Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.271

(05/2006)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Codage des images vidéo animées

Messages vidéo par voie de retour pour la transmission d'informations d'état et de demandes d'un récepteur vidéo à un émetteur vidéo

Recommandation UIT-T H.271

RECOMMANDATIONS UIT-T DE LA SÉRIE H

SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

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

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

Recommandation UIT-T H.271

Messages vidéo par voie de retour pour la transmission d'informations d'état et de demandes d'un récepteur vidéo à un émetteur vidéo

Résumé

La présente Recommandation spécifie le format des messages par voie de retour pour la transmission d'informations d'état et de demandes d'un récepteur vidéo à un émetteur vidéo.

La syntaxe des messages a été conçue de manière générique pour qu'elle puisse être utilisée avec la majorité des normes de codage vidéo internationale existantes. L'application des messages génériques aux Recommandations UIT-T H.261, H.263 et H.264 | ISO/CEI 14496-10 est spécifiée.

Source

La Recommandation UIT-T H.271 a été approuvée le 29 mai 2006 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

Mots clés

Message par voie de retour, retour du récepteur, sélection d'image de référence, vidéo.

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 avait 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 développeurs de consulter la base de données des brevets du TSB sous http://www.itu.int/ITU-T/ipr/.

TABLE DES MATIÈRES

Introduction

0.1 Objet

Pour certaines applications, la transmission de données additionnelles (dans la bande ou hors bande) est utile pour améliorer la qualité vidéo du service. Des charges utiles de données à utiliser pour diverses techniques de codage vidéo sont spécifiées dans la présente Recommandation, et ce de façon générique pour qu'elles soient applicables aux principales normes de codage vidéo existantes. L'application des messages génériques aux Recommandations H.261, H.263 et H.264 est spécifiée.

0.2 Aperçu général

Les informations suivantes peuvent être signalées d'un récepteur vidéo à un émetteur vidéo à l'aide d'un ou plusieurs des messages de retour définis dans la présente Recommandation:

– rapports d'état:
  - une ou plusieurs images sans erreur d'adaptation de flux binaire détectée;
  - pertes au niveau image et/ou macrobloc;
  - informations relatives à d'importantes données d'en-tête.

– Demandes de mise à jour:
  - demande de "réinitialisation" indiquant que l'émetteur devrait procéder à un rafraîchissement complet du flux binaire vidéo comme si aucune donnée de flux binaire préalable n'avait été reçue.

L'application aux Recommandations UIT-T H.261, H.263 et H.264 des messages par voie de retour est spécifiée.

Recommandation UIT-T H.271

Messages vidéo par voie de retour pour la transmission d'informations d'état et de demandes d'un récepteur vidéo à un émetteur vidéo

1 Domaine d'application

La présente Recommandation donne la spécification des messages par voie de retour pour le codage vidéo par blocs.

2 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.261 (1993), *Codec vidéo pour services audiovisuels à $p \times 64$ kbit/s*.

– Recommandation UIT-T H.263 (2005), *Codage vidéo pour communications à faible débit*.

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

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

3 Définitions

La présente Recommandation définit les termes suivants:

3.1 **voie de retour**: moyen d'acheminer des messages par voie de retour d'un récepteur de flux binaire vidéo à un émetteur de flux binaire vidéo.

3.2 **message par voie de retour**: message qui est généré par un récepteur de *flux binaire* vidéo et qui achemine des informations d'état ou des demandes du récepteur.

3.3 **flux binaire**: séquence de bits formant la représentation d'images *codées* et de données associées constituant une ou plusieurs *séquences vidéo codées*.

3.4 **erreur de flux binaire**: *flux binaire* erroné ou incomplet.

3.5 **erreur d'adaptation de flux binaire**: différence entre les valeurs ou le nombre d'*images* décodées résultant d'un flux binaire avec une ou plusieurs *erreurs de flux binaire* et les valeurs ou le nombre d'*images* décodées générées par l'application du processus de décodage à un *flux binaire* sans *erreur de flux binaire*.

3.6 **bloc**: matrice MxN (M colonnes par N lignes) d'échantillons luma et des échantillons chroma associés.

3.7 **erreur de flux binaire détectée**: *erreur de flux binaire* détectée par le récepteur.

3.8 **erreur d'adaptation de flux binaire détectée**: *erreur d'adaptation de flux binaire* pouvant être présente en raison d'*erreurs de flux binaire*.

3.9 ordre de décodage: ordre suivant lequel les *éléments syntaxiques* sont traités par le processus de décodage spécifié par une technique de codage vidéo.

3.10 macrobloc: *bloc* 16x16 d'échantillons luma et deux blocs correspondants d'échantillons chroma.

3.11 ensemble de paramètres: structure syntaxique contenant un certain nombre d'*éléments syntaxiques* pouvant être utilisés dans le processus de décodage pour une ou plusieurs *images*.

3.12 image: terme collectif désignant une trame ou une image vidéo codée en tant qu'unité distincte à l'aide d'une technique de codage vidéo.

3.13 image de référence: *image* contenant des échantillons pouvant être utilisés à des fins de prédiction interimage dans le processus de décodage d'*images* ultérieures du flux binaire vidéo.

3.14 réservé: lorsqu'il est utilisé dans les paragraphes spécifiant certaines valeurs d'un élément syntaxique particulier, ce terme indique une utilisation future par l'UIT-T. Ces valeurs ne doivent pas être utilisées dans des messages par voie de retour conformes à la présente Recommandation, mais pourront l'être dans des extensions futures par l'UIT-T de cette dernière.

