation s'applique au raccordement de flux MPEG-2 pour l'insertion de programmes numériques, notamment des annonces publicitaires. Elle définit un mécanisme de repérage dans le flux qui signale les occasions de raccordement et d'insertion et dont l'objet n'est pas de garantir un raccordement transparent. Se faisant, la présente Recommandation ne spécifie pas la méthode de raccordement utilisée ou les contraintes appliquées aux flux en cours de raccordement et ne traite pas des contraintes imposées aux dispositifs de raccordement. La présente Recommandation porte par ailleurs sur la signalisation précise des événements dans le flux.

Elle part de l'hypothèse d'un flux de transport entièrement conforme MPEG-2 (flux de transport multiprogramme ou monoprogramme). Aucune contrainte autre que l'introduction des messages de repérage définis n'est imposée au flux.

La présente Recommandation définit une technique pour acheminer la notification des prochains points de raccordement et d'autres informations de synchronisation dans le flux de transport. Elle définit une table d'informations de raccordement pour notifier aux dispositifs avals des événements de raccordement tels qu'une pause dans le réseau ou une reprise après une telle pause. La table des informations de raccordement, qui s'applique à un programme donné, est acheminée dans un ou plusieurs identificateurs PID mentionnés dans la table de contenu (PMT, *program map table*) du programme en question. Ainsi la notification de l'événement de raccordement peut traverser la plupart des remultiplexeurs de flux de transport sans nécessiter un traitement spécial.

### 2 Références

#### 2.1 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

- Recommandation UIT-T H.222.0 (2000) | ISO/CEI 13818-1:2000, *Technologies de l'information – Codage générique des images animées et du son associé: systèmes*.
- Recommandation UIT-T H.262 (2000) | ISO/CEI 13818-2:2000, *Technologies de l'information – Codage générique des images animées et du son associé: données vidéo*.
- ISO/CEI 13818-4:1998, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 4: Essais de conformité*, plus Corrigendum 2 (1998).
- FIPS PUB 46-3-1999, *Data Encryption Standard (DES)*.
- FIPS PUB 81-1980, *DES Modes of Operation*.

## 2.2 Références informatives

- SMPTE 312M, SMPTE Standard for Television – Splice Points for MPEG-2 Transport Streams.

## 3 Définition de termes

Les termes dont il est question ci-dessous ont un sens spécifique dans la présente Recommandation. Par ailleurs, certains termes définis dans l'ISO/CEI 13818 ayant un sens technique très particulier, le lecteur est invité à s'y référer. La présente Recommandation définit brièvement les termes suivants.

**3.1 unité d'accès:** représentation codée d'une unité de présentation (voir la Rec. UIT-T H.262 | ISO/CEI 13818-2).

**3.2 signal de repérage analogique:** dans un système analogique, signal qui est généralement une suite de tonalités DTMF ou une fermeture de contact qui indique à un équipement d'insertion d'annonces publicitaires qu'un espace disponible correspondant va s'ouvrir ou se fermer.

**3.3 espace disponible:** espace temporel fourni aux câblo-opérateurs par des services de programmation au cours d'un programme pour être utilisé par l'opérateur de télévision CATV; cet espace est généralement vendu à des annonceurs locaux ou utilisé pour la promotion personnelle du canal.

**3.4 pause:** espace disponible ou insertion en cours.

**3.5 mode raccordement de composantes:** mode du message de repérage pour lequel le fanion `program_splice_flag` est mis à 0 et indique que chaque identificateur PID ou composante destiné à être raccordé sera énuméré séparément dans la syntaxe qui suit. Les composantes qui ne figurent pas sur la liste du message ne sont pas raccordées.

**3.6 message de repérage:** voir message.

**3.7 événement:** événement de raccordement ou de visionnage.

**3.8 point d'entrée:** point dans un flux, convenant pour l'entrée, qui se trouve à la frontière d'une unité de présentation élémentaire. Il se trouve en fait entre deux unités de présentation et ne constitue pas en lui-même une unité de présentation.

**3.9 dispositif dans le flux:** dispositif recevant directement le flux de transport et capable de déterminer les informations de synchronisation directement à partir de ce flux.

**3.10 message:** dans le contexte de la présente Recommandation, un message est le contenu d'une section `splice_info_section`.

**3.11 flux de transport multiprogramme (*MPTS, multi program transport stream*):** flux de transport comprenant plusieurs programmes.

**3.12 dispositif hors flux:** dispositif recevant le message de repérage transmis par un dispositif dans le flux au travers d'une connexion distincte du flux de transport. Un dispositif hors flux ne reçoit pas ou ne transmet pas directement le flux de transport.

**3.13 point de sortie:** point dans un flux, convenant pour la sortie, qui se trouve à la frontière d'une unité de présentation élémentaire. Il se trouve en fait entre deux unités de présentation et ne constitue pas en lui-même une unité de présentation.

**3.14 indicateur de début d'unité de capacité utile (`payload_unit_start_indicator`):** bit qui figure dans un en-tête de paquet de transport et qui signale, entre autres, qu'une section commence dans la capacité utile qui suit (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1).

**3.15 identificateur de paquet (PID, *packet identifier*):** valeur unique codée sur 13 bits et utilisée pour indiquer le type de données enregistrées dans la capacité utile du paquet (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1).

- 3.16 flux PID:** tous les paquets d'un flux de transport qui ont le même identificateur PID.
- 3.17 champ de pointeur (pointer\_field):** premier octet d'une capacité utile de paquet de transport, nécessaire lorsqu'une section commence dans ce paquet (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1).
- 3.18 instant de présentation:** instant auquel une unité de présentation est présentée dans le décodeur cible du système (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1).
- 3.19 unité de présentation:** unité d'accès audio décodé ou image décodée (voir la Rec. UIT-T H.262 | ISO/CEI 13818-2).
- 3.20 programme:** ensemble de flux PID vidéo, audio et de données qui ont en commun un même numéro de programme à l'intérieur d'un flux MPTS (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1).
- 3.21 point d'entrée de programme:** groupe de points d'entrée de flux PID qui correspondent à l'instant de présentation.
- 3.22 point de sortie de programme:** groupe de points de sortie de flux PID qui correspondent à l'instant de présentation.
- 3.23 mode raccordement de programme:** mode du message de repérage pour lequel le fanion `program_splice_flag` est mis à 1 et indique que le message se réfère à un point de raccordement de programme et que tous les indicateurs PID ou composantes du programme doivent être raccordés.
- 3.24 point de raccordement de programme:** point d'entrée ou un point de sortie de programme.
- 3.25 dispositif de réception:** dispositif qui reçoit ou interprète des sections conformément aux dispositions de la présente Recommandation. Les raccordeurs, les serveurs d'annonces publicitaires, les segmenteurs et les récepteurs de satellite sont des exemples d'un tel dispositif.
- 3.26 descripteur d'enregistrement:** descripteur transporté dans la table PMT d'un programme pour indiquer que les sections `splice_info_section`, lorsqu'elles signalent des événements de raccordement, doivent être acheminées dans un flux PID à l'intérieur de ce programme. La présence du descripteur d'enregistrement signifie la conformité du programme à la présente Recommandation.
- 3.27 réservé:** le terme "réservé", lorsqu'il est utilisé dans les clauses qui définissent le flux binaire codé, indique que la valeur peut être utilisée dans des extensions futures de la Recommandation. Sauf indication contraire dans la présente Recommandation, tous les bits réservés doivent être mis à "1" et les champs correspondants doivent être ignorés par l'équipement de réception.
- 3.28 flux de transport de programme unique (SPTS, single program transport stream):** flux de transport comprenant un seul programme.
- 3.29 événement de raccordement:** occasion de raccorder un ou plusieurs flux PID.
- 3.30 mode raccordement immédiat:** mode du message de repérage pour lequel le dispositif de raccordement doit choisir la possibilité de raccordement la plus rapprochée dans le flux, relativement à la table d'informations de raccordement. Lorsque ce mode n'est pas en vigueur, le message donne un instant `pts_time` qui, après modification par le champ `pts_adjustment`, donne l'instant de présentation correspondant au moment envisagé pour le raccordement.
- 3.31 point de raccordement:** point dans un flux PID qui est un point de sortie ou un point d'entrée.

**3.32 événement de visionnage:** programme de télévision ou intervalle de données compressées à l'intérieur d'un service; par opposition à un événement de raccordement, qui correspond à un instant.

## 4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

ATSC Comité des systèmes de télévision évolués (*advanced television systems committee*)

bslbf chaîne binaire, bit de gauche en premier, le terme "gauche" renvoyant à l'ordre dans lequel les chaînes binaires sont écrites (*bit string, left bit first*)

DVB diffusion vidéonumérique (*digital video broadcast*)

MPTS flux de transport multiprogramme (*multi program transport stream*)

PMT table de contenu de programme (*program map table*) (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1)

PTS horodateur de présentation (*presentation time stamp*) (voir la Rec. UIT-T H.222.0 | ISO/CEI 13818-1)

rpchof coefficients du polynôme de reste, ordre le plus élevé en premier (*remainder polynomial coefficients, highest order first*)

SPTS flux de transport de programme unique (*single program transport stream*)

STC horloge de base du système (*system time clock*)

uimsbf entier non signé, bit de plus fort poids en premier (*unsigned integer, most significant bit first*)

## 5 Introduction

### 5.1 Points de raccordement (informatif)

La présente Recommandation définit des points de raccordement pour permettre de raccorder des flux binaires compressés. Dans un flux de transport MPEG-2, les points de raccordement offrent la possibilité de commuter les flux élémentaires d'une source à une autre. Ils indiquent un endroit pour la commutation ou un endroit dans le flux binaire où une commutation peut avoir lieu. Un raccordement réalisé en un tel point peut se traduire par une qualité visuelle et sonore plus ou moins bonne, celle-ci étant déterminée par la qualité de fonctionnement du dispositif de raccordement.

Les flux de transport sont créés par multiplexage de flux PID. Deux types de points de raccordement pour des flux PID sont définis dans la présente Recommandation: des points de sortie et des points d'entrée. Les points d'entrée sont des endroits dans les flux binaires où il est acceptable d'entrer, du point de vue du raccordement. Les points de sortie sont des endroits où il est acceptable de quitter le flux binaire. La présente Recommandation définit le groupage des points d'entrée de flux PID individuels en points d'entrée de programme afin de permettre la commutation de programmes entiers (vidéo et audio). Elle définit également des points de sortie de programme, permettant de quitter un programme.

Les points de sortie et les points d'entrée sont des points imaginaires du flux binaire situés entre deux unités de présentation de flux élémentaire. Ils ne sont pas nécessairement alignés sur les paquets de transport ou sur les paquets PES. Un point de sortie et un point d'entrée peuvent être situés au même endroit; autrement dit, une même frontière d'unité de présentation peut servir d'endroit sûr pour quitter un flux binaire et d'endroit sûr pour y entrer.

Le résultat d'une opération de commutation simple contiendra des données d'unité d'accès provenant d'un flux jusqu'à son point de sortie suivies de données d'un autre flux commençant par la première unité d'accès qui suit un point d'entrée. On peut avoir des opérations de raccordement plus complexes pour lesquelles les données qui précèdent un point de sortie ou les données suivant un point d'entrée sont modifiées par un dispositif de raccordement. Celui-ci peut également introduire des données entre le point de sortie d'un flux et le point d'entrée d'un autre flux. Le comportement des dispositifs de raccordement ne sera pas spécifié ou limité de quelque façon que ce soit par la présente Recommandation.

## **5.2 Points de raccordement de programme (informatif)**

Les points d'entrée de programme et les points de sortie de programme sont des ensembles de points d'entrée et de points de sortie de flux PID qui correspondent à l'instant de présentation.

Bien que les points de raccordement d'un programme correspondent à l'instant de présentation, il ne sont généralement pas situés les uns à côté des autres dans le flux de transport. Etant donné que la vidéo compressée est beaucoup plus longue à décoder que l'audio compressé, les points de raccordement audio peuvent être décalés de plusieurs dixièmes de seconde par rapport aux points de raccordement vidéo, décalage qui peut varier au cours du programme.

La présente Recommandation définit deux façons de signaler les points de raccordement d'un programme qui doivent être raccordés. Un fanion `program_splice_flag`, lorsqu'il est "Vrai", indique que le mode raccordement de programme est actif et que tous les identificateurs PID d'un programme peuvent être raccordés (l'identificateur PID de la table d'informations de raccordement est une exception; le raccordement ou le passage de ces messages ne relève pas du domaine de la présente Recommandation). Un fanion `program_splice_flag`, lorsqu'il est "Faux", indique que le mode raccordement de composantes est actif et que le message spécifiera de manière non ambiguë les identificateurs PID qu'il y a lieu de raccorder; il peut donner un instant de raccordement distinct pour chacun d'eux. Cela est nécessaire pour ordonner au dispositif de raccordement de raccorder ou de ne pas raccorder divers types de données non spécifiés ainsi que l'image et le son.

