

Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.808

(01/2005)

SÉRIE T: TERMINAUX DES SERVICES
TÉLÉMATIQUES

**Technologies de l'information – Système de
codage d'images JPEG 2000: outils
d'interactivité, interfaces de programmes
d'application et protocoles**

Recommandation UIT-T T.808



**Technologies de l'information – Système de codage d'images JPEG 2000: outils
d'interactivité, interfaces de programmes d'application et protocoles**

Résumé

L'objet de la présente Recommandation | Norme internationale consiste à offrir un protocole de réseau qui permet la transmission interactive et progressive de données ou de fichiers à codage JPEG 2000 entre un serveur distant et un client. Ce protocole permet à un client de demander seulement les portions d'une image (par région, par qualité ou par niveau de résolution) qui sont applicables à ses propres besoins. Il permet également au client d'accéder à des métadonnées ou à un autre contenu à partir du fichier.

Source

La Recommandation UIT-T T.808 a été approuvée le 8 janvier 2005 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Un texte identique est publié comme Norme Internationale ISO/CEI 15444-9.

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 l'implémentation 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 l'implémentation de consulter la base de données des brevets du TSB.

© UIT 2006

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

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives..... 1
3	Définitions..... 2
3.1	Définitions relatives à la Partie 1 de la norme JPEG 2000..... 2
3.2	Définitions relatives au protocole HTTP..... 2
3.3	Définitions relatives au protocole JPIP..... 2
3.4	Symboles..... 3
4	Abréviations..... 5
5	Conventions..... 6
5.1	Règles de formalisme ABNF..... 6
5.2	Règles de formalisme ABNF relatives au format de fichier..... 6
5.3	Légende des descriptions graphiques de boîtes (pour information)..... 7
6	Description générale..... 7
6.1	Protocole JPIP..... 7
6.2	Objet..... 8
7	Conformité..... 9
Annexe A (normative) – Types de média à flux JPP et JPT..... 10	
A.1	Introduction..... 10
A.2	Structure d'en-tête de message..... 11
A.3	Segments de données..... 13
A.4	Conventions relatives à l'analyse sémantique et à la remise de flux JPP et JPT (pour information)..... 22
A.5	Conventions relatives à l'interopérabilité avec des flux JPP ou JPT (pour information)..... 22
Annexe B (normative) – Sessions, canaux, modèle de cache et ensembles de modèles..... 23	
B.1	Requêtes émises au cours d'une session par opposition aux requêtes sans description d'état..... 23
B.2	Canaux et sessions..... 23
B.3	Gestion du modèle de cache..... 24
B.4	Interrogation et manipulation des ensembles de modèles..... 24
Annexe C (normative) – Requête émise par un client..... 26	
C.1	Syntaxe des requêtes..... 26
C.2	Champs d'identification de cible..... 27
C.3	Champs de travail avec des sessions et des canaux..... 29
C.4	Champs de requête de fenêtre de visualisation..... 30
C.5	Champs de requête de métadonnées..... 39
C.6	Champs de requête de limitation de données..... 42
C.7	Champs de requête de commande de serveur distant..... 43
C.8	Champs de requête de gestion de cache..... 45
C.9	Paramètres de requête de téléchargement amont..... 51
C.10	Champs de requête de capacité et de préférence du client..... 51
Annexe D (normative) – Signalisation de la réponse du serveur distant..... 58	
D.1	Syntaxe de réponse..... 58
D.2	En-têtes de réponse JPIP..... 59
D.3	Données de réponse..... 64
Annexe E (normative) – Téléchargement amont d'images vers le serveur distant..... 66	
E.1	Introduction..... 66
E.2	Requête de téléchargement amont..... 66
E.3	Réponse du serveur distant..... 67
E.4	Fusionnement de données dans le serveur distant..... 67
Annexe F (normative) – Utilisation du protocole JPIP au-dessus du protocole HTTP..... 69	
F.1	Introduction..... 69
F.2	Requêtes..... 69
F.3	Etablissement de session..... 70