3.15 élément syntaxique: élément de données représenté dans le *flux binaire* ou dans un message de retour.

4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

CRC code de redondance cyclique

LSB bit de plus faible poids (*least significant bit*)

MSB bit de plus fort poids (*most significant bit*)

5 Conventions

Dans la présente Recommandation, les remarques précédées de la mention "NOTE –" sont informatives et ne font pas partie intégrante de la présente Recommandation.

NOTE – Les opérateurs mathématiques utilisés dans la présente Recommandation sont similaires à ceux utilisés dans le langage de programmation C. Les conventions de numérotage et de comptage commencent généralement à partir de 0.

5.1 Opérateurs arithmétiques

Les opérateurs arithmétiques ci-après sont définis comme suit:

+ Addition

– Soustraction (opérateur binaire) ou négation (opérateur unaire de préfixe)

* Multiplication

/ Division entière avec troncature du résultat tendant vers zéro. Par exemple, les valeurs 7/4 et (–7) / (–4) sont tronquées à 1 et les valeurs (–7) / 4 et 7 / (–4) sont tronquées à –1.

x % y Modulo (congruence). Reste de x divisé par y, défini seulement pour des entiers x et y avec $x \geq 0$ et $y > 0$.

Lorsque l'ordre de priorité n'est pas indiqué explicitement par l'utilisation de parenthèses, les règles suivantes s'appliquent:

– les opérations de multiplication et division sont censées être effectuées avant les additions et les soustractions;

– les opérations de multiplication et de division apparaissant en séquence sont effectuées séquentiellement de gauche à droite;
– les opérations d'addition et de soustraction apparaissant en séquence sont effectuées séquentiellement de gauche à droite.

5.2 Opérateurs logiques

Les opérateurs logiques ci-après sont définis comme suit:

x && y opérateur logique "et" appliqué à x et y

x || y opérateur logique "ou" appliqué à x et y

! opérateur logique "non"

x ? y: z si x est "VRAI" ou non égal à 0, évaluation de la valeur de y; sinon, évaluation de la valeur de z.

5.3 Opérateurs relationnels

Les opérateurs relationnels suivants sont définis comme suit:

> supérieur à

>= supérieur ou égal à

< inférieur à

<= inférieur ou égal à

== égal à

!= différent de

5.4 Opérateurs binaires

Les opérateurs binaires suivants sont définis comme suit:

& "et" binaire. Lorsqu'il agit sur des arguments entiers, il agit sur une représentation du complément à deux de la valeur entière. Lorsqu'il agit sur un argument binaire qui contient moins de bits qu'un autre argument, l'argument le plus court est élargi en ajoutant plus de bits significatifs égaux à 0.

| "ou" binaire. Lorsqu'il agit sur des arguments entiers, il agit sur une représentation du complément à deux de la valeur entière. Lorsqu'il agit sur un argument binaire qui contient moins de bits qu'un autre argument, l'argument le plus court est élargi en ajoutant plus de bits significatifs égaux à 0.

^ Ou exclusif bit à bit. Lorsqu'il est appliqué à des arguments entiers, cet opérateur agit sur la représentation en complément à deux des valeurs entières. Lorsqu'il est appliqué à des chaînes binaires de longueurs différentes, l'argument le plus court est complété par des bits significatifs mis à 0.

x >> y décalage arithmétique vers la droite de y chiffres binaires de la représentation entière du complément à deux de x. Cette fonction n'est définie que pour des valeurs entières positives de y. Les bits devenus les bits de plus fort poids après décalage à droite ont une valeur égale aux bits de plus fort poids de x avant l'application du décalage.

x << y décalage arithmétique vers la gauche de y chiffres binaires de la représentation entière du complément à deux de x. Cette fonction n'est définie que pour des valeurs entières positives de y. Les bits devenus les bits de plus faible poids après décalage à gauche doivent avoir une valeur égale à 0.

5.5 Opérateurs d'affectation

Les opérateurs arithmétiques suivants sont définis comme suit:

= opérateur d'affectation.

++ Incrémentation, x++ équivalant à x = x + 1, utilisé dans un indice matriciel, il évalue la valeur de la variable avant l'opération d'incrémentation.

– – Décrémentation, x– – équivalant à x = x – 1; utilisé dans un indice matriciel, il évalue la valeur de la variable avant l'opération de décrémentation.

+= Incrémentation d'une quantité spécifiée. Ainsi, x += 3 est équivalent à x = x + 3, et x += (–3) est équivalent à x = x + (–3).

– = Décrémentation d'une quantité spécifiée. Ainsi, x – = 3 est équivalent à x = x – 3, et x – = (–3) est équivalent à x = x – (–3).

5.6 Variables, éléments syntaxiques et tableaux

Les éléments syntaxiques du flux binaire sont représentés en caractères **gras**. Chaque élément syntaxique est décrit par son nom (toujours en lettres minuscules avec des caractères de soulignement), par une ou deux catégories syntaxiques et par un ou deux descripteurs pour sa méthode de représentation codée. Le comportement du processus de décodage dépend de la valeur de l'élément syntaxique et des valeurs des éléments syntaxiques précédemment décodés. Lorsqu'elle est utilisée dans les tableaux syntaxiques ou dans le texte, la valeur d'un élément syntaxique apparaît en caractères normaux (c'est-à-dire pas en gras).