Bien que la présente Recommandation permette d'utiliser un instant de raccordement distinct pour chaque composante d'un programme, on prévoit que la plupart des messages du mode raccordement de composantes utiliseront un seul instant de raccordement (un instant de raccordement par défaut) pour toutes les composantes, comme indiqué au § 7. La possibilité de spécifier facultativement un instant de raccordement distinct pour chaque composante est prévue pour être utilisée lorsqu'une ou plusieurs composantes diffèrent notablement en ce qui concerne leurs instants de départ et d'arrêt par rapport à d'autres composantes du même message. Un exemple serait un appel téléchargé qui devrait arriver au boîtier convertisseur-décodeur plusieurs secondes avant une annonce publicitaire.

## **5.3 Événements de raccordement (informatif)**

La présente Recommandation propose une méthode pour signaler dans la bande des événements de raccordement à l'aide de commandes de raccordement à l'équipement de raccordement aval. La signalisation d'un événement de raccordement identifie le point de raccordement d'un flux qu'il y a lieu d'utiliser pour un raccordement. Un dispositif de raccordement peut décider de donner suite ou non à un événement signalé (un événement signalé devrait être interprété comme une occasion de faire un raccordement et non comme une commande). Une table d'informations de raccordement contient les informations sur les occasions d'événement de raccordement. Chaque événement de raccordement signalé est analogue à une tonalité de repérage analogique. La table d'informations de raccordement englobe la fonctionnalité des tonalités de repérage et l'étend pour permettre de programmer à l'avance les événements de raccordement.

La présente Recommandation établit que la table d'informations de raccordement est acheminée pour chaque programme dans un ou plusieurs flux PID avec un type de flux `stream_type` désigné. Le ou les identificateurs PID d'informations de raccordement de programme sont désignés dans la table de contenu de programme (PMT). Ainsi, la table d'informations de raccordement est commutée avec le programme lorsqu'elle subit des opérations de remultiplexage. Un type `stream_type` commun identifie tous les flux PID qui acheminent des tables d'informations de raccordement. Les remultiplexeurs ou les raccordeurs peuvent utiliser ce champ `stream_type` pour supprimer des informations de raccordement avant l'envoi du flux de transport jusqu'au dispositif de l'utilisateur final.

L'équipement d'insertion de messages de repérage peut envoyer des messages suivant des intervalles de temps n'indiquant pas qu'un point de raccordement est utilisé pour des messages de pulsation qui permettent de s'assurer du fonctionnement correct du système. Cela pourrait être réalisé en envoyant périodiquement des messages `splice_null()` ou en envoyant des messages `splice_insert` cryptés générés à l'aide d'une clé non distribuée. Puisque les messages de repérage sont à présent envoyés deux fois par heure dans un réseau type, le choix d'une période moyenne de 5 minutes semble raisonnable. Si un message n'est pas reçu au cours d'une période de 10 minutes, le dispositif de réception pourrait alerter un opérateur pour signaler un dysfonctionnement éventuel du système (un tel comportement serait propre à l'implémentation considérée).

## **5.4 Sélection de l'identificateur PID**

### **5.4.1 Sélection de l'identificateur PID (normatif)**

Les informations de raccordement peuvent être acheminées par plusieurs identificateurs PID. Le nombre maximal d'identificateurs PID susceptibles d'acheminer les informations de raccordement ne doit pas être supérieur à 8. Ces identificateurs peuvent soit apparaître en clair (lorsque les bits de transport `scrambling_control` sont mis à "00"), soit être embrouillés par un système d'accès conditionnel. Chaque identificateur PID de message de repérage peut inclure le descripteur `cue_identifier_descriptor` défini au § 6.2 pour décrire les commandes de raccordement qu'il contient. Lorsque plusieurs identificateurs PID sont utilisés pour acheminer les informations de raccordement, le premier identificateur de message de repérage dans la table de contenu de programme doit contenir uniquement les types de commande de raccordement 0x00 (`splice_null`), 0x04 (`splice_schedule`) et 0x05 (`splice_insert`). En outre, l'identificateur `splice_event_id` doit être unique parmi tous les identificateurs PID d'informations de raccordement dans le programme.

### **5.4.2 Sélection d'identificateur PID (informatif)**

Si l'utilisation de plusieurs identificateurs de message de repérage est une pratique autorisée, il convient de noter que les équipements peuvent ne pas tous répondre de la même manière à l'arrivée d'un flux contenant plusieurs identificateurs de message de repérage. Certains équipements peuvent limiter le nombre d'identificateurs qu'ils peuvent transmettre ou recevoir. Dans le cas d'un système utilisant plusieurs identificateurs PID via divers dispositifs dans l'intention d'atteindre le boîtier décodeur, la réalisation d'un test complet de bout en bout est conseillée.

Dans de nombreux systèmes, la fourniture d'identificateurs PID acheminant des informations de raccordement au-delà de l'équipement d'insertion d'annonces publicitaires dans la tête de réseau n'est pas souhaitée. Dans ces systèmes, le dispositif de raccordement ou de multiplexage supprimera tous ces messages (PID) de telle sorte qu'ils ne soient pas fournis au boîtier décodeur. Dans d'autres systèmes, ce dispositif pourra transmettre de façon sélective certains identificateurs PID au boîtier décodeur afin d'en émuler une fonctionnalité. Une troisième possibilité consiste pour le dispositif à regrouper les différents identificateurs PID acheminant les informations de raccordement pour former un seul identificateur PID afin de traiter les questions se posant en aval, au niveau du boîtier décodeur, pour ces différents identificateurs. Il est recommandé que l'action d'ignorer ou de transmettre le message dépende du choix de l'utilisateur, un mode de fonctionnement par défaut approprié étant proposé par le concepteur.

Lorsqu'un dispositif de raccordement ou de multiplexage reçoit un identificateur PID conforme à la présente Recommandation avec des bits d'embrouillage mis à 1 dans l'en-tête, le mode de fonctionnement par défaut devrait être de supprimer cet identificateur PID et de ne pas le transmettre en sortie. Le choix de ce mode devrait théoriquement être du ressort de l'utilisateur, puisque dans certains cas l'identificateur PID peut être désembrouillé par un dispositif aval.

La fourniture de messages hors de l'emplacement de réception vers le client peut être fondée sur des accords commerciaux. Par exemple, un programmeur peut souhaiter que les messages de repérage envoyés au boîtier décodeur émulent une méthode d'offres publicitaires ciblée alors que pour un autre programmeur, ces messages devront être supprimés pour s'assurer qu'ils ne pourront pas être utilisés par un supprimeur d'annonces publicitaires.

Lorsque plusieurs identificateurs PID de raccordement sont identifiés dans la table PMT, tous devraient être traités par le dispositif de raccordement. S'il est présent, le descripteur `cue_identifier_descriptor` peut être utilisé par le dispositif de raccordement ou de multiplexage pour choisir de manière plus sélective les identificateurs PID sur lesquels le dispositif va agir.

Parmi les raisons possibles d'utiliser plusieurs PID pour un tel message, on citera la fourniture sélective de messages de repérage pour différents niveaux de publicité ou pour distinguer les messages de repérage des messages de segmentation. Bien qu'une des méthodes possibles pour traiter ces questions soit d'utiliser les méthodes de cryptage indiquées dans la présente Recommandation, de nombreux mécanismes de distribution peuvent prendre en charge la fourniture conditionnelle par un identificateur PID en mode sécurisé. L'équipement de distribution (émetteur/récepteur de satellite, remultiplexeur) peuvent filtrer les identificateurs PID du flux de telle sorte à ne permettre la transmission dans le flux que d'un seul identificateur ou que d'un petit nombre d'identificateurs. Cette méthode peut être utilisée pour créer plusieurs programmes dans l'alimentation sur la base d'une habilitation. La décision d'utiliser un ou plusieurs identificateurs PID sera fondée sur le niveau de sécurité requis et sur la disponibilité d'équipements d'accès conditionnel dans le système.

## **5.5 Flux de message (informatif)**

Les messages décrits dans la présente Recommandation peuvent provenir de diverses sources. Ils sont conçus pour être transmis dans le flux à destination des dispositifs aval. Les dispositifs aval peuvent traiter ces messages ou les envoyer à un dispositif de traitement qui n'est pas dans le flux. On peut citer l'exemple d'un raccordeur communiquant via le protocole SCTE 30 avec un serveur d'annonces publicitaires. Les dispositifs dans le flux peuvent transmettre les messages au dispositif suivant dans la chaîne de transmission ou peuvent facultativement décider de supprimer les messages. Les concepteurs sont instamment priés de laisser le choix de ces décisions à l'utilisateur plutôt que de les coder "en dur" de façon arbitraire.

Tout dispositif qui horodate à nouveau les champs `pcr/pts/dts` et qui transmet ces messages de repérage à un dispositif aval devrait modifier le champ `pts_time` ou le champ `pts_adjustment` dans tous les identificateurs PID conformes à la présente Recommandation. La modification du champ `pts_adjustment` est préférée parce que le dispositif de réhorodatage n'aura pas à connaître le champ `pts_time` qui peut apparaître dans plusieurs commandes (et éventuellement dans des commandes futures).

Le message `bandwidth_reservation()` est destiné à être un message utilisé sur un trajet fermé reliant le système d'origine sur satellite (codeur) à un récepteur. Il est également prévu que le récepteur supprime (remplace par un paquet NULL) ce message, mais cela n'est pas obligatoire. S'il parvenait à un dispositif dans le flux (un raccordeur par exemple), le message ne devrait pas être transmis à un dispositif hors flux (par exemple un serveur d'annonces publicitaires) et peut être soit ignoré soit transmis par un dispositif situé dans le flux. Il est recommandé que l'action d'ignorer ou de transmettre un message dépende du choix de l'utilisateur, un mode de fonctionnement par défaut approprié étant proposé par le concepteur.

## 6 Descripteurs de table PMT

### 6.1 Descripteur d'enregistrement

Le descripteur d'enregistrement (voir Tableau 2-45 ("Descripteur d'enregistrement"), au § 2.6.8 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1) est défini pour identifier sans équivoque les programmes qui sont conformes à la présente Recommandation. Le descripteur d'enregistrement doit être acheminé dans la boucle `program_info` de la table PMT pour chaque programme conforme à la présente Recommandation. Il doit résider dans toutes les tables PMT de tous les programmes conformes à l'intérieur d'un multiplex. La présence du descripteur d'enregistrement indique également, lors de la signalisation d'événements de raccordement, que des sections `splice_info_section` devront être acheminées dans un ou plusieurs flux PID à l'intérieur de ce programme.

La présence de ce descripteur d'enregistrement dans la table PMT signale les points suivants:

- 1) les éléments de programme ne comprennent pas la table d'informations de raccordement définie par la norme SMPTE 312M;
- 2) les seuls descripteurs qui peuvent être présents dans la boucle `ES_descriptor_loop` de la table PMT pour le ou les identificateurs PID qui acheminent la table des informations de raccordement sont ceux définis dans la présente Recommandation ou sont des descripteurs privés d'utilisateur.

On notera que ce descripteur s'applique au programme indiqué et non à l'ensemble du multiplex. Le contenu du descripteur d'enregistrement est spécifié dans le Tableau 6-1 et ci-dessous:

Tableau 6-1/J.181 – `registration_descriptor()`

| Syntaxe   | Nombre de bits                    | Mnémonique                                      |
|---|-----------------------------------|---|
| <code>registration_descriptor() {</code><br><code>  <b>descriptor_tag</b></code><br><code>  <b>descriptor_length</b></code><br><code>  <b>SCTE_splice_format_identifieur</b></code><br><code>}</code> | <b>8</b><br><b>8</b><br><b>32</b> | <b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b> |

#### 6.1.1 Définition sémantique des champs du descripteur d'enregistrement

**descriptor\_tag** (étiquette de descripteur): champ de 8 bits identifiant chaque descripteur. Pour l'enregistrement, ce champ doit être mis à 0x05.

**descriptor\_length** (longueur du descripteur): champ de 8 bits qui spécifie le nombre d'octets du descripteur immédiatement après le champ `descriptor_length`. Pour ce descripteur d'enregistrement, la longueur `descriptor_length` sera mise à 0x04.

**SCTE\_splice\_format\_identifieur** (identificateur de format de raccordement): la SCTE a attribué une valeur de 0x43554549 ("CUEI" en ASCII) à ce champ de 4 octets pour identifier le programme (dans un multiplex) dans lequel il est acheminé en conformité avec la présente Recommandation.