	<i>Page</i>
F.4 Réponses	71
F.5 Caractéristiques additionnelles du protocole HTTP	72
F.6 Protocole HTTP et champ de requête de longueur (pour information)	72
Annexe G (normative) – Utilisation du protocole JPIP avec requêtes HTTP et retours TCP	73
G.1 Introduction	73
G.2 Requêtes émises par un client	73
G.3 Etablissement de session	73
G.4 Réponse des serveurs distants	74
G.5 Protocole TCP et champ de requête de longueur (pour information)	74
Annexe H (informative) – Utilisation du protocole JPIP avec des transports de remplacement	75
H.1 Introduction	75
H.2 Requêtes fiables avec données non fiables	75
H.3 Requêtes non fiables avec données non fiables	76
H.4 Syntaxe de requête et de réponse	77
H.5 Etablissement de session	77
Annexe I (normative) – Indexation de fichiers JPEG 2000 pour le protocole JPIP	78
I.1 Introduction (pour information)	78
I.2 Identification de l'utilisation de boîtes d'indexation JPIP dans la liste de compatibilités des formats de fichier JPEG 2000	79
I.3 Boîtes d'indexation définies	79
I.4 Association des indices de flux codé avec des flux codés	88
I.5 Restrictions d'emplacement (pour information)	88
Annexe J (normative) – Enregistrement des extensions à la présente Recommandation Norme internationale	89
J.1 Introduction relative à l'enregistrement	89
J.2 Eléments d'enregistrement	89
J.3 Critères d'évaluation d'enregistrement	89
J.4 Eléments qui peuvent être étendus par enregistrement	89
J.5 Processus d'enregistrement	90
J.6 Créneaux temporels pour le processus d'enregistrement	90
Annexe K (informative) – Exemples d'application	91
K.1 Introduction	91
K.2 Utilisation du protocole JPIP avec des flux codés dans d'autres formats de fichier	91
K.3 Techniques d'implémentation par éléments de pavé	91
K.4 Techniques d'implémentation par districts	92
K.5 Transcriptions en protocole JPIP	93
K.6 Utilisation du protocole JPIP avec le langage HTML	97
Annexe L (informative) – Production du protocole JPIP en formalisme ABNF	98
L.1 Formalisme ABNF d'une requête JPIP	98
L.2 Formalisme BNF d'une réponse JPIP	104
Annexe M (informative) – Revendication de droits de propriété intellectuelle	108
Annexe N (informative) – Bibliographie	109

FIGURES

	<i>Page</i>
Figure 1 – Exemple des figures descriptives de boîte.....	7
Figure 2 – Exemple des figures descriptives de superboîte.....	7
Figure 3 – Vue d'ensemble du protocole JPIP.....	8
Figure 4 – Empilement du protocole JPIP.....	8
Figure A.1 – Exemples de relations entre fichier JPEG 2000, segments de données JPIP et flux JPIP (d'après G.J. Colyer et R.A. Clark, IEEE Trans. Consumer Electronics, 49 (2003), p. 850–854).....	10
Figure A.2 – Structure des segments VBAS.....	11
Figure A.3 – Structure des segments VBAS à identificateur de segment.....	11
Figure A.4 – Exemple de segment de données de district.....	14
Figure A.5 – Exemple de système colorimétrique de segment de métadonnées.....	16
Figure A.6 – Echantillon de fichier JP2.....	16
Figure A.7 – Echantillon de fichier JP2 subdivisé en trois segments de métadonnées.....	17
Figure A.8 – Superboîte avec segment de métadonnées référencé.....	18
Figure A.9 – Subdivision illégale du fichier en segments de métadonnées.....	18
Figure A.10 – Exemple de l'utilisation de flux équivalents.....	19
Figure A.11 – Structure de boîte générique.....	19
Figure C.1 – Recherche de région dans une image.....	31
Figure C.2 – Recherche de région par rapport à la grille de référence sous-échantillonnée.....	32
Figure C.3 – Spécification de l'espace chromatique: procédure de sélection de boîte.....	55
Figure G.1 – Structure des données de réponse au-dessus de la connexion http-tcp.....	74
Figure I.1 – Exemple partiel de fichier JPEG 2000 contenant des boîtes d'indexation JPIP.....	79
Figure I.2 – Organisation du contenu d'une boîte d'indexation de flux codé.....	80
Figure I.3 – Organisation du contenu d'une boîte de localisateur de flux codé.....	80
Figure I.4 – Organisation du contenu d'une boîte de résumé.....	81
Figure I.5 – Organisation du contenu d'une boîte d'indexation d'une série tabulaire de fragments.....	82
Figure I.6 – Organisation du contenu d'une boîte de table d'indices d'en-tête.....	83
Figure I.7 – Organisation du contenu d'une boîte de table d'indices d'élément de pavé.....	84
Figure I.8 – Organisation du contenu d'une boîte de table d'indices d'en-tête de pavé.....	84
Figure I.9 – Organisation du contenu d'une boîte de table d'indices de paquet de district.....	85
Figure I.10 – Organisation du contenu d'une boîte d'indices d'en-tête de paquet.....	85
Figure I.11 – Organisation du contenu d'une boîte d'indexation de fichier.....	86
Figure I.12 – Organisation du contenu d'une boîte de localisateur de fichier.....	87
Figure I.13 – Organisation du contenu d'une boîte intermédiaire.....	87
Figure I.14 – Organisation du contenu d'une boîte de localisateur d'indice.....	88