Dans certains cas, les tableaux syntaxiques peuvent utiliser les valeurs d'autres variables déduites de valeurs d'éléments syntaxiques. De telles variables apparaissent dans les tableaux syntaxiques, ou dans le texte, et sont désignées par l'association de majuscules et de minuscules sans caractère de soulignement. Les variables commençant par une majuscule sont utilisées pour le décodage de la structure syntaxique courante et de toutes les structures syntaxiques qui en dépendent. Les variables commençant par une majuscule peuvent être utilisées dans le processus de décodage de structures syntaxiques ultérieures en mentionnant la structure syntaxique d'origine de la variable. Les variables commençant par une minuscule ne sont utilisées qu'à l'intérieur du paragraphe dont elles sont tirées.

Dans certains cas, on utilise de façon interchangeable avec leurs valeurs numériques des noms "mnémoniques" pour des valeurs d'élément syntaxique ou de variable. Parfois, des noms "mnémoniques" sont utilisés sans valeur numérique associée. L'association entre valeurs et noms est spécifiée dans le texte. Les noms sont construits à partir d'un ou de plusieurs groupes de lettres séparées par un caractère de soulignement. Chaque groupe commence par une majuscule et peut contenir plusieurs majuscules.

NOTE – La syntaxe décrite s'apparente aux structures syntaxiques du langage C.

Une fonction est décrite par son nom, construit comme un nom d'élément syntaxique suivi de parenthèses gauche et droite incluant zéro, un ou plusieurs noms de variables (à des fins de définition) ou valeurs (à des fins d'utilisation) séparés par des virgules (s'il y a plusieurs variables).

La notation binaire est signalée par l'inclusion de la chaîne de valeurs binaires entre des apostrophes simples. Par exemple, '01000001' représente une chaîne de huit bits dont seuls les second et dernier bits sont égaux à 1.

La notation hexadécimale, signalée par la présence du préfixe "0x" devant le nombre hexadécimal, peut être utilisée à la place de la notation binaire lorsque le nombre de bits est un multiple entier de 4. Par exemple, 0x41 représente une chaîne binaire de huit bits dont seuls les second et dernier bits sont égaux à 1.

Les valeurs numériques non incluses entre apostrophes simples et non précédées du préfixe "0x" sont des valeurs décimales.

Une valeur égale à 0 représente une condition "FAUX" dans une instruction de test. La valeur "VRAI" est représentée par toute autre valeur que zéro.

5.7 Description textuelle des opérations logiques

Dans le texte, une instruction formée d'opérations logiques et qui serait décrite comme suit en pseudo-code:

```
si( condition 0 )
    instruction 0
    sinon si( condition 1 )
      instruction 1
    …
    sinon /* remarque informative sur la condition restante */
      instruction n
```

peut être décrite comme suit:

... comme suit / ... on applique ce qui suit:

– Si condition 0, instruction 0

– Sinon, si condition 1, instruction 1

– …

– Sinon (remarque informative sur la condition restante), instruction n.

Chaque instruction "si...sinon, si...sinon, ..." dans le texte est introduite par "... comme suit" ou "... on applique ce qui suit" immédiatement suivi de "si ... ". La dernière condition du "si...sinon, si...sinon, ..." est toujours un "sinon, ...". L'entrelacement des instructions "si...sinon, si...sinon, ..." peut se reconnaître en recherchant les "... comme suit" ou "... on applique ce qui suit" se terminant par "sinon, ...".

Dans le texte, une instruction formée d'opérations logiques et qui serait décrite comme suit en pseudo-code:

```
si( condition 0a  &&  condition 0b )
      instruction 0
    sinon si( condition 1a  ||  condition 1b )
      instruction 1
    …
    sinon
      instruction n
```

peut être décrite de la façon suivante:

... comme suit / ... on applique ce qui suit.

– Si toutes les conditions suivantes sont vraies, instruction 0
  - condition 0a
  - condition 0b

– Sinon, si une des conditions suivantes est vraie, instruction 1
  - condition 1a
  - condition 1b

– …

– Sinon, instruction n

Dans le texte, une instruction formée d'opérations logiques et qui serait décrite comme suit en pseudo-code:

```
si( condition 0 )
    instruction 0
si( condition 1 )
    instruction 1
```

peut être décrite de la façon suivante:

Lorsque condition 0, instruction 0

Lorsque condition 1, instruction 1

5.8 Méthode de description de la syntaxe sous forme de tableaux

Le tableau suivant donne une liste d'exemples de pseudo-codes utilisés pour décrire la syntaxe. Lorsqu'il apparaît, le code **syntax_element** spécifie qu'un élément syntaxique est analysé syntaxiquement à partir du flux binaire et que le pointeur de flux binaire est avancé jusqu'à la position suivante au-delà de l'élément syntaxique dans le processus d'analyse syntaxique du flux binaire.