## 6.2 Descripteur d'identificateur de repérage

Le descripteur `cue_identifier_descriptor` peut être utilisé dans la table PMT pour étiqueter les identificateurs PID acheminant des commandes de raccordement de telle sorte que ces identificateurs puissent être différenciés du point de vue du type ou du niveau de commandes de raccordement acheminés. Le descripteur `cue_identifier_descriptor` doit être, s'il est présent, situé dans la boucle du descripteur élémentaire. Si le descripteur `cue_identifier_descriptor` n'est pas utilisé, le flux peut acheminer toute commande valable dans la présente Recommandation. Voir Tableau 6-2.

Tableau 6-2/J.181 – `cue_identifier_descriptor()`

| Syntaxe   | Nombre de bits             | Mnémonique   |
|---|----------------------------|--|
| <pre>cue_identifier_descriptor() {   descriptor_tag   descriptor_length   cue_stream_type }</pre> | <p>8</p> <p>8</p> <p>8</p> | <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> |

### 6.2.1 Définition sémantique des champs dans le descripteur d'identificateur de repérage

**descriptor\_tag** (étiquette de descripteur): champ de 8 bits qui identifie chaque descripteur. Pour le descripteur `cue_identifier_descriptor`, ce champ doit être mis à 0x8A.

**descriptor\_length** (longueur du descripteur): champ de 8 bits qui spécifie le nombre d'octets du descripteur immédiatement après le champ `descriptor_length`. Pour ce descripteur, la longueur `descriptor_length` doit être mise à 0x01.

**cue\_stream\_type** (type de flux de repérage): champ de 8 bits dont les valeurs sont définies dans le Tableau 6-3.

Tableau 6-3/J.181 – valeurs du type `cue_stream_type`

| <code>cue_stream_type</code> | Utilisation de l'identificateur PID  |
|------------------------------|--|
| 0x00                         | <code>splice_insert</code> , <code>splice_null</code> , <code>splice_schedule</code> |
| 0x01                         | Toutes les commandes   |
| 0x02                         | Segmentation   |
| 0x03                         | Raccordement par paliers   |
| 0x04                         | Segmentation par paliers   |
| 0x05-0x7F                    | Réservée   |
| 0x80-0xFF                    | Définie par l'utilisateur  |

### 6.2.2 Description de l'utilisation du type `cue_stream_type`

**0x00 – `splice_insert`, `splice_null`, `splice_schedule`**: seuls ces messages de repérage sont autorisés dans ce flux PID. Un identificateur PID au maximum est identifié par ce type `cue_stream_type`. S'il existe, cet identificateur PID doit être le premier flux conforme à la présente Recommandation dans la boucle de flux élémentaire dans la table PMT.

**0x01 – Toutes les commandes**: valeur par défaut si ce descripteur n'est pas présent. Tous les messages peuvent être utilisés dans cet identificateur PID.

**0x02 – Segmentation:** l'identificateur PID considéré achemine la commande `time_signal` et le descripteur de segmentation. Il peut également acheminer toutes les autres commandes si cela est nécessaire pour l'application, mais l'objectif principal est de transmettre les informations de segmentation de contenu.

**0x03 – raccordement par paliers:** le raccordement par paliers se réfère à un système d'insertion pour lequel l'opérateur fournit différentes possibilités d'insertion de programme dans un espace disponible donné pour différents clients. Les implémentations physique et logique peuvent être faites de diverses manières, dont certaines ne relèvent pas du domaine d'application de la présente Recommandation.

**0x04 – segmentation par paliers:** la segmentation par paliers se réfère à un système pour lequel un opérateur fournit différentes possibilités de segmentation de programme pour différents clients. Les implémentations physique et logique peuvent être faites de diverses manières, dont certaines ne relèvent pas du domaine d'application de la présente Recommandation.

**0x05-0x7F:** réservé pour des extensions futures de la présente Recommandation.

**0x80-0xFF:** intervalle défini par l'utilisateur.

### 6.3 Descripteur d'identificateur de flux

Le descripteur d'identificateur de flux peut être utilisé dans la table PMT pour étiqueter les flux de composantes d'un service afin qu'ils puissent être différenciés. Le descripteur d'identificateur de flux doit être situé dans la boucle du descripteur élémentaire qui suit le champ `ES_info_length` qui s'applique. Le descripteur d'identificateur de flux doit être utilisé si le fanion `program_splice_flag` ou `program_segmentation_flag` est mis à zéro. Si l'on utilise des descripteurs d'identificateur de flux, de tels descripteurs doivent être présents dans chaque occurrence de la boucle de flux élémentaire dans la table PMT et auront des étiquettes de composante uniques à l'intérieur d'un programme donné. Voir Tableau 6-4.

Tableau 6-4/J.181 – `stream_identifier_descriptor()`

| Syntaxe  | Nombre de bits             | Mnémonique   |
|--|----------------------------|--|
| <pre>stream_identifier_descriptor() {   descriptor_tag   descriptor_length   component_tag }</pre> | <p>8</p> <p>8</p> <p>8</p> | <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> |

#### 6.3.1 Définition sémantique des champs du descripteur d'identificateur de flux

**descriptor\_tag** (étiquette de descripteur): champ de 8 bits qui identifie chaque descripteur. Pour le descripteur d'identificateur de flux, ce champ doit être mis à 0x52.

**descriptor\_length** (longueur du descripteur): champ de 8 bits qui spécifie le nombre d'octets du descripteur immédiatement après le champ `descriptor_length`. Pour ce descripteur, la longueur `descriptor_length` doit être mise à 0x01.

**component\_tag** (étiquette de composante): ce champ de 8 bits identifie le flux de composantes en vue de l'associer à une description donnée dans un descripteur de composante. A l'intérieur d'une section de contenu de programme, chaque descripteur d'identificateur de flux doit avoir une valeur différente pour ce champ.

## 7 Table d'informations de raccordement

### 7.1 Aperçu général

La table d'informations de raccordement fournit au dispositif de raccordement les informations de commande et de contrôle. Elle notifie au dispositif de raccordement les événements de raccordement avant que ceux-ci n'aient lieu. Elle est conçue pour prendre en charge l'insertion d'une annonce publicitaire dans une alimentation réseau. Dans cet environnement, des exemples d'événements de raccordement sont:

- 1) un raccordement pour le passage d'une alimentation réseau à une annonce publicitaire;
- 2) un raccordement pour le passage d'une annonce publicitaire à l'alimentation réseau.

La table d'informations de raccordement peut être envoyée plusieurs fois et des événements de raccordement peuvent être annulés. La syntaxe relative à une section `splice_info_section` est définie de manière à pouvoir acheminer la table d'informations de raccordement. La section `splice_info_section` est acheminée dans un ou plusieurs flux PID avec le ou les identificateurs PID déclarés dans la table PMT de ce programme.

Un événement de raccordement indique la possibilité de raccorder un ou plusieurs flux élémentaires à l'intérieur d'un programme. Chaque événement de raccordement est identifié de manière univoque par un identificateur `splice_event_id`. On peut communiquer des événements de raccordement de trois façons différentes: ils peuvent être programmés à l'avance, un avertissement précédant le lancement peut être donné ou une commande d'exécution de l'événement de raccordement peut être donnée en tout point du raccordement spécifié. Ces trois types de message sont envoyés via la section `splice_info_section`. Le champ `splice_command_type` spécifie le message en cours d'envoi. Selon la valeur de ce champ, des contraintes différentes s'appliquent à la syntaxe restante.

Les types de commande suivants sont spécifiés: `splice_null()`, `splice_schedule()`, `splice_insert()`, `time_signal()` et `bandwidth_reservation()`. S'il ne prend pas en charge une commande, le dispositif de réception peut ignorer la totalité de la section `splice_info_section`.

La commande `splice_null()` est destinée à des extensions. Elle peut être utilisée pour fournir un message de pulsation à l'équipement de raccordement aval.

La commande `splice_schedule()` permet d'acheminer à l'avance un horaire des événements de raccordement.

La commande `splice_insert()` doit être envoyée au moins une fois avant chaque point de raccordement. Les paquets contenant l'ensemble de la table `splice_info_table` doivent toujours précéder le paquet contenant le point de raccordement correspondant (c'est-à-dire le premier paquet qui contient le premier octet d'une unité d'accès dont l'instant de présentation est le plus proche de l'instant signalé dans la section `splice_info_section`).

Pour donner un avertissement anticipé d'un raccordement imminent préalable au lancement (une fonction), la commande `splice_insert()` peut être envoyée plusieurs fois avant le point de raccordement, par exemple à 8, 5, 4 et 2 secondes avant le paquet contenant le point de raccordement correspondant. Pour respecter d'autres délais de raccordement dans le système, tout message reçu moins de quatre secondes avant un avertissement anticipé risque de ne pas conduire au résultat souhaité. Le message `splice_insert()` doit être envoyé au moins une fois et au minimum 4 secondes avant l'instant de raccordement souhaité dans le cas d'une condition de point de sortie de réseau.

La commande `time_signal()` est fournie à des fins d'extension tout en conservant la synchronisation précise permise dans la commande `splice_insert()`. Il s'agit de permettre la prise en charge de nouvelles caractéristiques qui ne sont pas directement liées au raccordement en utilisant les capacités de synchronisation offertes par la présente Recommandation tout en veillant à limiter au minimum l'incidence sur les dispositifs de raccordement conformes à la présente Recommandation.

Cela permettra aux dispositifs qui inséreront les données de synchronisation dans le message de repérage d'avoir une position bien définie.

La commande `bandwidth_reservation()` est fournie pour permettre aux dispositifs d'insertion de commande d'utiliser une valeur cohérente de largeur de bande de flux de transport. Des descripteurs peuvent être utilisés dans cette commande, mais on ne peut pas s'attendre à ce qu'ils soient traités et envoyés vers l'aval pour fournir des informations de signalisation.

Il existe deux méthodes pour modifier après lancement les paramètres d'une commande. L'une consiste à annuler la commande émise en envoyant une section `splice_info_section` avec l'indicateur `splice_event_cancel_indicator` mis à 1 puis à envoyer une nouvelle section `splice_info_section` avec les paramètres nouveaux/corrects. L'autre méthode consiste simplement à envoyer un nouveau message avec les nouvelles données (sans annuler l'ancien message à l'aide d'un message de repérage dont l'indicateur `splice_event_cancel_indicator` est mis à 1).

### **7.1.1 Discontinuités de la base de temps**

Lorsque se présente une discontinuité de la base de temps du système, les paquets contenant une commande `splice_insert()` ou `time_signal()` avec un instant exprimé dans la nouvelle base de temps ne doivent pas arriver avant l'occurrence de cette discontinuité. Les paquets contenant une commande `splice_insert()` ou `time_signal()` avec un instant exprimé dans la base de temps précédente ne doivent pas parvenir après l'occurrence de la discontinuité de la base de temps. Voir l'ISO/CEI 13818-4.

La syntaxe complète est présentée ci-après et suivie de la définition de termes et de contraintes.

### **7.2 Section des informations de raccordement**

La section `splice_info_section` doit être acheminée dans des paquets de transport, sachant qu'un paquet de transport donné ne peut contenir qu'une seule section ou une section partielle. Les sections `splice_info_section` doivent toujours commencer au début d'une capacité utile de paquet de transport. Lorsqu'une section commence dans un paquet de transport, le champ `pointer_field` doit être présent et égal à 0x00, le bit indicateur `payload_unit_start_indicator` devant être égal à un (conformément aux prescriptions de l'utilisation de la syntaxe de section définies dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1). Voir Tableau 7-1.