TABLEAUX

	<i>Page</i>
Tableau A.1 – Identificateur de segment: indication supplémentaire de segment VBAS	12
Tableau A.2 – Identificateurs de classe pour différentes classes de message de segment de données	13
Tableau A.3 – Valeurs légales pour le champ de fanions "Flags" d'une boîte générique	21
Tableau C.1 – Options relatives au paramètre "round-direction"	34
Tableau C.2 – Fanions qualificateurs d'une requête de métadonnées	42
Tableau C.3 – Frontières de verrouillage fondées sur le type de segment.....	43
Tableau C.4 – Types légaux de retour d'image.....	44
Tableau C.5 – Résumé des options de descripteur de cache.....	48
Tableau C.6 – Capacités légales du paramètre "processing-capabilities"	52
Tableau C.7 – Valeurs légales du paramètre "config-capability"	52
Tableau C.8 – Préférences relatives à la manipulation de la fenêtre de visualisation.....	53
Tableau C.9 – Préférences du client concernant la méthode de détermination de l'espace chromatique.....	54
Tableau C.10 – Préférences relatives aux éléments génériques.....	56
Tableau C.11 – Préférences relatives au séquençement des flux codés.....	57
Tableau D.1 – Valeurs légales du paramètre de transport	60
Tableau D.2 – Codes de cause définis	64
Tableau I.1 – Boîtes d'indexation définies (pour information)	79
Tableau I.2 – Valeurs de type de conteneur.....	81
Tableau I.3 – Valeurs de version	83
Tableau K.1 – Exemple de l'utilisation de champs auxiliaires dans un cas simple.....	92
Tableau K.2 – Exemple de l'utilisation de champs auxiliaires dans un cas plus compliqué.....	92

Introduction

La Rec. UIT-T T.800 | ISO/CEI 15444-1 (JPEG 2000) est une spécification qui décrit un système de compression d'image offrant une grande flexibilité, non seulement pour la compression d'images mais également pour l'accès au flux codé, lequel offre un certain nombre de mécanismes permettant de localiser et d'extraire des portions de données d'image comprimée aux fins de retransmission, stockage, affichage ou édition. Cet accès permet le stockage et l'extraction de données d'image comprimée appropriées à une application donnée, sans décodage.

L'objet de la présente Recommandation | Norme internationale consiste à offrir un protocole de réseau permettant la transmission interactive et progressive de données et de fichiers à codage JPEG 2000, d'un serveur distant à un client. Ce protocole permet à un client de demander les seules portions d'une image (par région, qualité ou niveau de résolution) qui sont applicables à ses propres besoins. Il permet également au client d'accéder à des métadonnées ou à un autre contenu à partir du fichier.

Toute organisation envisageant l'utilisation de la présente Recommandation | Norme internationale devrait examiner en détail son applicabilité.

L'Union internationale des télécommunications (UIT), l'Organisation internationale de normalisation (ISO) et la Commission Electrotechnique Internationale (CEI) appellent l'attention sur le fait qu'il est revendiqué que la conformité à la présente Recommandation | Norme internationale peut impliquer l'utilisation d'un brevet.

L'UIT, l'ISO et la CEI ne prennent aucune position concernant l'évidence, la validité et le domaine d'application de ce droit de brevet.

Le détenteur de ce droit de brevet a assuré l'UIT, l'ISO et la CEI qu'il était disposé à négocier, avec des requérants situés dans le monde entier, l'octroi de licences selon des termes et conditions raisonnables et non discriminatoires. A cet égard, la déclaration du détenteur de ce droit de brevet est enregistrée auprès de l'UIT, de l'ISO et de la CEI. De plus amples renseignements peuvent être obtenus auprès des sociétés énumérées dans l'Annexe M.

L'attention est appelée sur la possibilité que certains éléments de la présente Recommandation | Norme internationale puissent faire l'objet de droits de brevet autres que ceux qui sont identifiés dans l'Annexe M. L'UIT, l'ISO et la CEI ne doivent pas être tenues responsables de l'identification de tout ou partie de tels droits de brevet.

**NORME INTERNATIONALE
RECOMMANDATION UIT-T**

**Technologies de l'information – Système de codage d'images JPEG 2000: outils
d'interactivité, interfaces de programmes d'application et protocoles**

1 Domaine d'application

La présente Recommandation | Norme internationale définit de façon extensible des syntaxes et des méthodes permettant l'interrogation à distance et la modification facultative de flux et de fichiers à codage JPEG 2000 conformément à leur définition, donnée dans les parties ci-après de l'ISO/CEI 15444:

- Rec. UIT-T T.800 | ISO/CEI 15444-1:2004, contenant la définition d'un flux à codage JPEG 2000 et d'un format de fichier JP2.
- La famille des formats de fichier JPEG 2000, tels que définis dans les parties suivantes de l'ISO/CEI 15444.

Dans la présente Recommandation | Norme internationale, les syntaxes et méthodes définies sont désignées par le terme de "*protocole interactif JPEG 2000*" ou "JPIP", tandis que les applications interactives utilisant ce protocole JPIP sont désignées par le terme de "*systèmes JPIP*".

Le protocole JPIP spécifié se compose d'une série structurée d'interactions entre un client et un serveur distant au moyen desquelles des métadonnées de fichier d'image, une structure et des flux partiels ou entiers de codage d'images peuvent être échangés de façon efficace au cours d'une communication. La présente Recommandation | Norme internationale contient les définitions de la sémantique et des valeurs à échanger. Elle suggère également comment ces valeurs peuvent être transmises au moyen d'une variété de réseaux de transport existants.