	Descripteur
/* Une instruction peut être un élément syntaxique avec une catégorie syntaxique et un descripteur associés ou peut être une expression utilisée pour spécifier les conditions relatives à l'existence, au type et à la quantité des éléments syntaxiques, comme dans les deux exemples suivants */	
syntax_element	ue(v)
instruction ou structure posant une condition {	
/* Un groupe d'instructions entre accolades est une instruction composée traitée d'un point de vue fonctionnel comme une instruction unique */	
instruction	
instruction	
…	
}	
/* Une structure "tant que" est une instruction posant une condition qui spécifie un test visant à déterminer si une condition est Vraie. Si c'est le cas, elle spécifie l'évaluation répétitive d'une instruction (ou d'une instruction composée) jusqu'à ce que la condition ne soit plus Vraie */	
tant que(condition)	
instruction	
/* Une structure "faire … tant que" est un structure posant une condition qui spécifie l'évaluation d'une instruction une seule fois, suivie par un test visant à déterminer si une condition est Vraie. Si c'est le cas, elle spécifie une évaluation répétée de l'instruction jusqu'à ce que la condition ne soit plus Vraie */	
faire	
instruction	
tant que(condition)	

	Descripteur
/* Une structure "si … sinon" spécifie un test visant à déterminer si une condition est Vraie. Si c'est le cas, elle spécifie l'évaluation d'une instruction primaire; sinon, elle spécifie l'évaluation d'une instruction alternative. La partie "sinon" de la structure et l'instruction alternative associée est omise si l'évaluation d'une instruction alternative n'est pas nécessaire */	
si(condition)	
instruction primaire	
sinon	
instruction alternative	
/* Une structure "pour" spécifie l'évaluation d'une instruction initiale suivie par le test d'une condition. Si la condition est Vraie, elle spécifie l'évaluation répétée d'une instruction principale suivie d'une instruction ultérieure jusqu'à ce que la condition ne soit plus Vraie */	
pour(instruction initiale; condition; instruction ultérieure)	
instruction principale	

5.9 Spécification de fonctions, catégories et descripteurs syntaxiques

Les fonctions présentées ici sont utilisées dans la description syntaxique. Ces fonctions supposent l'existence d'un pointeur de flux binaire avec indication de la position du prochain bit à lire par le processus de décodage du flux binaire.

byte_aligned( ) est spécifié comme suit:

– si la position courante dans le flux binaire se situe à la frontière d'un octet, c'est-à-dire si le bit suivant dans le flux binaire est le premier bit d'un octet, la valeur renvoyée par byte_aligned( ) est égale à "VRAI";

– sinon, la valeur renvoyée par byte_aligned( ) est égale à "FAUX".

more_msg_data( ) est spécifié comme suit:

– s'il y a encore des données dans la structure syntaxique msg_data( ), la valeur renvoyée par more_msg_data( ) est égale à "VRAI";

– sinon, la valeur renvoyée par more_msg_data( ) est égale à "FAUX".

La méthode permettant de déterminer s'il y a encore des données dans la structure syntaxique msg_data( ) est spécifiée par l'application.

next_bits(n) fournit les bits suivants du flux binaire à des fins de comparaison, sans faire avancer le pointeur de flux binaire. Cette fonction permet de connaître la valeur des n bits suivants du flux binaire, avec n comme argument.

read_bits(n) lit les n bits suivants du flux binaire et avance le pointeur de flux binaire de n positions de bit. Lorsque n est égal à 0, read_bits(n) est spécifié de manière à renvoyer une valeur égale à 0 et à ne pas avancer le pointeur de flux binaire.

Les descripteurs suivants spécifient le processus d'analyse syntaxique de chaque élément syntaxique.

– f(n): chaîne binaire à séquence fixe utilisant n bits écrits (de gauche à droite), le premier étant le bit de gauche. Le processus d'analyse syntaxique pour ce descripteur est spécifié par la valeur renvoyée par la fonction read_bits(n).

– u(n): entier non signé utilisant n bits. Lorsque n est "v" dans le tableau syntaxique, le nombre de bits varie en fonction de la valeur des autres éléments syntaxiques. Le processus d'analyse syntaxique pour ce descripteur est spécifié par la valeur renvoyée par la fonction read_bits(n), interprétée comme une représentation binaire d'un entier non signé avec le bit de plus fort poids écrit en premier.

– ue(v): élément syntaxique entier non signé codé en Exp-Golomb avec le bit de gauche en premier. Le processus d'analyse syntaxique pour ce descripteur est spécifié ci-après.

Les éléments syntaxiques codés ue(v) sont codés en Exp-Golomb. Le processus d'analyse syntaxique qui leur est applicable commence par la lecture des bits à partir de l'emplacement courant dans le flux binaire pour se poursuivre jusqu'au premier bit non nul inclusivement, en comptant le nombre de bits initiaux égaux à 0. Ce processus doit être équivalent à ce qui suit:

```
leadingZeroBits = −1;
pour( b = 0; !b; leadingZeroBits++ )
    b = read_bits( 1 )
```

La valeur de la variable codeNum est alors calculée comme suit:

$$codeNum = 2^{leadingZeroBits} - 1 + read\_bits(leadingZeroBits)$$

où la valeur renvoyée par read_bits(leadingZeroBits) est interprétée comme une représentation binaire d'un entier non signé dont le bit de plus fort poids est écrit en premier.

La valeur de l'élément syntaxique est égale à codeNum.