Tableau 7-1/J.181 – splice\_info\_section()

| Syntaxe                          | Nombre de bits | Mnémonique    | Crypté |
|----------------------------------|----------------|---------------|--------|
| splice_info_section() {          |                |               |        |
| <b>table_id</b>                  | 8              | <b>uimsbf</b> |        |
| <b>section_syntax_indicator</b>  | 1              | <b>bslbf</b>  |        |
| <b>private_indicator</b>         | 1              | <b>bslbf</b>  |        |
| <b>reserved</b>                  | 2              | <b>bslbf</b>  |        |
| <b>section_length</b>            | 12             | <b>uimsbf</b> |        |
| <b>protocol_version</b>          | 8              | <b>uimsbf</b> |        |
| <b>encrypted_packet</b>          | 1              | <b>bslbf</b>  |        |
| <b>encryption_algorithm</b>      | 6              | <b>uimsbf</b> |        |
| <b>pts_adjustment</b>            | 33             | <b>uimsbf</b> |        |
| <b>cw_index</b>                  | 8              | <b>uimsbf</b> |        |
| <b>reserved</b>                  | 12             | <b>bslbf</b>  |        |
| <b>splice_command_length</b>     | 12             | <b>uimsbf</b> |        |
| <b>splice_command_type</b>       | 8              | <b>uimsbf</b> | E      |
| if (splice_command_type == 0x00) |                |               |        |
| splice_null()                    |                |               | E      |
| if (splice_command_type == 0x04) |                |               |        |
| splice_schedule()                |                |               | E      |
| if (splice_command_type == 0x05) |                |               |        |
| splice_insert()                  |                |               | E      |
| if (splice_command_type == 0x06) |                |               |        |
| time_signal()                    |                |               | E      |
| if (splice_command_type == 0x07) |                |               |        |
| bandwidth_reservation()          |                |               | E      |
| <b>descriptor_loop_length</b>    | 16             | <b>uimsbf</b> | E      |
| for (i = 0; i < N1; i++)         |                |               |        |
| splice_descriptor()              |                |               | E      |
| for (i = 0; i < N2; i++)         |                |               |        |
| <b>alignment_stuffing</b>        | 8              | <b>bslbf</b>  | E      |
| if (encrypted_packet)            |                |               |        |
| <b>E_CRC_32</b>                  | 32             | <b>rpchof</b> | E      |
| <b>CRC_32</b>                    | 32             | <b>rpchof</b> |        |
| }                                |                |               |        |

### 7.2.1 Définition sémantique des champs de la section splice\_info\_section()

**table\_id** (identificateur de table): champ codé sur 8 bits. Sa valeur doit être 0xFC.

**section\_syntax\_indicator** (indicateur de syntaxe de section): champ codé sur 1 bit qui doit toujours être mis à "0", indiquant qu'il y a lieu d'utiliser des sections courtes MPEG.

**private\_indicator** (indicateur "privé"): fanion codé sur 1 bit qui doit être mis à 0.

**section\_length** (longueur de section): champ codé sur 12 bits qui spécifie le nombre d'octets restants dans la section `splice_info_section` immédiatement après le champ `section_length` jusqu'à la fin de la section `splice_info_section`. La valeur de ce champ ne doit pas dépasser 4093.

**protocol\_version** (version de protocole): champ d'entier non signé codé sur 8 bits dont la fonction est de permettre, dans l'avenir, à ce type de table d'acheminer des paramètres qui peuvent être structurés de manière différente de celles définies dans le protocole actuel. A l'heure actuelle, la seule valeur valable pour **protocol\_version** est zéro. Des valeurs non nulles de **protocol\_version** peuvent être utilisées dans une future version de la présente Recommandation pour indiquer des tables structurellement différentes.

**encrypted\_packet** (paquet crypté): lorsqu'il est mis à "1", ce bit indique que des portions de la section `splice_info_section`, qui commence par le type `splice_command_type` et se termine par un code `E_CRC_32` inclus, sont cryptées. Quand ce bit est à "0", aucune partie du message n'est cryptée. Les portions éventuellement cryptées de la table `splice_info_table` sont désignées par la lettre E dans la colonne "Crypté" du Tableau 7-1.

**encryption\_algorithm** (algorithme de cryptage): cet entier non signé codé sur 6 bits spécifie l'algorithme de cryptage utilisé pour crypter le message en question. Lorsque le bit `encrypted_packet` est à zéro, ce champ est présent mais non défini. Voir le § 9, et en particulier le Tableau 9-1 pour les détails sur l'utilisation de ce champ.

**pts\_adjustment** (ajustement d'horodate PTS): entier non signé codé sur 33 bits qui apparaît en clair et qui doit être utilisé par un dispositif de raccordement comme une compensation à ajouter au ou aux champs `pts_time` (parfois) cryptés dans l'ensemble du message pour obtenir le ou les instants de raccordement voulus. Lorsque ce champ a la valeur zéro, le ou les champs `pts_time` doivent être utilisés sans compensation. Normalement, le créateur d'un message de repérage mettra ce champ à zéro. Cette valeur d'ajustement est le moyen par lequel un dispositif amont, qui horodate à nouveau les `pcr/pts/dts`, peut acheminer jusqu'au dispositif de raccordement les moyens permettant de convertir le champ `pts_time` du message en un domaine temporel nouvellement imposé.

On prévoit que le premier dispositif qui horodate à nouveau les `pcr/pts/dts` et qui transmet le message de repérage insérera une valeur dans le champ `pts_adjustment` qui est le temps delta entre le domaine temporel d'entrée de ce dispositif et son domaine temporel de sortie. Tous les dispositifs subséquents, qui horodatent aussi à nouveau les `pcr/pts/dts`, peuvent encore modifier le champ `pts_adjustment` en ajoutant leur temps delta au temps delta existant du champ et en remettent le résultat dans le champ `pts_adjustment`. Après chaque modification du champ `pts_adjustment`, le dispositif faisant la modification doit recalculer et mettre à jour le champ `CRC_32`.

La valeur du champ `pts_adjustment` doit à tout moment être la valeur appropriée qu'il y a lieu d'utiliser pour convertir le champ `pts_time` à la base de temps en vigueur. La conversion est faite par l'adjonction de deux champs. En présence d'une situation de bouclage ou de dépassement de capacité, il faut ignorer l'acheminement.

**cw\_index** (indice de mot de contrôle): entier non signé codé sur 8 bits qui achemine le mot de contrôle (clé) à utiliser pour décrypter le message. Le dispositif de raccordement peut enregistrer jusqu'à 256 clés précédemment fournies à cet effet. Lorsque le bit `encrypted_packet` est à zéro, ce champ est présent mais non défini.

**splice\_command\_length** (longueur de la commande de raccordement): champ de 12 bits indiquant la longueur de la commande de raccordement. Si ce champ vaut `0xffff`, la longueur n'est pas définie.

**splice\_command\_type** (type de commande de raccordement): entier non signé codé sur 8 bits auquel est attribué une des valeurs montrées dans le Tableau 7-2.

**Tableau 7-2/J.181 – Valeurs de splice\_command\_type**

| Valeur de splice_command_type | Commande              |
|-------------------------------|-----------------------|
| 0x00                          | splice_null           |
| 0x01                          | Réservé               |
| 0x02                          | Réservé               |
| 0x03                          | Réservé               |
| 0x04                          | splice_schedule       |
| 0x05                          | splice_insert         |
| 0x06                          | time_signal           |
| 0x07                          | bandwidth_reservation |
| 0x08-0xFF                     | Réservé               |

**descriptor\_loop\_length** (longueur de boucle de descripteur): entier non signé codé sur 16 bits qui spécifie le nombre d'octets utilisés dans la boucle de descripteur de raccordement qui suit immédiatement.

**alignment\_stuffing** (bourrage pour alignement): lorsqu'un cryptage est utilisé, ce champ dépend de l'algorithme de cryptage particulier qui a été choisi. Etant donné que certains algorithmes de cryptage requièrent une longueur spécifique pour les données cryptées, il est nécessaire de permettre l'insertion d'octets de bourrage. L'algorithme DES, par exemple, requiert un multiple de 8 octets pour permettre le cryptage jusqu'à la fin du paquet. Cela permet d'utiliser l'algorithme DES standard plutôt que de recourir à une version spéciale d'algorithme de cryptage.

En l'absence de cryptage, ce champ ne doit pas être utilisé pour acheminer des données valables mais peut être présent.

**E\_CRC\_32**: champ codé sur 32 bits qui contient la valeur du code CRC pour laquelle la sortie des registres du décodeur défini dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 a pour valeur zéro après traitement de l'ensemble de la partie décryptée de la section splice\_info\_section. Ce champ a pour objet de donner une indication du fait que le décryptage a abouti. Donc la valeur zéro en sortie est obtenue après décryptage et par traitement des champs splice\_command\_type via le code E\_CRC\_32.

**CRC\_32**: champ codé sur 32 bits qui contient la valeur du code CRC pour laquelle la sortie des registres du décodeur défini dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 a pour valeur zéro après traitement de l'ensemble de la section splice\_info\_section, c'est-à-dire du champ table\_id au code CRC\_32. Le traitement du code CRC\_32 aura lieu avant le décryptage des champs cryptés et doit utiliser les champs cryptés dans leur état crypté.

### 7.3 Commandes de raccordement

#### 7.3.1 splice\_null()

La commande splice\_null() est prévue pour les futures extensions de la présente Recommandation. La commande splice\_null() permet l'envoi d'une table splice\_info\_table qui peut acheminer de nouveaux descripteurs sans qu'il y ait lieu d'envoyer l'une des autres commandes définies. Cette commande peut également être utilisée en tant que "message de pulsation" pour surveiller l'intégrité de l'équipement d'insertion de messages de repérage et l'intégrité de la liaison. Voir Tableau 7-3.

**Tableau 7-3/J.181 – splice\_null()**

| Syntaxe              | Nombre de bits | Mnémonique |
|----------------------|----------------|------------|
| splice_null() {<br>} |                |            |

### 7.3.2 splice\_schedule()

La commande splice\_schedule() a pour but de permettre l'acheminement à l'avance d'un horaire d'événements de raccordement. Voir Tableau 7-4.

**Tableau 7-4/J.181 – splice\_schedule ()**

| Syntaxe  | Nombre de bits  | Mnémonique  |
|--|---|---|
| splice_schedule() {<br><b>splice_count</b><br>for (i = 0; i < splice_count; i++) {<br><b>splice_event_id</b><br><b>splice_event_cancel_indicator</b><br><b>reserved</b><br>if (splice_event_cancel_indicator == '0') {<br><b>out_of_network_indicator</b><br><b>program_splice_flag</b><br><b>duration_flag</b><br><b>reserved</b><br>if (program_splice_flag == '1')<br><b>utc_splice_time</b><br>if (program_splice_flag == '0') {<br><b>component_count</b><br>for (j = 0; j < component_count; j++) {<br><b>component_tag</b><br><b>utc_splice_time</b><br>}<br>}<br>}<br>if (duration_flag)<br>break_duration()<br><b>unique_program_id</b><br><b>avail_num</b><br><b>avails_expected</b><br>}<br>} | 8<br><br>32<br>1<br>7<br><br>1<br>1<br>1<br>5<br><br>32<br><br>8<br><br>8<br>32<br><br><br>16<br>8<br>8 | <b>uimsbf</b><br><br><b>uimsbf</b><br><b>bslbf</b><br><b>bslbf</b><br><br><b>bslbf</b><br><b>bslbf</b><br><b>bslbf</b><br><b>bslbf</b><br><br><b>uimsbf</b><br><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b> |

#### 7.3.2.1 Définition sémantique des champs de splice\_schedule()

**splice\_count** (nombre de raccordements): entier non signé codé sur 8 bits qui indique le nombre d'événements de raccordement spécifiés dans la boucle qui suit.

**splice\_event\_id** (identificateur d'événement de raccordement): identificateur unique, codé sur 32 bits, d'événement de raccordement.

**splice\_event\_cancel\_indicator** (indicateur d'annulation d'événement de raccordement): fanion codé sur 1 bit qui, mis à "1", indique qu'un événement de raccordement précédemment envoyé, identifié par `splice_event_id`, a été annulé.

**out\_of\_network\_indicator** (indicateur de sortie de réseau): fanion codé sur 1 bit qui, mis à "1", indique que l'événement de raccordement est une occasion de sortir de l'alimentation réseau et que la valeur de `utc_splice_time` doit se référer à un point de sortie ou à un point de sortie de programme visé. Lorsqu'il est mis à zéro, le fanion indique que l'événement de raccordement est une occasion de revenir à l'alimentation réseau et que la valeur de `utc_splice_time` doit se référer à un point d'entrée ou à un point d'entrée de programme visé.

**program\_splice\_flag** (fanion de raccordement de programme): fanion codé sur 1 bit qui, mis à "1", indique que le message se réfère à un point de raccordement de programme et que le mode est le mode raccordement de programme, pour lequel tous les identificateurs PID ou composantes de programme doivent être raccordés. Lorsqu'il est mis à zéro, ce champ indique que le mode est le mode raccordement de composantes, pour lequel chaque composante destinée à être raccordée sera énumérée séparément dans la syntaxe qui suit.

**duration\_flag** (fanion de durée): fanion codé sur 1 bit qui indique la présence du champ `break_duration()`.

**utc\_splice\_time** (instant de raccordement UTC): entier non signé codé sur 32 bits qui représente l'instant de l'événement de raccordement signalé sous forme de nombre de secondes suivant 00 heures UTC, le 6 janvier 1980, compte tenu des secondes intercalaires. L'instant `utc_splice_time` peut être converti en heure UTC sans utiliser la valeur de `GPS_UTC_offset` fournie par la table de temps du système. Le champ **utc\_splice\_time** est utilisé uniquement dans la commande `splice_schedule()`.