Avec le protocole JPIP, les tâches suivantes peuvent être accomplies de diverses façons compatibles:

- L'échange de capacités.
- La négociation des capacités à utiliser dans une session.
- La requête et le transfert des éléments suivants à partir d'une variété de conteneurs tels que les fichiers de la famille JPEG 2000, les flux à codage JPEG 2000 et autres fichiers conteneurs:
 - segments de données sélectives;
 - structures sélectives et définies;
 - parties d'une image ou de ses métadonnées associées.

2 Références normatives

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

- Recommandation UIT-T T.800 (2002) | ISO/CEI 15444-1:2004, *Technologies de l'information – Système de codage d'images JPEG 2000: Système de codage noyau*.
- Recommandation UIT-T T.801 (2002) | ISO/CEI 15444-2:2004, *Technologies de l'information – Système de codage d'images JPEG 2000: Extensions*.
- Recommandation UIT-T T.802 (2005) | ISO/CEI 15444-3:2005, *Technologies de l'information – Système de codage d'images JPEG 2000 – Images JPEG 2000 animées*.
- ISO/CEI 15444-6:2003, *Technologies de l'information – Système de codage d'images JPEG 2000 – Partie 6: Format de fichier d'image composite*.

ISO/CEI 15444-9:2005 (F)

- IETF RFC 768 (1980), *Protocole du service de datagrammes d'utilisateur*. Disponible sur la toile numérique mondiale: <<http://www.ietf.org/rfc/rfc0768.txt>>.
- IETF RFC 793 (1981), *Transmission Control Protocol*. Disponible sur la toile numérique mondiale: <<http://www.ietf.org/rfc/rfc0793.txt>>.
- IETF RFC 2046 (1996), *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Disponible sur la toile numérique mondiale: <<http://www.ietf.org/rfc/rfc2046.txt>>.
- IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF*. Disponible sur la toile numérique mondiale: <<http://www.ietf.org/rfc/rfc2234.txt>>.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*. Disponible sur la toile numérique mondiale: <<http://www.ietf.org/rfc/rfc2396.txt>>.
- IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*. Disponible sur la toile numérique mondiale: <<http://www.ietf.org/rfc/rfc2616.txt>>.

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

3.1 Définitions relatives à la Partie 1 de la norme JPEG 2000

Les définitions figurant au § 3 de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004 et au § 3 de la Rec. UIT-T T.801 | ISO/CEI 15444-2:2004 s'appliquent également à la présente Recommandation | Norme internationale.

3.2 Définitions relatives au protocole HTTP

Les définitions suivantes visent à correspondre au protocole HTTP/1.1. En cas de différence, ces définitions doivent être utilisées.

3.2.1 connexion: circuit virtuel de couche transport, établi entre deux programmes aux fins de la communication.

3.2.2 entité: information transférée comme charge utile d'une requête ou d'une réponse. Une entité se compose d'une partie de métadonnées se présentant sous la forme de champs d'en-tête d'entité et d'autre part d'un contenu se présentant sous la forme d'un corps d'entité.

3.2.3 (serveur) intermédiaire: programme intermédiaire qui agit à la fois comme serveur distant et comme client aux fins de la formulation de requêtes pour le compte d'autres clients. Les requêtes sont desservies en interne ou sont transmises, avec éventuelle conversion, à d'autres serveurs distants.

3.3 Définitions relatives au protocole JPIP

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent. Dans certains cas, ces définitions diffèrent de celles qui sont utilisées dans d'autres normes et/ou Recommandations.

3.3.1 cache (antémémoire) (côté client): stockage des segments de données JPIP. Le client peut avoir un cache limité et peut de temps en temps avoir à purger des segments antémémorisés de données JPIP.

3.3.2 antémémorisable: une réponse est antémémorisable si un cache est autorisé à mémoriser une copie du message de réponse afin de l'utiliser lors de la réponse à des requêtes ultérieures. Même si une ressource est antémémorisable, il peut y avoir des contraintes supplémentaires quant à la question de savoir si un cache peut utiliser la copie antémémorisée en présence d'une requête particulière.

3.3.3 modèle de cache (côté serveur): estimation par le serveur distant des portions des segments de données disponibles dans le cache du client. Le serveur distant peut ajouter des éléments à son estimation du cache du client parce qu'il part du principe que la remise a été réussie, ou parce qu'il a reçu des acquittements des données transmises, ou parce qu'il a reçu des instructions de mise à jour du modèle de cache.

3.3.4 canal: mécanisme permettant de regrouper des requêtes et des réponses de façon qu'une seule requête/réponse soit active à un instant donné dans le groupe. De multiples requêtes et réponses simultanées nécessitent des canaux multiples.

3.3.5 client: programme qui établit des connexions afin d'envoyer des requêtes.

3.3.6 région d'image à flux codé: intersection entre l'image et la région, définie par le décalage et par la taille de la région. La région d'image à flux codé peut être vide (sans zone).