6 Charges utiles de message

6.1 Syntaxe

msg_data( ) {	**Descripteur**
faire	
message( )	
tant que(more_msg_data( ))	
}	

message( ) {	**Descripteur**
payloadType = 0	
tant que(next_bits(8) == 0xFF) {	
ff_byte /* égal à 0xFF */	f(8)
payloadType += 255	
}	
last_payload_type_byte	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
tant que(next_bits(8) == 0xFF) {	
ff_byte /* égal à 0xFF */	f(8)
payloadSize += 255	
}	
last_payload_size_byte	u(8)
payloadSize += last_payload_size_byte	
msg_payload(payloadType, payloadSize)	
}	

msg_payload(payloadType, payloadSize) {	**Descripteur**
si(payloadType <= 4)	
ref_pic_id	u(32)
si(payloadType == 0) {	
num_ref_pics_minus1	ue(v)
pour(i = 1; i <= num_ref_pics_minus1; i++)	
good_ref_pic_id[i]	u(32)
} sinon si(payloadType == 1)	
delta_ref_pic_id	ue(v)
sinon si(payloadType == 2) {	
data_partition_idc	ue(v)
run_length_flag	u(1)
si(run_length_flag) {	
first_blk_lost	ue(v)
num_blks_lost_minus1	ue(v)
} sinon {	
top_left_blk	ue(v)
bottom_right_blk	ue(v)
}	
} sinon si((payloadType == 3) || (payloadType == 4)) {	
param_set_type	ue(v)
param_set_crc	u(16)
si(payloadType == 3)	
param_set_id	ue(v)

}	
stop_one_bit /* égal à 1 */	f(1)
tant que(!byte_aligned( ))	
alignment_zero_bit /* égal à 0 */	f(1)
}	

6.2 Sémantique

ff_byte doit être égal à 0xFF.

last_payload_type_byte est le dernier octet utilisé pour spécifier le type de charge utile d'un message.

last_payload_size_byte est le dernier octet utilisé pour spécifier la taille d'un message.

La taille de la structure syntaxe msg_payload( ) doit être égale à payloadSize * 8 bits.

Les types de message correspondant aux diverses valeurs de payloadType sont indiqués dans le Tableau 6-1.

Tableau 6-1/H.271 – Types de message

payloadType	Message
0	Une ou plusieurs images sans erreur d'adaptation de flux binaire détectée
1	Une ou plusieurs images entièrement ou partiellement perdues
2	Ensemble de blocs d'une image entièrement ou partiellement perdue
3	Code CRC associé à un ensemble de paramètres
4	Code CRC associé à tous les ensembles de paramètres d'un certain type
5	Demande de "réinitialisation" indiquant que l'émetteur devrait procéder à un rafraîchissement complet du flux binaire vidéo comme si aucune donnée de flux binaire préalable n'avait été reçue
> 5	Réservé pour une utilisation future par l'UIT-T

Les messages dont le champ payloadType a une valeur supérieure à 5 doivent être supprimés et rejetés suivant la taille du message indiquée par payloadSize.

ref_pic_id identifie une image. La sémantique qui lui est associée, qui dépend de la valeur de payloadType, est spécifiée dans le Tableau 6-2.

Tableau 6-2/H.271 – Sémantique associée à ref_pic_id

payloadType	Sémantique associée à ref_pic_id
0	Image sans erreur d'adaptation de flux binaire détectée
1	Image entièrement ou partiellement perdue
2	Image partiellement perdue
3	Spécification d'une image. Un code CRC existe pour un ensemble de paramètres associé à l'image spécifiée
4	Spécification d'une image. Un code CRC existe pour tous les ensembles de paramètres d'un certain type stockés à l'instant associé au décodage de l'image spécifiée

num_ref_pics_minus1 plus 1 indique le nombre d'images sans erreur d'adaptation de flux binaire détectée. num_ref_pics_minus1 doit avoir une valeur comprise entre 0 et 31 inclus.

good_ref_pic_id[i] désigne la ième image indiquée dans le message courant qui est sans erreur d'adaptation de flux binaire détectée.

delta_ref_pic_id spécifie une incrémentation par rapport à ref_pic_id identifiant un ensemble d'images totalement ou partiellement perdues. Sa valeur doit être comprise entre 0 et 31 inclus.

data_partition_idc indique la perte de toutes les données (lorsque data_partition_idc est égal à 0) ou d'une partition des données (lorsque data_partition_idc n'est pas égal à 0) pour les blocs spécifiés par top_left_blk et bottom_right_blk . Sa valeur doit être comprise entre 0 et 15 inclus.

run_length_flag indique, lorsqu'il est égal à 1, la présence des éléments syntaxiques first_blk_lost et num_blks_lost_minus1. Lorsqu'il est égal à 0, il indique la présence des éléments syntaxiques top_left_blk et bottom_right_blk.

first_blk_lost indique l'adresse du premier bloc suivant l'ordre de balayage de trames. Une adresse de bloc est l'indice d'un bloc suivant le balayage par blocs de l'image, l'indice 0 étant attribué au bloc supérieur gauche de l'image.

num_blks_lost_minus1 plus 1 désigne le nombre de blocs consécutifs suivant l'ordre de balayage de trames, le premier bloc étant identifié par first_blk_lost.

top_left_blk et **bottom_right_blk** désignent respectivement les adresses des blocs supérieur gauche et inférieur droit de la région rectangulaire entièrement ou partiellement perdue. Les contraintes suivantes doivent être observées par les valeurs des éléments syntaxiques top_left_blk et bottom_right_blk, la variable PicWidthInBlks désignant la largeur de l'image exprimée en blocs.

– top_left_blk doit être inférieur ou égal à bottom_right_blk.