**component\_count** (nombre de composantes): entier non signé codé sur 8 bits qui spécifie le nombre d'instances de données de flux PID élémentaires dans la boucle qui suit. Les composantes sont équivalentes aux flux PID élémentaires.

**component\_tag** (étiquette de composante): valeur codée sur 8 bits qui identifie le flux PID élémentaire contenant le point de raccordement spécifié par la valeur du champ `splice_time()` qui suit. La valeur doit être la même que celle utilisée dans le champ `stream_identification_descriptor()` pour identifier ce flux PID élémentaire.

**unique\_program\_id** (identificateur de programme unique): cette valeur devrait donner une identification unique pour un événement de visionnage au sein du service. Etant donné que la présente Recommandation est censée prendre en compte toute norme connexe, la valeur doit être indépendante de toute autre norme. Ce champ a la même fonction que le champ `event_id` de la norme ATSC ou DVB.

**avail\_num** (espace disponible, précédemment désigné sous le terme "avail"): ce champ établit l'identification d'un espace disponible spécifique dans un champ `unique_program_id`. Cette valeur devrait être incrémentée à chaque nouvel espace disponible dans un événement de visionnage. Elle devrait être remise à un pour le premier espace disponible d'un nouvel événement de visionnage. Ce champ devrait être incrémenté à chaque nouvel espace disponible. Facultativement, il peut acheminer la valeur zéro pour indiquer sa non-utilisation.

**avails\_expected** (nombre d'espaces disponibles, précédemment désigné sous le terme "avail\_count"): ce champ donne le nombre total escompté d'espaces disponibles individuels dans l'événement de visionnage en cours. Lorsque ce champ est à zéro, il indique que le champ `avail_num` n'a pas de signification.

### 7.3.3 splice\_insert()

La commande splice\_insert() doit être envoyée au moins une fois pour chaque événement de raccordement. Se référer au § 5.3 pour l'utilisation de ce message. Voir aussi le Tableau 7-5.

Tableau 7-5/J.181 – splice\_insert()

| Syntaxe   | Nombre de bits | Mnémonique    |
|---|----------------|---------------|
| splice_insert() {   |                |               |
| <b>splice_event_id</b>  | <b>32</b>      | <b>uimsbf</b> |
| <b>splice_event_cancel_indicator</b>                                | <b>1</b>       | <b>bslbf</b>  |
| <b>reserved</b>   | <b>7</b>       | <b>bslbf</b>  |
| if (splice_event_cancel_indicator == "0") {                         |                |               |
| <b>out_of_network_indicator</b>                                     | <b>1</b>       | <b>bslbf</b>  |
| <b>program_splice_flag</b>  | <b>1</b>       | <b>bslbf</b>  |
| <b>duration_flag</b>  | <b>1</b>       | <b>bslbf</b>  |
| <b>splice_immediate_flag</b>  | <b>1</b>       | <b>bslbf</b>  |
| <b>reserved</b>   | <b>4</b>       | <b>bslbf</b>  |
| If ((program_splice_flag == "1") && (splice_immediate_flag == "0")) |                |               |
| splice_time()   |                |               |
| if (program_splice_flag == "0") {                                   |                |               |
| <b>component_count</b>  | <b>8</b>       | <b>uimsbf</b> |
| for (i = 0; i < component_count; i++) {                             |                |               |
| <b>component_tag</b>  | <b>8</b>       | <b>uimsbf</b> |
| if (splice_immediate_flag == "0")                                   |                |               |
| splice_time()   |                |               |
| }   |                |               |
| }   |                |               |
| if (duration_flag == "1")   |                |               |
| break_duration()  |                |               |
| <b>unique_program_id</b>  | <b>16</b>      | <b>uimsbf</b> |
| <b>avail_num</b>  | <b>8</b>       | <b>uimsbf</b> |
| <b>avails_expected</b>  | <b>8</b>       | <b>uimsbf</b> |
| }   |                |               |
| }   |                |               |

#### 7.3.3.1 Définition sémantique des champs de splice\_insert()

**splice\_event\_id** (identificateur d'événement de raccordement): identificateur unique, codé sur 32 bits, d'événement de raccordement.

**splice\_event\_cancel\_indicator** (indicateur d'annulation d'événement de raccordement): fanion codé sur 1 bit qui, mis à "1", indique qu'un événement de raccordement précédemment envoyé, identifié par le champ splice\_event\_id, a été annulé.

**out\_of\_network\_indicator** (indicateur de sortie de réseau): fanion codé sur 1 bit qui, mis à "1", indique que l'événement de raccordement est une occasion de sortir de l'alimentation réseau et que la valeur de `splice_time()`, telle que modifiée par le champ `pts_adjustment`, doit se référer à un point de sortie ou à un point de sortie de programme visé. Lorsqu'il est mis à "0", le fanion indique que l'événement de raccordement est une occasion de revenir à l'alimentation réseau et que la valeur de `splice_time()`, telle que modifiée par le champ `pts_adjustment`, doit se référer à un point d'entrée ou à un point d'entrée de programme visé.

**program\_splice\_flag** (fanion de raccordement de programme): fanion codé sur 1 bit qui, mis à "1", indique que le message concerne un point de raccordement de programme et que le mode est le mode raccordement de programme, pour lequel tous les identificateurs PID ou composantes de programme doivent être raccordés. Lorsqu'il est mis à "0", ce champ indique que le mode est le mode raccordement de composantes, pour lequel chaque composante qu'il est prévu de raccorder sera énumérée séparément dans la syntaxe qui suit.

**duration\_flag** (fanion de durée): fanion codé sur 1 bit qui, mis à "1", indique la présence du champ `break_duration()`.

**splice\_immediate\_flag** (fanion de raccordement immédiat): fanion qui, mis à "1", indique d'une part l'absence du champ `splice_time()` et d'autre part que le mode de raccordement doit être le mode raccordement immédiat, pour lequel le dispositif de raccordement doit choisir la possibilité de raccordement la plus rapprochée dans le flux, par rapport au paquet d'informations de raccordement. Mis à "0", ce fanion indique la présence du champ `splice_time()` dans un emplacement au moins à l'intérieur de la commande `splice_insert()`.

**component\_count** (nombre de composantes): entier non signé codé sur 8 bits qui spécifie le nombre d'instances de données de flux PID élémentaires dans la boucle qui suit. Les composantes sont équivalentes aux flux PID élémentaires.

**component\_tag** (étiquette de composante): valeur codée sur 8 bits qui identifie le flux PID élémentaire contenant le point de raccordement spécifié par la valeur du champ `splice_time()` qui suit. La valeur doit être la même que celle utilisée dans le champ `stream_identification_descriptor()` pour identifier ce flux PID élémentaire.

**unique\_program\_id** (identificateur de programme unique): cette valeur devrait donner une identification unique pour un événement de visionnage au sein du service. Etant donné que la présente Recommandation est censée prendre en compte toute norme connexe, la valeur doit être indépendante de toute autre norme. Ce champ a la même fonction que le champ `event_id` de la norme ATSC ou DVB.

**avail\_num** (espace disponible, précédemment appelé "avail"): ce champ établit l'identification d'un espace disponible spécifique dans un champ `unique_program_id`. Cette valeur devrait être incrémentée à chaque nouvel espace disponible dans un événement de visionnage. Elle devrait être remise à un pour le premier espace disponible d'un nouvel événement de visionnage. Ce champ devrait être incrémenté à chaque nouvel espace disponible. Facultativement, il peut acheminer la valeur zéro pour indiquer sa non-utilisation.

**avails\_expected** (nombre d'espaces disponibles, précédemment appelé "avail\_count"): ce champ donne le nombre total escompté d'espaces disponibles individuels dans l'événement de visionnage en cours. Lorsque ce champ est à zéro, il indique que le champ `avail` n'a pas de signification.

### 7.3.4 time\_signal()

La commande `time_signal()` assure un mécanisme de fourniture de données synchronisé dans le temps. Sa syntaxe permet d'assurer la synchronisation entre les informations acheminées dans ce message et l'horloge de base du système (STC, *system time clock*). Bien que l'unique charge utile du message soit acheminé dans le descripteur, les capacités de syntaxe et de transport offertes aux messages `splice_insert()` le sont également à la commande `time_signal()`. L'acheminement peut cependant se faire dans un identificateur PID différent de celui qui achemine les autres messages de repérage utilisés pour la signalisation des points de raccordement.

Si le fanion `time_specified_flag` est mis à 0 (indiquant ainsi qu'il n'y a pas de champ `pts_time` dans le message), la commande doit être interprétée comme une commande à effet immédiat. Il faut bien comprendre que ce mode d'utilisation conduira à une erreur de précision dont l'amplitude n'est pas spécifiée.

Etant donné que la commande `time_signal()` utilise des descripteurs pour la plupart des informations spécifiques, sa longueur pourrait être supérieure à celle d'un paquet de transport MPEG. Il est fortement recommandé de conserver cette commande dans un seul paquet si possible. Cela peut ne pas être toujours possible dans des situations, par exemple, où la longueur de ces informations "uniques" est importante ou lorsqu'une autre spécification est utilisée pour la définition de ces informations. Voir Tableau 7-6.

Tableau 7-6/J.181 – time\_signal()

| Syntaxe  | Nombre de bits | Mnémonique |
|--|----------------|------------|
| <code>time_signal() {<br/>    splice_time()<br/>}</code> |                |            |

#### 7.3.4.1 Définition sémantique de time\_signal()

Cette commande `time_signal()` fournit une méthode uniforme permettant d'associer un champ `pts_time` à un ou plusieurs descripteurs arbitraires fournis par la syntaxe de la section `splice_info_section` (voir Tableau 7-1). Veuillez vous référer au § 8 relatif aux descripteurs de raccordement.

### 7.3.5 bandwidth\_reservation()

La commande `bandwidth_reservation()` permet de réserver une largeur de bande dans un multiplex. Citons comme utilisation type le cas d'un système de distribution par satellite dont les paquets d'un certain identificateur PID doivent toujours arriver suivant la fréquence de répétition voulue afin de garantir une certaine largeur de bande pour cet identificateur. Cette commande diffère d'une commande `splice_null()` de telle sorte à pouvoir être facilement traitée de manière unique par l'équipement de réception (c'est-à-dire supprimée du multiplex par le récepteur du satellite). Si un descripteur est envoyé avec cette commande, on ne peut pas s'attendre à ce qu'il soit acheminé à travers toute la chaîne de transmission; il devrait s'agir d'un descripteur privé utilisé uniquement par ce processus de réservation de largeur de bande. Voir Tableau 7-7.

Tableau 7-7/J.181 – bandwidth\_reservation()

| Syntaxe                                      | Nombre de bits | Mnémonique |
|--|----------------|------------|
| <code>bandwidth_reservation() {<br/>}</code> |                |            |

## 7.4 Instant

### 7.4.1 splice\_time()

La structure du champ splice\_time() spécifie, après modification par le champ pts\_adjustment, l'instant de l'événement de raccordement. Voir Tableau 7-8.

Tableau 7-8/J.181 – splice\_time()

| Syntaxe   | Bits             | Mnemonic  |
|---|------------------|---|
| splice_time() {<br><b>time_specified_flag</b><br>if (time_specified_flag == 1) {<br><b>reserved</b><br><b>pts_time</b><br>}<br>else<br><b>reserved</b><br>} | 1<br><br>6<br>33 | <b>bslbf</b><br><br><b>bslbf</b><br><b>uimsbf</b> |
|   | 7                | <b>bslbf</b>                                      |

#### 7.4.1.1 Définition sémantique des champs de splice\_time()

**time\_specified\_flag** (fanion d'instant spécifié): fanion codé sur 1 bit qui, mis à "1", indique la présence du champ pts\_time et des bits réservés associés.

**pts\_time** (instant PTS): champ codé sur 33 bits qui indique l'instant, en nombre de tops de l'horloge à 90 kHz du programme. Ce champ représente, après modification par le champ pts\_adjustment, l'instant du point de raccordement visé.

### 7.4.2 break\_duration()

La structure break\_duration() spécifie la durée de la ou des pauses publicitaires. Elle peut être utilisée pour indiquer au dispositif de raccordement quand la pause se terminera et quand le point d'entrée du réseau produira. Voir Tableau 7-9.