3.3.7 segment de données: ensemble d'octets du même type de données qui peut être remis partiellement.

- 3.3.8 flux codé incrémentiel:** représentation du flux codé comme un ensemble de segments de données (en-tête principal, en-tête de pavé, segments de données de district ou de pavé) possédant le même identificateur de flux codé.
- 3.3.9 table d'indices JPIP:** boîte de format de fichier qui donne des informations sur l'emplacement de portions d'un fichier ou d'un flux codé.
- 3.3.10 cible logique:** représentation spécifique d'une certaine ressource nommée originale, ou série d'octets extraite d'une certaine ressource nommée originale, vers laquelle la requête JPIP est dirigée. Cette représentation spécifique pourrait être transcodée à partir de la ressource nommée originale.
- 3.3.11 message:** ensemble d'octets extraits d'un même segment de données, avec l'en-tête identifiant ces octets et ce segment de données.
- 3.3.12 flux codé à traiter:** représentation du flux codé comme un seul segment de métadonnées.
- 3.3.13 requête:** groupe de champs et de valeurs envoyé par le client au serveur distant afin d'obtenir des portions d'une image ou des métadonnées.
- 3.3.14 ressource:** objet ou service de données de réseau qui peut être désigné par un identificateur URI. Cible du protocole HTTP.
- 3.3.15 réponse:** octets envoyés par le serveur distant au client après réception d'une requête.
- 3.3.16 serveur distant:** programme applicatif qui accepte des connexions afin de desservir des requêtes par renvoi de réponses. Un programme quelconque peut être à la fois client et serveur distant; l'utilisation de ces termes se rapporte seulement au rôle joué par le programme pour une connexion particulière, plutôt qu'aux capacités de ce programme en général.
- 3.3.17 session:** ensemble des requêtes et réponses s'appliquant à la même ressource, pour lequel le serveur distant tient à jour un modèle de cache.
- 3.3.18 mode session:** processus au cours duquel le serveur distant tient à jour un modèle de cache.
- 3.3.19 sans description d'état:** requête isolée dans laquelle le serveur distant ne fait pas appel à un modèle de cache lors de la détermination de la réponse.
- 3.3.20 cible:** identification logique de données JPIP. Nom de la cible principale (souvent celui d'un fichier dans le serveur distant).
- NOTE – Des fichiers ou flux à codage JPEG 2000 peuvent être disponibles dans de multiples représentations (par exemple, selon le type de retour, selon la taille de district) ou peuvent varier d'autres façons, chacune étant identifiée comme une unique cible logique.
- 3.3.21 en-tête de pavé:** tous les en-têtes des éléments d'un pavé spécifique.
- 3.3.22 fenêtre de visualisation:** portion des données d'image recherchées par le client, exprimées par la combinaison des champs suivants, qui apparaissent dans la requête: taille de la région, décalage, longueur de trame, flux codé, contexte du flux codé, fréquence d'échantillonnage, région d'image (ROI) et couches. La fenêtre de visualisation est souvent plus petite que l'ensemble des données d'image. Si une fenêtre de visualisation est impliquée mais non spécifiée, alors elle doit être considérée comme une fenêtre de visualisation de toutes les données iconographiques contenues dans la cible logique.

3.4 Symboles

Pour les besoins de la présente Recommandation | Norme internationale, les symboles suivants s'appliquent. Les symboles définis au § 4 de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004 et au § 4 de la Rec. UIT-T T.801 | ISO/CEI 15444-2:2004 s'appliquent également à la présente Recommandation | Norme internationale.

c	indice (à partir de 0) du composant d'image auquel le district appartient
fx	longueur de trame sur l'axe des x pour une fenêtre de visualisation de requête du client
fy	longueur de trame sur l'axe des y pour une fenêtre de visualisation de requête du client
fx'	longueur de trame sur l'axe des x pour une résolution appropriée du flux codé
fy'	longueur de trame sur l'axe des y pour une résolution appropriée du flux codé
fx''	longueur de trame en format jpx modifié sur l'axe des x pour une résolution appropriée
fy''	longueur de trame en format jpx modifié sur l'axe des y pour une résolution appropriée

ISO/CEI 15444-9:2005 (F)