– bottom_right_blk doit être inférieur à la taille de l'image exprimée en blocs.

– top_left_blk % PicWidthInBlks doit être inférieur ou égal à bottom_right_blk % PicWidthInBlks.

param_set_type indique le type du ou des ensembles de paramètres. Sa valeur doit être comprise entre 0 et 15 inclus.

param_set_crc désigne le code CRC de l'ensemble de paramètres identifié par param_set_id et param_set_type lorsque payloadType est égal à 3, ou lorsque payloadType est égal à 4 le code CRC de tous les ensembles de paramètres du type spécifié par param_set_type.

La valeur de param_set_crc doit être égale à la valeur de crcVal obtenue en appliquant le pseudo-code indiqué ci-après.

```
paramSet[pLen     ] = 0
paramSet[pLen + 1] = 0
crcVal = 0xFFFF
pour( bitIdx = 0; bitIdx < ( pLen + 2 ) * 8; bitIdx++ ) {
   crcMsb = ( crcVal >> 15 ) & 1
   bitVal = ( paramSet[bitIdx >> 3] >> ( 7 – ( bitIdx & 7 ) ) ) & 1
   crcVal = ( ( ( crcVal << 1 ) + bitVal ) & 0xFFFF ) ^ ( crcMsb * 0x1021 )
}                                                                    (6-1)
```

où la variable paramSet est une chaîne d'octets contenant, en début de chaîne, les données pour lesquelles le code CRC est calculé; la variable pLen est le nombre d'octets des données pour lesquelles le code CRC est calculé; et la chaîne d'octets de la variable paramSet est suffisamment longue pour pouvoir ajouter deux octets nuls supplémentaires à la fin des données pour lesquelles le code CRC est calculé.

Lorsque payloadType est égal à 3, paramSet contient, suivant l'ordre d'octets du réseau, les octets de l'ensemble de paramètres reçu identifié par param_set_id et param_set_type.

Lorsque payloadType est égal à 4, paramSet contient une concaténation, suivant l'ordre d'octets du réseau, des octets de tous les ensembles de paramètres du type identifié par param_set_type, suivant l'ordre croissant des identificateurs d'ensemble de paramètres. Pour tout ensemble de paramètres n'ayant jamais été reçu, on doit considérer que sa représentation a une longueur de deux octets, le premier contenant les huit bits de plus fort poids et le second les huit bits de plus faible poids d'une représentation binaire à 16 bits non signée de la valeur de l'identificateur de l'ensemble de paramètres.

param_set_id désigne l'identificateur de l'ensemble de paramètres du type spécifié par param_set_type pour lequel un code CRC est présent. Sa valeur doit être comprise entre 0 et 65535 inclus.

stop_one_bit doit être égal à 1.

alignment_zero_bit doit être égal à 0.

7 Utilisations spécifiques normalisées des messages

Dans le présent paragraphe, le bit n d'une variable désigne le nième bit le moins significatif de sa représentation binaire. Le bit de plus faible poids est le bit 0.

7.1 Utilisation spécifique H.261 des messages

La valeur de payloadType doit être égale à 0, 1, 2 ou 5. Les messages ayant d'autres valeurs de payloadType doivent être ignorés.

Si ref_pic_id ou good_ref_pic_id[i] est présent, les cinq bits de plus faible poids de cet élément syntaxique indiquent la valeur de la référence temporelle (égale à ref_pic_id & 0x1F ou good_ref_pic_id[i] & 0x1F) d'une image. Les bits restants de ref_pic_id ou good_ref_pic_id[i] sont réservés pour une utilisation future par l'UIT-T; ils doivent être égaux à 0 et être ignorés.

Les considérations décrites ci-après s'appliquent, suivant la valeur de payloadType.

– Si payloadType est égal à 0, l'image de référence temporelle égale à ref_pic_id & 0x1F ou good_ref_pic_id[i] & 0x1F ne contient pas d'erreur d'adaptation de flux binaire détectée.
– Sinon, si payloadType est égal à 1, toutes les images, considérées suivant l'ordre de décodage à partir d'une image de référence temporelle égale à ref_pic_id & 0x1F et jusqu'à (inclusivement) une image de référence temporelle égale à ((ref_pic_id & 0x1F) + delta_ref_pic_id) % 32, sont, lorsque de telles images sont présentes dans le flux binaire vidéo, signalées comme ayant été entièrement ou partiellement perdues.
– Sinon, si payloadType est égal à 2, l'image de référence temporelle égale à ref_pic_id & 0x1F a été partiellement perdue. Chaque bloc est un macrobloc. La valeur de data_partition_idc doit être égale à 0, ce qui indique que toutes les données des macroblocs identifiées sont considérées comme perdues. Toutes les autres valeurs de data_partition_idc sont réservées pour une utilisation future par l'UIT-T et doivent être ignorées.
– Sinon (payloadType est égal à 5), il n'y a pas d'autre spécification.

7.2 Utilisation spécifique H.263 des messages

Les caractéristiques H.263 prises en charge par la présente syntaxe sont les suivantes:

– images durables;
– partitionnement des données.

Les caractéristiques H.263 suivantes ne sont pas prises en compte par la présente syntaxe:

– trames entrelacées.

La valeur de payloadType doit être égale à 0, 1, 2 ou 5. Les messages ayant d'autres valeurs de payloadType doivent être ignorés.