Tableau 7-9/J.181 – break\_duration()

| Syntaxe   | Nombre de bits   | Mnémonique  |
|---|------------------|---|
| break_duration() {<br><b>auto_return</b><br><b>reserved</b><br><b>duration</b><br>} | 1<br><br>6<br>33 | <b>bslbf</b><br><br><b>bslbf</b><br><b>uimsbf</b> |

#### 7.4.2.1 Définition sémantique des champs de break\_duration()

**auto\_return** (retour automatique): fanion codé sur 1 bit qui, mis à "1", indique que le champ duration doit être utilisé par le dispositif de raccordement pour savoir à quel moment doit se produire le retour à l'alimentation réseau (fin de la pause). Il n'est pas prévu d'envoyer une commande splice\_insert() dont l'indicateur out\_of\_network\_indicator est mis à 0 pour mettre fin à cette pause. Lorsque ce fanion est mis à "0", le champ duration, s'il est présent, n'est pas nécessaire pour mettre fin à la pause étant donné qu'une nouvelle commande splice\_insert() sera envoyée à la

fin de la pause. Dans ce cas, la présence du champ `break_duration` agit comme un mécanisme de sécurité dans l'éventualité où la commande `splice_insert()` serait perdue à la fin d'une pause.

**duration** (durée): champ codé sur 33 bits qui indique le temps écoulé, en nombre de tops de l'horloge à 90 kHz du programme.

## 7.5 Contraintes

### 7.5.1 Contraintes imposées au champ `splice_info_section()`

Le champ `splice_info_section` doit être acheminé dans un ou plusieurs flux PID propres à un programme et mentionnés dans la table PMT. Le ou les identificateurs PID de la section `splice_info_section` doivent être identifiés dans la table PMT par la valeur 0x86 du champ `stream_type`.

Le champ `splice_info_section` acheminé dans un ou plusieurs flux PID mentionnés dans la table PMT d'un programme doit contenir uniquement les informations sur les événements de raccordement qui ont lieu dans ce programme.

Un événement de raccordement doit être défini au moyen d'une valeur unique de l'identificateur `splice_event_id`.

En cas d'utilisation du mode raccordement de composantes, chaque flux PID élémentaire doit être identifié au moyen d'un descripteur `stream_identfier_descriptor` acheminé dans la boucle PMT, un par identificateur PID. Ce descripteur doit comporter une étiquette `component_tag` qui correspond à un seul flux PID parmi ceux que contient le programme et qui sont énumérés dans la table PMT de ce programme.

Tout identificateur `splice_event_id` qui est envoyé dans une section `splice_info_section` au moyen d'une commande `splice_schedule()` doit être envoyé à nouveau, avant l'événement, au moyen d'une commande `splice_insert()`. Dès lors, il y aura une correspondance entre les valeurs d'identificateur `splice_event_id` choisies pour des événements donnés signalés par la commande `splice_schedule()` (futur éloigné) et les valeurs d'identificateur `splice_event_id` utilisées dans la commande `splice_insert()` (futur proche) pour indiquer les mêmes événements.

Il n'est pas nécessaire d'envoyer les valeurs de l'identificateur `splice_event_id` par ordre croissant dans les messages subséquents, ni de les incrémenter chronologiquement. Les valeurs de l'identificateur `splice_event_id` peuvent être choisies au hasard. Lorsqu'on utilise la commande `splice_schedule()`, les valeurs de l'identificateur `splice_event_id` doivent être uniques sur la période de la commande `splice_schedule()`. Une valeur d'identificateur `splice_event_id` peut être réutilisée lorsque son instant `splice_time` associé est passé.

Lorsque le fanion `splice_immediate_flag` est mis à 1, l'instant auquel le raccordement doit avoir lieu est l'instant présent. Il s'agit du "mode raccordement immédiat". Lorsque celui-ci est utilisé avec la commande `splice_insert()`, le raccordement peut avoir lieu avant ou après l'occasion la plus proche qui est détectée par le dispositif de raccordement. On peut utiliser le mode raccordement immédiat tant pour les points d'entrée que pour les points de sortie du raccordement, c'est-à-dire pour les deux états de l'indicateur `out_of_network_indicator`.

La terminaison de tout espace disponible par un message de mode raccordement de programme, un message de mode raccordement de composantes ou l'absence de message (lorsque la durée `break_duration` est atteinte) devra être autorisée, indépendamment de la nature du message au début de l'espace disponible.

## 7.5.2 Contraintes imposées à l'interprétation du temps

### 7.5.2.1 Contraintes imposées au champ `splice_time()` pour la commande `splice_insert()`

Les contraintes suivantes imposées au champ `splice_time()` s'appliquent lorsque le type `splice_command_type` est égal à 0x05 (`splice_insert()`):

au moins un message relatif à un point de sortie de réseau doit arriver au minimum 4 secondes avant l'instant de raccordement signalé (instant `pts_time` tel que modifié par le champ `pts_adjustment`) si cet instant est spécifié. Un message de mode raccordement immédiat est autorisé pour un point de sortie de réseau, mais l'instant de raccordement effectif n'est pas défini et il est recommandé de n'utiliser les messages de mode raccordement immédiat que pour mettre prématurément fin aux pauses. En cas d'utilisation de messages de repérage en mode non immédiat raccordement pour des points d'entrée de réseau, le message de repérage doit parvenir au raccordeur avant l'arrivée dans le récepteur de l'image de point d'entrée signalée.

Un point de sortie est situé entre deux unités de présentation. Le point de sortie visé d'un événement de raccordement signalé doit être le point de sortie qui précède immédiatement l'unité de présentation dont l'instant de présentation est le plus proche de l'instant `pts_time` signalé tel que modifié par le champ `pts_adjustment`.

Un point d'entrée est situé entre deux unités de présentation. Le point d'entrée visé d'un événement de raccordement signalé doit être le point d'entrée qui précède immédiatement l'unité de présentation dont l'instant de présentation est le plus proche de l'instant `pts_time` signalé tel que modifié par le champ `pts_adjustment`.

Lorsque le mode raccordement de composantes est en vigueur et que l'indicateur `out_of_network_indicator` est à 1 (le début d'une pause), chaque composante énumérée dans la boucle de composantes `splice_insert()` doit être commutée de la composante réseau à la composante fournie par le dispositif de raccordement, à l'instant indiqué. Les composantes qui ne sont pas énumérées dans la boucle de composantes du message restent inchangées: si une composante produite par le dispositif de raccordement était la composante réseau, elle reste la composante réseau; si une composante produite par le dispositif de raccordement était une composante fournie par le dispositif de raccordement, elle reste la composante produite par le dispositif de raccordement.

Lorsque le mode raccordement de composantes est en vigueur et que l'indicateur `out_of_network_indicator` est à 0 (la fin d'une pause), chaque composante énumérée dans la boucle de composantes `splice_insert()` doit être commutée de la composante fournie par le dispositif de raccordement à la composante réseau, à l'instant indiqué. Les composantes qui ne sont pas énumérées dans la boucle de composantes du message restent inchangées. Si une composante produite par le dispositif de raccordement était la composante réseau, elle reste la composante réseau; si une composante produite par le dispositif de raccordement était une composante fournie par le dispositif de raccordement, elle reste la composante produite par le dispositif de raccordement.

Lorsque le mode raccordement de composantes est en vigueur et que le mode raccordement immédiat ne l'est pas, la première composante énumérée dans la boucle de composantes de la commande `splice_insert()` doit avoir un instant `pts_time` valable dans le champ `splice_time()` associé et cet instant `pts_time` est désigné comme étant l'instant `pts_time` par défaut. Les composantes subséquentes énumérées dans la boucle de composantes du même message qui n'ont pas d'instant `pts_time` associé doivent utiliser cet instant `pts_time` par défaut. Il faut permettre à toutes les composantes suivant la première composante énumérée d'une commande `splice_insert()` de contenir un instant `pts_time` unique qui soit différent de l'instant `pts_time` par défaut.

En mode raccordement de composantes, toutes les valeurs `pts_time` indiquées dans la boucle de composantes de la commande `splice_insert()` doivent être modifiées à l'aide du champ `pts_adjustment` pour obtenir les différentes valeurs voulues pour le point de sortie ou le point d'entrée signalée. Le champ `pts_adjustment`, fourni par tout dispositif générant ou modifiant une valeur de ce champ, doit s'appliquer à tous les champs `pts_time` dans le message.

### **7.5.2.2 Contraintes imposées au champ `break_duration()` pour la commande `splice_insert()`**

Les contraintes suivantes imposées au champ `break_duration()` s'appliquent lorsque le type `splice_command_type` est égal à 0x05 (raccordement – insertion):

La valeur indiquée dans le champ `break_duration()` est interprétée comme la durée prévue de la pause publicitaire. C'est un champ facultatif qu'il convient d'utiliser lorsque l'indicateur `out_of_network_indicator` est à 1. Ce champ peut être utilisé dans la commande `splice_insert()` qui spécifie l'instant de début de la pause, de telle manière que le dispositif de raccordement puisse calculer l'instant auquel la pause se terminera.

On peut mettre fin à une pause par l'émission d'une commande `splice_insert()` dont l'indicateur `out_of_network_indicator` est mis à 0. On peut donner un instant `splice_time()` ou utiliser le mode raccordement immédiat. Lorsqu'une durée `break_duration` a été indiquée au début de la pause (lorsque le champ `auto_return` a été mis à 0), la valeur du champ `break_duration` peut être utilisée comme un mécanisme de secours pour assurer qu'un retour au réseau se produira effectivement dans l'éventualité d'une perte de paquet de repérage.

On peut également mettre fin à une pause en donnant une durée de pause au début de la pause et en se fiant au dispositif de raccordement pour revenir à l'alimentation réseau au moment approprié. Le fanion `auto_return` doit être mis à 1. Ceci sera appelé le mode de retour automatique. Les pauses en mode de retour automatique ne requièrent pas et ne rejettent pas les messages de repérage en fin de pause dont l'indicateur `out_of_network_indicator` mis à 0. Un dispositif de réception ne devrait donc pas attendre un message de repérage à la fin d'une pause pour fonctionner correctement. Dans ce mode, on peut toutefois mettre fin prématurément à la pause. Pour cela, il faut donner une deuxième commande `splice_insert()` dans laquelle l'indicateur `out_of_network_indicator` est à 0. Le nouvel instant du raccordement de retour au réseau peut être donné par un instant `splice_time()` mis à jour ou par l'emploi du message mode raccordement immédiat. Un message de repérage dont l'indicateur `out_of_network_indicator` est mis à 0 doit toujours avoir priorité sur le champ `duration` d'un message de repérage précédent (dont l'indicateur `out_of_network_indicator` est à 1) si la durée signalée de la pause n'est encore dépassée.

## **8 Descripteurs de raccordement**

### **8.1 Aperçu général**

Le descripteur `splice_descriptor` est un prototype utilisé pour ajouter de nouveaux champs à la section `splice_info_section`. Tous les descripteurs inclus utilisent la même syntaxe pour les six premiers octets. Pour permettre l'adjonction d'informations privées, le code identificateur identifier a été inclus. Cela supprime la nécessité d'un descripteur d'enregistrement dans la boucle de descripteur.

Un équipement de réception doit ignorer tout descripteur ayant un identificateur inconnu ou une étiquette de descripteur inconnue. Dans le cas des descripteurs ayant un identificateur connu, l'équipement de réception doit ignorer les descripteurs ayant une étiquette `splice_descriptor_tag` inconnue.

On peut avoir des descripteurs de raccordement dans la section `splice_info_section` en vue d'extensions propres aux diverses commandes. Voir Tableau 8-1.

**Tableau 8-1/J.181 – Etiquettes de descripteur de raccordement**

| Etiquette | Descripteurs pour l'identificateur "CUEI"                   |
|-----------|---|
| 0x00      | avail_descriptor  |
| 0x01      | DTMF_descriptor   |
| 0x02      | segmentation_descriptor                                     |
| 0x03-0xFF | Réservé pour des descripteurs SCTE splice_descriptor futurs |

## 8.2 Descripteur de raccordement

La syntaxe du descripteur de raccordement donnée dans le présent paragraphe doit être utilisée comme un gabarit pour des implémentations spécifiques d'un descripteur destinées à la section splice\_info\_section. Il convient de noter que les descripteurs de raccordement sont utilisés uniquement dans une section splice\_info\_section. Ils ne doivent pas être utilisés dans une syntaxe MPEG telle que la table PMT, ou dans la syntaxe d'une autre norme. Cela permet de s'appuyer sur l'ensemble de la gamme d'étiquettes de descripteur pour établir de nouveaux descripteurs. Voir Tableau 8-2.