H_{cod}	hauteur du flux codé telle qu'enregistrée dans la boîte d'en-tête d'image (ihdr) (voir Annexe I.5.3.1/Rec. UIT-T T.800 ISO/CEI 15444-1:2004)
H_{comp}	hauteur du résultat composé, fournie dans la boîte d'options de composition en format JPX (voir Annexe M.11.10.1/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
H_{reg}	hauteur de la couche de composition, telle qu'elle apparaît sur la grille d'enregistrement de la couche de composition
Hs_{inst}	hauteur recadrée
Ht_{inst}	hauteur composée
l	identificateur unique du district dans son flux codé
N_L	nombre de niveaux de décomposition
num_components	nombre de composants codés
num_pavés	nombre de pavés dans le flux codé
ox	décalage sur l'axe des x pour une fenêtre de visualisation de requête du client
ox'	décalage sur l'axe des x pour une région appropriée du flux codé
ox''	décalage en format jpx modifié sur l'axe des x pour une région appropriée
oy	décalage sur l'axe des y pour une fenêtre de visualisation de requête du client
oy'	décalage sur l'axe des y pour une région appropriée du flux codé
oy''	décalage sur l'axe des y en format jpx modifié pour une région appropriée
r	niveau de résolution
s	numéro séquentiel qui désigne le district dans son élément de pavé
sx	longueur de l'axe des x d'une fenêtre de visualisation de requête du client
sx'	longueur de l'axe des x pour une région appropriée du flux codé
sx''	longueur de l'axe des x en format jpx modifié pour région appropriée
sy	longueur de l'axe des y d'une fenêtre de visualisation de requête du client
sy'	longueur de l'axe des y pour une région appropriée du flux codé
sy''	longueur de l'axe des y en format jpx modifié pour une région appropriée
t	indice (à partir de 0) du pavé auquel le district appartient
W_{cod}	largeur du flux codé telle qu'enregistrée dans la boîte d'en-tête d'image (ihdr) (voir Annexe I.5.3.1/Rec. UIT-T T.800 ISO/CEI 15444-1:2004)
W_{comp}	largeur du résultat composé, fournie dans la boîte d'options de composition en format JPX (voir Annexe M.11.10.1/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
W_{reg}	largeur de la couche de composition, telle qu'elle apparaît sur la grille d'enregistrement de la couche de composition
Ws_{inst}	largeur recadrée
Wt_{inst}	largeur composée
XC_{inst}	décalage du recadrage sur l'axe des x fourni par l'instruction correspondante (voir Annexe M.11.10.2.1/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
XO_{inst}	décalage de la composition sur l'axe des x, décrit par l'instruction de composition correspondante (voir Annexe M.11.10.2.1/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
XO_{reg}	décalage sur l'axe des x de l'enregistrement du flux codé

XO_{siz}	décalage horizontal, à partir de l'origine de la grille de référence, du segment correspondant de marqueur SIZ du flux codé
XR_{reg}	facteur d'échantillonnage sur l'axe des x lors de l'enregistrement du flux codé, décrit au début d'une boîte quelconque d'enregistrement de flux codé (voir Annexe M.11.7.7/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
Xs_{iz}	largeur de la grille de référence du segment correspondant de marqueur SIZ du flux codé
XS_{reg}	précision d'enregistrement sur l'axe des x, décrite au début d'une boîte quelconque d'enregistrement de flux codé (voir Annexe M.11.7.7/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
YC_{inst}	décalage du recadrage sur l'axe des y, fourni par l'instruction correspondante (voir Annexe M.11.10.2.1/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
YO_{inst}	décalage de la composition sur l'axe des y, fourni par l'instruction de composition correspondante (voir Annexe M.11.10.2.1/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
YO_{reg}	décalage sur l'axe des y de l'enregistrement du flux codé
YO_{siz}	décalage vertical, à partir de l'origine de la grille de référence, du segment correspondant de marqueur SIZ du flux codé
YR_{reg}	facteur d'échantillonnage du flux codé sur l'axe des y lors de l'enregistrement, décrit au début d'une boîte quelconque d'enregistrement de flux codé (voir Annexe M.11.7.7/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
Ys_{iz}	hauteur de la grille de référence du segment correspondant de marqueur SIZ du flux codé
YS_{reg}	précision d'enregistrement sur l'axe des y, décrite au début d'une boîte quelconque d'enregistrement de flux codé (voir Annexe M.11.7.7/Rec. UIT-T T.801 ISO/CEI 15444-2:2004)

4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes s'appliquent.

ABNF	formalisme de Backus-Naur étendu (<i>augmented Backus-Naur form</i>)
DICOM	Imagerie et communications numériques en médecine (<i>digital imaging and communications in medicine</i>)
DWT	transformation en ondelette discrète (<i>discrete wavelet transformation</i>)
EOR	fin de réponse (<i>end of response</i>)
GT 1	Groupe de travail 1
HTML	langage de balisage hypertexte (<i>hypertext markup language</i>)
IP	protocole Internet (<i>Internet protocol</i>)
JP3D	Partie 10 de la norme JPEG 2000: 3-D et données en virgule flottante
JPIP	protocole interactif JPEG 2000 (<i>JPEG 2000 interactive protocol</i>)
JPP	district JPIP (<i>JPIP precinct</i>)
JPSEC	Partie 8 de la norme JPEG 2000: Codage JPEG 2000 sécurisé
JPT	pavé JPIP (<i>JPIP tile-part</i>)
JPWL	Partie 11 de la norme JPEG 2000: Communications sans fil
JTC 1	Comité technique mixte 1 (<i>joint technical committee 1</i>)
MTF	fonction de transfert de modulation (<i>modulation transfer function</i>)
PDF	format de document portable (<i>portable document format</i>)
SC 29	Sous-Comité 29
SVG	graphe vectoriel modulable (<i>scalable vector graphics</i>)
TCP	protocole de commande de transmission (<i>transmission control protocol</i>)
UDP	protocole datagramme d'utilisateur (<i>user datagram protocol</i>)
UUID	identificateur unique universel (<i>universal unique identifier</i>)