Lorsque ref_pic_id ou good_ref_pic_id[i] est présent, les douze bits de plus faible poids donnent la valeur de la variable picIdentifier, égale à ref_pic_id & 0x0FFF ou good_ref_pic_id[i] & 0x0FFF. Lorsque le mode de l'Annexe U/H.263 est utilisé et que payloadType est égal à 0, le bit 12 indique que l'image enregistrée est une image durable (lorsque le bit est égal à 1) ou provisoire (lorsque le bit est égal à 0). Le bit 13 doit être égal à 0 sauf en cas d'utilisation du mode facultatif d'échelonnabilité temporelle, SNR et spatiale (Annexe O/H.263). Lorsque le bit 13 est égal à 1, le message se rapporte à une couche d'amélioration (plutôt qu'à la couche de base) et les bits 14 à 17 (inclusivement) donnent la valeur de ELNUM, égale à ref_pic_id & 0x03C000 ou good_ref_pic_id[i] & 0x03C000. Les bits restants de ref_pic_id ou good_ref_pic_id[i] sont réservés pour une utilisation future par l'UIT-T, ils doivent être égaux à 0 et être ignorés.

– Si le mode de l'Annexe U/H.263 n'est pas utilisé, picIdentifier indique la valeur de référence temporelle d'une image. Cette valeur doit être inférieure à MaxTR, qui correspond à la valeur possible maximale de référence temporelle plus 1. Les considérations décrites ci-après s'appliquent, suivant la valeur de payloadType.

  • Si payloadType est égal à 0, le bit 12 de ref_pic_id ou good_ref_pic_id[i] doit être égal à 0. L'image de référence temporelle égale à picIdentifier et ayant, lorsque le bit 13 de ref_pic_id ou good_ref_pic_id[i] est égal à 1, l'indication ELNUM égale à ref_pic_id & 0x03C000 ou good_ref_pic_id[i] & 0x03C000, ne contient pas d'erreur d'adaptation de flux binaire détectée.

  • Sinon, si payloadType est égal à 1, le bit 12 de ref_pic_id doit être égal à 0. Toutes les images, considérées suivant l'ordre de décodage dans la couche de base ou la même couche avec ELNUM égal à ref_pic_id & 0x03C000, à partir d'une image de référence temporelle égale à ref_pic_id & 0x0FFF et jusqu'à (inclusivement) une image de référence temporelle égale à ((ref_pic_id & 0x0FFF) + delta_ref_pic_id) % MaxTR, sont, lorsque de telles images sont présentes dans le flux binaire vidéo, signalées comme ayant été entièrement ou partiellement perdues.

  • Sinon, si payloadType est égal à 2, le bit 12 de ref_pic_id doit être égal à 0. L'image de référence temporelle égale à picIdentifier et ayant, lorsque le bit 13 de ref_pic_id est égal à 1, l'indication ELNUM égale à ref_pic_id & 0x03C000 est partiellement perdue. La valeur de data_partition_idc indique la perte de toutes les données (lorsque data_partition_idc est égal à 0), de la partition des données d'en-tête (lorsque data_partition_idc est égal à 1), de la partition du vecteur cinétique (lorsque data_partition_idc est égal à 2) ou de la partition des données de coefficient (lorsque data_partition_idc est égal à 3) des macroblocs identifiés. Toutes les autres valeurs de data_partition_idc sont réservées pour une utilisation future par l'UIT-T et doivent être ignorées.

  • Sinon (payloadType est égal à 5), il n'y a pas d'autre spécification.

– Sinon (le mode de l'Annexe U/H.263 est utilisé), les considérations décrites ci-après s'appliquent, suivant la valeur du type de charge utile.

  • Si payloadType est égal à 0, ref_pic_id ou good_ref_pic_id[i] identifie une image enregistrée sans erreur d'adaptation de flux binaire détectée. La variable picIdentifier indique la valeur du champ PN (lorsque l'image enregistrée est une image provisoire) ou LPIN (lorsque l'image enregistrée est une image durable) de l'image enregistrée.

    – S'il s'agit d'une image provisoire, la valeur de picIdentifier doit être inférieure à MaxPN, valeur possible maximale de PN plus 1.

    - Sinon (il s'agit d'une image durable), la valeur de picIdentifier doit être inférieure à MaxLPIN, valeur possible maximale de LPIN plus 1.

– Sinon, si payloadType est égal à 1, le bit 12 de ref_pic_id doit être égal à 0 et picIdentifier indique la valeur du champ PN d'une image. La valeur de PN doit être inférieure à MaxPN, valeur possible maximale de PN plus 1. Toutes les images, considérées suivant l'ordre de décodage dans la même couche d'amélioration ou dans la couche de base à partir d'une image ayant un champ PN égal à ref_pic_id & 0x0FFF et jusqu'à (inclusivement) une image ayant un champ PN égal à ((ref_pic_id & 0x0FFF) + delta_ref_pic_id) % MaxPN sont, lorsque de telles images sont présentes dans le flux binaire vidéo, signalées comme ayant été entièrement ou partiellement perdues. MaxPN est égal à la valeur possible maximale de PN plus 1.