**Tableau 8-2/J.181 – splice\_descriptor()**

| Syntaxe   | Nombre de bits    | Mnémonique   |
|---|-------------------|--|
| splice_descriptor() {<br><b>splice_descriptor_tag</b><br><b>descriptor_length</b><br><b>identifiant</b><br>for (i = 0; i < N; i++) {<br><b>private_byte</b><br>}<br>} | 8<br>8<br>32<br>8 | <b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b> |

### 8.2.1 Définition sémantique des champs du descripteur splice\_descriptor()

**splice\_descriptor\_tag** (étiquette de descripteur de raccordement): ce nombre codé sur 8 bits définit la syntaxe pour les octets privés qui constituent le corps de ce descripteur. Les étiquettes de descripteur sont définies par le propriétaire du descripteur, tel que celui-ci est enregistré au moyen de l'identificateur.

**descriptor\_length** (longueur du descripteur): ce nombre codé sur 8 bits donne la longueur, en octets, du descripteur qui suit ce champ. Les descripteurs étant limités à 256 octets, cette valeur est limitée à 254.

**identifiant** (identificateur): champ de 32 bits défini aux § 2.6.8 et 2.6.9 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, pour l'identificateur format\_identifiant du descripteur registration\_descriptor(). Seules les valeurs d'identificateur enregistrées auprès de l'autorité d'enregistrement SMPTE (société à responsabilité limitée) et reconnues par cette autorité devraient être utilisées (voir la page <http://www.smp-te-ra.org/mpegreg.html>). L'utilisation de cet identificateur dans le descripteur ne doit concerner et identifier que les informations privées contenues dans le descripteur. Ce nombre codé sur 32 bits est utilisé pour identifier le propriétaire du descripteur. Le code 0x43554549 ("CUEI" en ASCII) a été enregistré auprès de la SMPTE pour les descripteurs définis dans la présente Recommandation.

**private\_byte** (octet privé): le reste du descripteur est réservé aux champs de données requis par le descripteur en cours de définition.

### 8.3 Descripteurs de raccordement spécifiques

#### 8.3.1 avail\_descriptor()

Ce descripteur d'espace disponible est une implémentation d'un descripteur splice\_descriptor. Il offre une extension facultative à la commande splice\_insert() qui permet d'envoyer un identificateur d'autorisation pour un espace disponible. On peut inclure plusieurs copies de ce descripteur au moyen du mécanisme de boucle prévu. Cet identificateur est destiné à reproduire la fonctionnalité de système de tonalité de repérage dans des systèmes analogiques pour des insertions publicitaires. Ce descripteur doit être utilisé uniquement avec une commande splice\_insert() à l'intérieur d'une section splice\_info\_section. Voir Tableau 8-3.

**Tableau 8-3/J.181 – avail\_descriptor()**

| Syntaxe                      | Nombre de bits | Mnémonique    |
|------------------------------|----------------|---------------|
| avail_descriptor() {         |                |               |
| <b>splice_descriptor_tag</b> | 8              | <b>uimsbf</b> |
| <b>descriptor_length</b>     | 8              | <b>uimsbf</b> |
| <b>identifiant</b>           | 32             | <b>uimsbf</b> |
| <b>provider_avail_id</b>     | 32             | <b>uimsbf</b> |
| }                            |                |               |

##### 8.3.1.1 Définition sémantique des champs du descripteur avail\_descriptor()

**splice\_descriptor\_tag** (étiquette de descripteur de raccordement): ce nombre codé sur 8 bits définit la syntaxe pour les octets privés qui constituent le corps de ce descripteur. L'étiquette splice\_descriptor\_tag doit avoir la valeur 0x00.

**descriptor\_length** (longueur du descripteur): ce nombre codé sur 8 bits exprime la longueur, en octets, du descripteur qui suit ce champ. Le champ descriptor\_length doit avoir la valeur 0x08.

**identifiant** (identificateur): ce nombre codé sur 32 bits sert à identifier le propriétaire du descripteur. L'identificateur doit avoir la valeur 0x43554549 ("CUEI" en ASCII).

**provider\_avail\_id** (identificateur d'espace disponible de fournisseur): ce nombre codé sur 32 bits donne les informations qu'un dispositif de réception peut utiliser pour modifier son comportement dans ou en dehors d'un espace disponible. Il peut être utilisé de manière analogue à une tonalité de repérage analogique. Un exemple serait un réseau commandant une filiale ou une tête de réseau pour interrompre la diffusion d'un événement sportif.

#### 8.3.2 DTMF\_descriptor()

Le descripteur DTMF\_descriptor() est une implémentation d'un descripteur splice\_descriptor. Il offre une extension facultative à la commande splice\_insert() qui permet à un dispositif de réception de générer une ancienne séquence de tonalités DTMF analogiques sur la base de la section splice\_info\_section en train d'être reçue. Voir Tableau 8-4.

**Tableau 8-4/J.181 – DTMF\_descriptor()**

| Syntaxe   | Nombre de bits                   | Mnémonique   |
|---|----------------------------------|--|
| DTMF_descriptor() {<br><b>splice_descriptor_tag</b><br><b>descriptor_length</b><br><b>Identifiant</b><br><b>Preroll</b><br><b>dtmf_count</b><br><b>reserved</b><br>for (i = 0; i < <b>dtmf_count</b> ; i++) {<br><b>DTMF_char</b><br>}<br>} | 8<br>8<br>32<br>8<br>3<br>5<br>8 | <b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>bslbf</b><br><b>uimsbf</b> |

### 8.3.2.1 Définition sémantique des champs du descripteur DTMF\_descriptor()

**splice\_descriptor\_tag** (étiquette de descripteur de raccordement): ce nombre codé sur 8 bits définit la syntaxe pour les octets privés qui constituent le corps de ce descripteur. Il doit avoir pour valeur 0x01.

**descriptor\_length** (longueur du descripteur): ce nombre codé sur 8 bits donne la longueur, en octets, du descripteur qui suit ce champ.

**identifiant** (identificateur): ce nombre codé sur 32 bits est utilisé pour identifier le propriétaire du descripteur. Il doit avoir pour valeur 0x43554549 ("CUEI" en ASCII).

**preroll** (inertie): ce nombre codé sur 8 bits est le temps nécessaire, exprimé en dixièmes de seconde, pour présenter des tonalités DTMF au niveau de la sortie analogique du dispositif. Il est compris entre 0 et 25,5 secondes. La section des informations de raccordement doit être envoyée au moins deux secondes avant cette valeur. La valeur minimale suggérée du champ preroll est de 4 secondes.

**dtmf\_count** (nombre de caractères DTMF): la valeur de ce drapeau indique le nombre de caractères DTMF que le dispositif doit générer.

**DTMF\_char**: valeur ASCII associé à un chiffre compris entre "0" et "9" à "\*" ou à "#". Le dispositif doit utiliser ces valeurs pour générer une séquence DTMF au niveau d'une sortie analogique. La séquence doit s'achever avec pour dernier caractère envoyé l'horodate associée au champ preroll.

### 8.3.3 segmentation\_descriptor()

Ce descripteur de segmentation est une implémentation d'un descripteur splice\_descriptor. Il offre une extension facultative à la commande time\_signal() qui permet l'envoi de messages de segmentation de programme dans une méthode temps/vidéo précise. Ce descripteur peut être utilisé dans des commandes autres que la commande time\_signal().

Les dispositifs qui ne reconnaissent pas une valeur dans un champ quelconque doivent ignorer le message et ne déclencher aucune action. Voir Tableau 8-5.

**Tableau 8-5/J.181 – segmentation\_descriptor()**

| Syntaxe   | Nombre de bits | Mnémonique    |
|---|----------------|---------------|
| segmentation_descriptor() {                       |                |               |
| <b>splice_descriptor_tag</b>                      | <b>8</b>       | <b>uimsbf</b> |
| <b>descriptor_length</b>                          | <b>8</b>       | <b>uimsbf</b> |
| <b>identifiant</b>                                | <b>32</b>      | <b>uimsbf</b> |
| <b>segmentation_event_id</b>                      | <b>32</b>      | <b>uimsbf</b> |
| <b>segmentation_event_cancel_indicator</b>        | <b>1</b>       | <b>bslbf</b>  |
| <b>reserved</b>                                   | <b>7</b>       | <b>bslbf</b>  |
| if (segmentation_event_cancel_indicator == "0") { |                |               |
| <b>program_segmentation_flag</b>                  | <b>1</b>       | <b>bslbf</b>  |
| <b>segmentation_duration_flag</b>                 | <b>1</b>       | <b>bslbf</b>  |
| <b>reserved</b>                                   | <b>6</b>       | <b>bslbf</b>  |
| if (program_segmentation_flag == "0") {           |                |               |
| <b>component_count</b>                            | <b>8</b>       | <b>uimsbf</b> |
| for (i = 0; i < component_count; i++) {           |                |               |
| <b>component_tag</b>                              | <b>8</b>       | <b>uimsbf</b> |
| <b>reserved</b>                                   | <b>7</b>       | <b>bslbf</b>  |
| <b>pts_offset</b>                                 | <b>33</b>      | <b>uimsbf</b> |
| }   |                |               |
| }   |                |               |
| } if (segmentation_duration_flag == "1")          |                |               |
| segmentation_duration()                           |                |               |
| <b>segmentation_upid_type</b>                     | <b>8</b>       | <b>uimsbf</b> |
| <b>segmentation_upid_length</b>                   | <b>8</b>       | <b>uimsbf</b> |
| <b>segmentation_upid</b>                          |                | <b>uimsbf</b> |
| <b>segmentation_type_id</b>                       | <b>8</b>       | <b>uimsbf</b> |
| <b>chapter</b>                                    | <b>8</b>       | <b>uimsbf</b> |
| <b>chapter_count</b>                              | <b>8</b>       | <b>uimsbf</b> |
| }   |                |               |
| }   |                |               |

### 8.3.3.1 Définition sémantique des champs du descripteur segmentation\_descriptor()

**splice\_descriptor\_tag** (étiquette de descripteur de raccordement): ce nombre codé sur 8 bits définit la syntaxe pour les octets privés qui constituent le corps de ce descripteur. Il doit avoir pour valeur 0x02.

**descriptor\_length** (longueur du descripteur): ce nombre codé sur 8 bits donne la longueur, en octets, du descripteur qui suit ce champ.

**identifiant** (identificateur): ce nombre codé sur 32 bits est utilisé pour identifier le propriétaire du descripteur. Il doit avoir pour valeur 0x43554549 ("CUEI" en ASCII).

**segmentation\_event\_id** (identificateur d'événement de segmentation): identificateur unique d'événement de segmentation, codé sur 32 bits.

**segmentation\_event\_cancel\_indicator** (indicateur d'annulation d'événement de segmentation): fanion codé sur 1 bit qui, mis à "1", indique qu'un événement de segmentation précédemment envoyé, identifié par `segmentation_event_id`, a été annulé.

**program\_segmentation\_flag** (fanion de segmentation de programme): fanion codé sur 1 bit qui devrait être mis à "1" pour indiquer que le message se réfère à un point de segmentation de programme et que le mode est le mode segmentation de programme, pour lequel tous les identificateurs PID ou composantes de programme doivent être segmentés. Lorsqu'il est mis à zéro, ce champ indique que le mode est le mode segmentation de composantes, pour lequel chaque composante destinée à être segmentée sera énumérée séparément dans la syntaxe qui suit. Le fanion `program_segmentation_flag` peut être mis à des états différents pour différents messages de descripteurs au sein d'un programme.

**segmentation\_duration\_flag** (fanion de durée de segmentation): fanion codé sur 1 bit qui devrait être mis à "1" pour indiquer la présence du champ `segmentation_duration()`.

**component\_count** (nombre de composantes): entier non signé codé sur 8 bits qui spécifie le nombre d'instances de données de flux PID élémentaires dans la boucle qui suit. Les composantes sont équivalentes aux flux PID élémentaires.

**component\_tag** (étiquette de composante): valeur codée sur 8 bits qui identifie le flux PID élémentaire contenant le point de segmentation spécifié par la valeur du champ `splice_time()` qui suit. La valeur doit être la même que celle utilisée dans le champ `stream_identification_descriptor()` pour identifier ce flux PID élémentaire. La présence de ce champ provenant de la boucle des composantes indique la présence de cette composante dans la ressource considérée.

**pts\_offset** (compensation d'horodate PTS): entier non signé codé sur 33 bits qui doit être utilisé par un dispositif de raccordement comme une compensation à ajouter au champ `pts_time` dans le message `time_signal()` pour obtenir le ou les instants de raccordement voulus. Lorsque ce champ a la valeur zéro, le ou les champs `pts_time` doivent être utilisés sans compensation. Si le fanion `time_specified_flag` du champ `splice_time()` a pour valeur 0 ou si la commande dans laquelle ce descripteur est acheminé ne possède pas de champ `splice_time()`, ce champ doit être utilisé pour la compensation de l'instant de raccordement immédiat calculé.