VBAS	segment de longueur variable aligné à l'octet (<i>variable-length byte aligned segment</i>)
XHTML	langage de balisage extensible hypertexte (<i>extensible hypertext markup language</i>)
XML	langage de balisage extensible (<i>extensible markup language</i>)

5 Conventions

5.1 Règles de formalisme ABNF

La présente Recommandation | Norme internationale utilise la notation du formalisme ABNF qui est définie dans le document RFC 2234, y compris les règles essentielles de syntaxe ABNF: ALPHA (lettres), retour de chariot (CR, *carriage return*), CRLF (interligne de la norme Internet), caractères de virgule (CTL, *control characters*), DIGIT (chiffres décimaux), HEXDIG (chiffres hexadécimaux), changement de ligne (LF, *line feed*), espace blanc linéaire (LWSP, *linear white space*) et espace (SP, *space*). Aux fins de la présente Recommandation | Norme internationale, les règles suivantes de formalisme ABNF sont également applicables.

```

NZDIGIT = %x31-39           ; 1 à 9
UPPER = %x41-5A            ; A à Z
LOWER = %x61-7A            ; a à z
UINT = 1*DIGIT
NONZERO = "*"0" NZDIGIT *DIGIT
UINT-RANGE = UINT ["-" [UINT]]
UFLOAT = 1*DIGIT ["." 1*DIGIT]
ENCODED-CHAR = "%" HEXDIG HEXDIG
UUID = 16(HEXDIG)
TOKEN = 1*(ALPHA / DIGIT / "." / "_" )

```

La présente Recommandation | Norme internationale définit également la forme "chemin", qui représente un fichier ou un nom de chemin. Dans le cas général, les valeurs de "chemin" peuvent contenir un caractère quelconque, bien que, pour une architecture donnée de serveur distant, celui-ci doive rejeter tout caractère qui n'est pas légal sur ce serveur particulier. En outre, la forme chemin doit être correctement codée, comme spécifié par la technique de transport.

La forme "UINT-RANGE" spécifie une étendue de valeurs d'entier. Le premier entier contenu dans l'étendue spécifie le début de celle-ci. Si deux valeurs sont spécifiées, la première et la seconde spécifient respectivement le début et la fin de l'étendue. Si seule la première valeur et le caractère de tiret "-" sont spécifiés, l'étendue contient toutes les valeurs supérieures ou égales à la première valeur.

Une valeur numérique précédant immédiatement un élément de formalisme ABNF désigne une répétition du paramètre qui suit le nombre, pendant le nombre de fois indiqué par la valeur numérique, sans espaces intermédiaires entre chaque apparition.

La construction "1#" désigne une ou plusieurs répétitions du paramètre qui suit, dont chaque apparition est séparée par une virgule.

La construction "1\$" désigne une ou plusieurs répétitions du paramètre qui suit, dont chaque apparition est séparée par un point-virgule.

5.2 Règles de formalisme ABNF relatives au format de fichier

```

compatibility-code = 4(ALPHA / DIGIT / "_" / ENCODED-CHAR)
box-type = 4(ALPHA / DIGIT / "_" / ENCODED-CHAR)
box-type-list = "*" / 1#(box-type)

```

La forme "box-type" spécifie les quatre caractères du type de boîte. Chaque caractère alphanumérique (A..Z, a..z ou 0..9) qui est contenu dans le type de boîte est écrit directement dans la chaîne. Si le caractère est un espace (0x20), il doit être codé comme le caractère de soulignement ("_"). Tout autre caractère est remplacé par une chaîne de 3 caractères, composée d'un caractère de pourcentage ("%") suivi de deux chiffres hexadécimaux représentant la valeur hexadécimale du caractère extrait du type de boîte. La forme "compatibility-code" est codée de la même façon qu'une forme "box-type".

La forme "box-type-list" spécifie une liste de types de boîte. Si la valeur d'un champ "box-type-list" est "*", alors ce champ désigne tous les types de boîte.

5.3 Légende des descriptions graphiques de boîtes (pour information)

La description de chaque boîte est suivie par une figure qui montre l'ordre et la relation des paramètres dans cette boîte. La Figure 1 est un exemple de ce type de figure. Un rectangle sert à indiquer les paramètres dans la boîte. La largeur du rectangle est proportionnelle au nombre d'octets contenus dans le paramètre. Un rectangle ombré (bandes obliques) indique que le paramètre est de longueur variable. Deux paramètres avec des indices supérieurs et une zone grise intermédiaire indiquent une plage de plusieurs de ces paramètres. Une séquence de deux groupes de multiples paramètres avec des indices supérieurs séparés par une zone grise indique une plage de ce groupe de paramètres (un ensemble de chaque paramètre contenu dans le groupe, suivi par le prochain ensemble de chaque paramètre contenu dans le groupe). Les paramètres ou boîtes facultatifs seront représentés par un rectangle ombré.