– Sinon, si payloadType est égal à 2, le bit 12 de ref_pic_id doit être égal à 0 et picIdentifier indique la valeur PN d'une image. La valeur de PN doit être inférieure à MaxPN, valeur possible maximale de PN plus 1. L'image de référence temporelle égale à picIdentifier et ayant, lorsque le bit 13 de ref_pic_id est égal à 1, l'indication ELNUM égale à ref_pic_id & 0x03C000 est partiellement perdue. Chaque bloc est un macrobloc. La valeur de data_partition_idc indique la perte de toutes les données (lorsque data_partition_idc est égal à 0), de la partition des données d'en-tête (lorsque data_partition_idc est égal à 1), de la partition du vecteur cinétique (lorsque data_partition_idc est égal à 2) ou de la partition des données de coefficient (lorsque data_partition_idc est égal à 3) des macroblocs identifiés. Toutes les autres valeurs de data_partition_idc sont réservées pour une utilisation future par l'UIT-T et doivent être ignorées.

– Sinon (payloadType est égal à 5), il n'y a pas d'autre spécification.

7.3 Utilisation spécifique H.264 des messages

Les caractéristiques H.264 prises en charge par la présente syntaxe sont les suivantes:

– images de référence durables;

– partitionnement des données;

– ensemble de paramètres de séquence et d'image.

Les caractéristiques H.264 suivantes ne sont pas prises en charge par la présente syntaxe:

– images codées par codage d'image/de trame de monobloc adaptatif;

– images monotrames.

Lorsque ref_pic_id ou good_ref_pic_id[i] est présent, les considérations décrites ci-après s'appliquent. Les seize bits de plus faible poids indiquent la valeur de la variable picIdentifier, égale à ref_pic_id & 0xFFFF ou good_ref_pic_id[i] & 0xFFFF. Lorsque payloadType est égal à 0, le bit 16 indique que l'image de référence est une image de référence durable (lorsque le bit est égal à 1) ou une image de référence provisoire (lorsque le bit est égal à 0). Les bits restant de ref_pic_id ou good_ref_pic_id[i] sont réservés pour une utilisation future par l'UIT-T; ils doivent être égaux à 0 et être ignorés.

Les considérations décrites ci-après s'appliquent, suivant la valeur de payloadType.

– Si payloadType est égal à 0, ref_pic_id ou good_ref_pic_id[i] identifie une image de référence sans erreur d'adaptation de flux binaire détectée. La variable picIdentifier indique la valeur du champ FrameNum (lorsque l'image de référence est une image de référence provisoire) ou LongTermFrameIdx (lorsque l'image de référence est une image de référence durable) de l'image de référence.

    • Si l'image est une image de référence provisoire, la valeur de picIdentifier doit être inférieure à MaxFrameNum.

- Sinon (l'image est une image de référence durable), la valeur de picIdentifier ne doit pas être supérieure celle de MaxLongTermFrameIdx.

– Sinon, si payloadType est égal à 1, le bit 16 doit être égal à 0 et la variable picIdentifier indique la valeur FrameNum d'une image de référence. La valeur de FrameNum doit être inférieure à celle de MaxFrameNum. Toutes les images, considérées suivant l'ordre de décodage à partir d'une image ayant FrameNum égal à ref_pic_id & 0xFFFF jusqu'à (inclusivement) une image ayant FrameNum égal à ((ref_pic_id & 0xFFFF) + delta_ref_pic_id) % MaxFrameNum, sont, lorsque de telles images sont présentes dans le flux binaire vidéo, signalées comme ayant été entièrement ou partiellement perdues.

– Sinon, si payloadType est égal à 2, le bit 16 doit être égal à 0 et la variable picIdentifier indique la valeur FrameNum d'une image de référence partiellement perdue. La valeur de FrameNum doit être inférieure à celle de MaxFrameNum. Chaque bloc est un macrobloc. La valeur de data_partition_idc indique la perte de toutes les données (lorsque data_partition_idc est égal à 0), de la partition de données A (lorsque data_partition_idc est égal à 1), de la partition de données B (lorsque data_partition_idc est égal à 2) ou de la partition de données C (lorsque data_partition_idc est égal à 3) des macroblocs identifiés. Toutes les autres valeurs de data_partition_idc sont réservées pour une utilisation future par l'UIT-T et doivent être ignorées.

– Sinon, si payloadType est égal à 3 ou 4, les seize bits de plus faible poids de ref_pic_id donnent la valeur FrameNum, égale à ref_pic_id & 0xFFFF, d'une image de référence pour laquelle un code CRC est présent pour un ou tous les ensembles de paramètres de séquence ou d'image associés. La valeur de ref_pic_id & 0xFFFF doit être inférieure à celle de MaxFrameNum. Les bits restant de ref_pic_id sont réservés pour une utilisation future par l'UIT-T; ils doivent être égaux à 0 et être ignorés. Un ensemble de paramètres pour lequel param_set_type est égal à 0 est une unité couche d'abstraction du réseau (NAL, *network abstraction layer*) d'ensemble de paramètres de séquence. Un ensemble de paramètres pour lequel param_set_type est égal à 1 est une unité NAL d'ensemble de paramètres d'image. En cas d'utilisation pour le calcul d'un code CRC, les champs forbidden_zero_bit et nal_ref_idc d'une unité NAL d'ensemble de paramètres de séquence ou d'image sont respectivement mis à 0 et 3.

– Sinon (payloadType est égal à 5), il n'y a pas d'autre spécification.

SÉRIES DES RECOMMANDATIONS UIT-T

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

Imprimé en Suisse
Genève, 2006