**segmentation\_duration()** (durée de la segmentation): cette structure définit la durée du segment de programme. Elle peut être utilisée pour indiquer au raccordeur quand le segment sera terminé et quand le prochain message de segmentation arrivera. Il doit avoir la valeur 0 pour les messages de fin. Voir Tableau 8-6.

**Tableau 8-6/J.181 – segmentation\_duration()**

| Syntaxe  | Nombre de bits     | Mnémonique                               |
|--|--------------------|--|
| <pre>segmentation_duration() {     reserved     duration }</pre> | <p>7</p> <p>33</p> | <p><b>bslbf</b></p> <p><b>uimsbf</b></p> |

**duration** (durée): champ codé sur 33 bits qui indique le temps écoulé, en nombre de tops de l'horloge à 90 kHz du programme.

**segmentation\_upid\_type** (type d'identificateur de segmentation de programme unique): une des valeurs du tableau ci-dessous. Plusieurs types sont autorisés pour garantir que les programmeurs seront en mesure d'utiliser un identificateur pris en charge par leurs systèmes. Il est prévu que les utilisateurs de ces identificateurs disposeront d'une méthode hors bande pour collecter d'autres données relatives à ces nombres et que ceux-ci n'auront donc pas à être du même type. Ces

identificateurs peuvent se trouver dans d'autres descripteurs dans le programme et lorsqu'un même identificateur est utilisé (V-ISAN par exemple), il doit y avoir correspondance entre les programmes. En cas d'utilisation avec une insertion de raccordement, on ne peut pas imposer ou supposer que l'identificateur **segmentation\_upid** corresponde à l'identificateur **unique\_program\_id** du message splice\_insert. Voir Tableau 8-7.

**Tableau 8-7/J.181 – segmentation\_upid\_type**

| Type      | Longueur en octets | Nom                      | Description  |
|-----------|--------------------|--------------------------|--|
| 0x00      | 0                  | Inutilisé                | L'identificateur <b>segmentation_upid</b> n'est pas défini et n'est pas présent dans le descripteur.   |
| 0x01      | variable           | Défini par l'utilisateur | L'identificateur <b>segmentation_upid</b> ne suit pas un système de nommage normalisé.   |
| 0x02      | 8                  | ISCI                     | L'identificateur <b>segmentation_upid</b> est un code à 8 caractères généralement utilisé pour un contenu publicitaire.  |
| 0x03      | 12                 | Ad-ID                    | Défini par le Advertising Digital Identification (société à responsabilité limitée). Il s'agit essentiellement d'un identificateur à 12 caractères <a href="http://www.Ad-ID.org">http://www.Ad-ID.org</a> |
| 0x04      | 24                 | UMID                     | SMPTE 330M   |
| 0x05      | 8                  | ISAN                     | Norme ISO pour un équipement audiovisuel utilisant 16 chiffres hexadécimaux.<br><a href="http://www.nlc-bnc.ca/iso/tc46sc9/standard/15706e.htm">http://www.nlc-bnc.ca/iso/tc46sc9/standard/15706e.htm</a>  |
| 0x06      | 12                 | V-ISAN                   | Extension de la norme ISAN qui adjoint 8 caractères hexadécimaux pour un identificateur de version.  |
| 0x07      | 12                 | TID                      | Identificateur de programme Tribune Media Systems Program. 12 caractères, 2 lettres suivies par 10 chiffres.   |
| 0x08      | 8                  | Origine                  | Identificateur d'origine (Turner).   |
| 0x09-0x1F | variable           | Réservé                  | Réservé pour une normalisation future.   |

**segmentation\_upid\_length** (longueur de l'identificateur de segmentation de programme unique): longueur en octets de l'identificateur **segmentation\_upid**.

**segmentation\_upid** (identificateur de segmentation de programme unique): longueur et identification définies dans le Tableau 8-7. Le contenu et la longueur de ce champ sont déterminés par les champs **segmentation\_upid\_type** et **segmentation\_upid\_length**. On peut citer l'exemple du type 0x06 pour l'identificateur V-ISAN long de 12 octets. Ce champ contient l'identificateur V-ISAN pour le contenu auquel se réfère le descripteur considéré.

**segmentation\_type\_id** (identificateur de type de segmentation): cette valeur codée sur 8 bits doit contenir une des valeurs du Tableau 8-8 pour désigner le type de segmentation. Toutes les valeurs inutilisées sont réservées.

**Tableau 8-8/J.181 – segmentation\_type\_id**

| Message de segmentation               | Segmentation_type_id |
|---------------------------------------|----------------------|
| Début du programme                    | 0x10                 |
| Fin du programme                      | 0x11                 |
| Terminaison anticipée du programme    | 0x12                 |
| Interruption du programme             | 0x13                 |
| Reprise du programme                  | 0x14                 |
| Dépassement planifié du programme     | 0x15                 |
| Dépassement non planifié du programme | 0x16                 |
| Début du chapitre                     | 0x20                 |
| Fin du chapitre                       | 0x21                 |
| Début d'une publicité nationale       | 0x30                 |
| Fin d'un publicité nationale          | 0x31                 |
| Début d'une publicité locale          | 0x32                 |
| Fin d'une publicité locale            | 0x33                 |
| Unscheduled_event_start               | 0x40                 |
| Unscheduled_event_end                 | 0x41                 |

**chapter** (chapitre): ce champ établit l'identification d'un chapitre spécifique dans un identificateur `segmentation_unique_program_id`. Cette valeur devrait être incrémentée à chaque nouveau chapitre dans un événement de segmentation. Elle devrait être remise à un pour le premier chapitre d'un nouvel événement de visionnage. Ce champ devrait être incrémenté à chaque nouveau chapitre.

**chapter\_count** (nombre de chapitres): ce champ donne le nombre total escompté de chapitres individuels dans l'événement de segmentation en cours.

## 9 Cryptage

### 9.1 Aperçu général

La section `splice_info_section` prend en charge le cryptage d'une partie de la section afin qu'il soit possible d'empêcher l'accès à un espace disponible, sauf aux récepteurs qui sont autorisés pour ces espaces. Le présent paragraphe décrit les divers algorithmes de cryptage pouvant être utilisés. Le cryptage de la section est facultatif, tout comme l'implémentation du cryptage par le créateur du message ou par tout dispositif de réception. L'emploi du cryptage est réputé facultatif afin de permettre aux fabricants d'expédier des systèmes "en clair" sans avoir à se soucier de l'exportation de la technologie de cryptage. Si le cryptage est inclus dans le système, tout dispositif de réception doit implémenter tous les algorithmes énumérés dans la présente Recommandation, ce qui permet au créateur d'une table d'informations de raccordement d'utiliser n'importe lequel des algorithmes dans une transmission. L'emploi d'une technologie de cryptage privée est facultative et ne relève pas de la présente Recommandation.

### 9.2 Cryptage par clé fixe

Le cryptage utilisé dans la présente Recommandation part du principe de l'utilisation d'une clé fixe. La même clé est communiquée à l'émetteur et au récepteur. La méthode de remise de la clé à toutes les parties n'est pas spécifiée. Dans la présente Recommandation, jusqu'à 256 clés différentes sont prévues pour le décryptage. Le champ `cw_index` est utilisé pour déterminer la clé qu'il convient d'utiliser pour le décryptage d'une section. La longueur de la clé fixe dépend du type d'algorithme

utilisé. On part de l'hypothèse que la clé fixe remise à toutes les parties aura la longueur correcte pour l'algorithme qu'il est prévu d'utiliser.

### 9.3 Algorithmes de cryptage

Le champ `encryption_algorithm` de la section `splice_info_section` contient une valeur codée sur 6 bits, qui peut être une des valeurs montrées dans le Tableau 9-1. Toutes les variantes de la norme de chiffrement des données (DES, *data encryption standard*) utilisent une clé de 64 bits (en fait 56 bits plus une somme de contrôle) pour crypter ou décrypter un bloc de 8 octets. Dans le cas de l'algorithme DES triple, il faut trois clés de 64 bits, une pour chaque passage de l'algorithme DES. L'algorithme DES triple "standard" utilise en fait deux clés, la première et la troisième étant identiques. Voir les publications FIPS PUB 46-3 et FIPS PUB 81.

**Tableau 9-1/J.181 – Algorithme de cryptage**

| Valeur | Algorithme de cryptage     |
|--------|----------------------------|
| 0      | Pas de cryptage            |
| 1      | Mode DES – ECB             |
| 2      | Mode DES – CBC             |
| 3      | Mode DES EDE3 – ECB triple |
| 4-31   | Réservé                    |
| 32-63  | Privé de l'utilisateur     |

#### 9.3.1 Mode DES – ECB

Ce algorithme utilise la norme "Data Encryption Standard" (voir la publication FIPS PUB 81), en mode dictionnaire.

Pour utiliser ce type de cryptage, les données cryptées doivent contenir un multiple de 8 octets de données, des champs `splice_command_type` à `E_CRC_32`. La boucle `alignment_stuffing` peut être utilisée pour ajouter tout octet éventuellement nécessaire.

#### 9.3.2 Mode DES – CBC

Cet algorithme utilise la norme de chiffrement "Data Encryption Standard" (voir la publication FIPS PUB 81), en mode chaînage de bloc de chiffrement. L'algorithme de base est identique à celui du mode DES ECB. Chaque bloc de texte en clair de 64 bits subit une opération OU-exclusif au niveau du bit avec le bloc de texte chiffré précédent avant d'être crypté au moyen de la clé DES. Le premier bloc subit une opération OU-exclusif avec un vecteur initial. Dans le cadre de la présente Recommandation, le vecteur initial doit avoir une valeur fixe égale à zéro.

Pour utiliser ce type de cryptage, les données cryptées doivent contenir un multiple de 8 octets de données, des champs `splice_command_type` à `E_CRC_32`. La boucle `alignment_stuffing` peut être utilisée pour ajouter tout octet éventuellement nécessaire.

#### 9.3.3 Mode DES EDE3 – ECB triple

Cet algorithme utilise trois clés de 64 bits, utilisées chacune pour un passage d'algorithme DES – ECB (voir la publication FIPS PUB 46-3). Au niveau du dispositif d'émission, chaque bloc de données est d'abord crypté avec la première clé, décrypté avec la deuxième et enfin crypté avec la troisième. Sur le site de réception, chaque bloc est d'abord décrypté avec la troisième clé, crypté avec la deuxième et enfin décrypté avec la première.

Pour utiliser ce type de cryptage, les données cryptées doivent contenir un multiple de 8 octets de données, des champs `splice_command_type` à `E_CRC_32`. La boucle `alignment_stuffing` peut être utilisée pour ajouter tout octet éventuellement nécessaire.

### **9.3.4 Algorithmes privés d'utilisateur**

La présente Recommandation autorise l'utilisation d'algorithmes de cryptage privés. Elle ne spécifie pas la manière dont les dispositifs d'émission et de réception conviennent de l'algorithme à utiliser pour le code privé d'utilisateur. Par ailleurs, elle ne spécifie pas la manière dont la coordination des valeurs privées du champ `encryption_algorithm` devrait être enregistrée ou gérée.





## SÉRIES DES RECOMMANDATIONS UIT-T

|                |   |
|----------------|---|
| Série A        | Organisation du travail de l'UIT-T  |
| Série D        | Principes généraux de tarification  |
| Série E        | Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains                                      |
| Série F        | Services de télécommunication non téléphoniques   |
| Série G        | Systèmes et supports de transmission, systèmes et réseaux numériques  |
| Série H        | Systèmes audiovisuels et multimédias  |
| Série I        | Réseau numérique à intégration de services  |
| <b>Série J</b> | <b>Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias</b>                               |
| Série K        | Protection contre les perturbations   |
| Série L        | Construction, installation et protection des câbles et autres éléments des installations extérieures                                      |
| Série M        | RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux |
| Série N        | Maintenance: circuits internationaux de transmission radiophonique et télévisuelle  |
| Série O        | Spécifications des appareils de mesure  |
| Série P        | Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux   |
| Série Q        | Commutation et signalisation  |
| Série R        | Transmission télégraphique  |
| Série S        | Equipements terminaux de télégraphie  |
| Série T        | Terminaux des services télématiques   |
| Série U        | Commutation télégraphique   |
| Série V        | Communications de données sur le réseau téléphonique  |
| Série X        | Réseaux de données et communication entre systèmes ouverts  |
| Série Y        | Infrastructure mondiale de l'information, protocole Internet et réseaux de nouvelle génération  |
| Série Z        | Langages et aspects généraux logiciels des systèmes de télécommunication  |