La figure est suivie d'une liste qui décrit la signification de chaque paramètre contenu dans la boîte. Si des paramètres sont répétés, la longueur et la nature de la plage de paramètres sont définies. Par exemple, dans la Figure 1, les paramètres A, B, C et D ont respectivement une longueur de 8 bits, 16 bits, 32 bits et une longueur variable. La notation E^0 et E^{N-1} implique qu'il y a N paramètres différents, E^i , dans une rangée. Les groupes de paramètres F^0 et F^{M-1} et G^0 et G^{M-1} spécifient que la boîte va contenir F^0 , suivi par G^0 , suivi par F^1 et G^1 , et ainsi de suite jusqu'à F^{M-1} et G^{M-1} (M instances de chaque paramètre contenu au total). De même, le champ D est facultatif et ne peut pas être découvert dans cette boîte.

Par ailleurs, dans une figure décrivant le contenu d'une superboîte, des points de suspension (...) serviront à indiquer que le contenu du fichier situé entre deux boîtes n'est pas spécifiquement défini. Une boîte (ou séquence de boîtes) quelconque peut, sauf spécification contraire dans la définition de cette boîte, être découverte à la place des points de suspension.

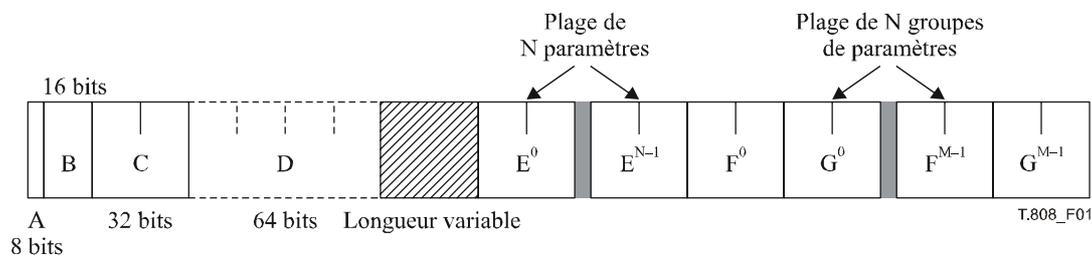


Figure 1 – Exemple des figures descriptives de boîte

Par exemple, la superboîte représentée dans la Figure 2 doit impérativement contenir une boîte AA et une boîte BB. Par ailleurs, la boîte BB doit impérativement suivre la boîte AA. Cependant, on peut découvrir d'autres boîtes entre les boîtes AA et BB. La façon de traiter les boîtes inconnues est examinée dans l'Annexe I.8 de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004.

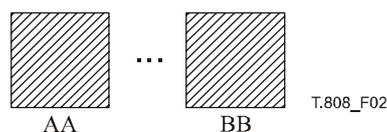


Figure 2 – Exemple des figures descriptives de superboîte

6 Description générale

6.1 Protocole JPIP

La présente Recommandation | Norme internationale décrit les syntaxes et méthodes qui sont utilisées quand un client doit accéder à l'imagerie comprimée par codage JPEG 2000 et aux données associées à cette imagerie, résidant dans un serveur distant activé en protocole JPIP. La présente Recommandation | Norme internationale offre la flexibilité et la fonctionnalité dont la réalisation, lors de transports multiples entre client et serveur, est prévue dans la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004.

Le protocole JPIP définit l'interactivité qui permet de réaliser l'échange efficace de l'imagerie JPEG 2000 et de ses données associées. Le protocole définit les interactions entre client et serveur sur la base d'une requête du client et de la réponse du serveur distant, comme représenté dans la Figure 3. La présente Recommandation | Norme internationale définit la requête JPIP émise par un client et la réponse JPIP des serveurs distants. Les protocoles HTTP/1.1 (RFC 2616), TCP (RFC 793) et UDP (RFC 768) sont représentés comme exemples de transports possibles par le protocole JPIP. Le client utilise une requête de fenêtre de visualisation afin de définir la résolution, la taille,

l'emplacement, les composants, les couches et autres paramètres relatifs à l'image et aux données associées à l'imagerie, qui sont demandées par le client. Le serveur distant répond en remettant l'imagerie et les données y associées au moyen de flux en mode district, de flux en mode pavé, ou d'images entières. Le protocole permet également la négociation, entre client et serveur distant, de capacités et de limitations. Le client peut demander au serveur distant des informations sur une image comme défini dans les tables d'indices JPIP, ce qui permet au client de raffiner sa requête de fenêtre de visualisation en fonction de paramètres propres à l'image (par exemple des requêtes portant sur des séries d'octets). Le modèle de cache du serveur distant est fondé sur les capacités définies par le client et sur l'adaptabilité de la session.

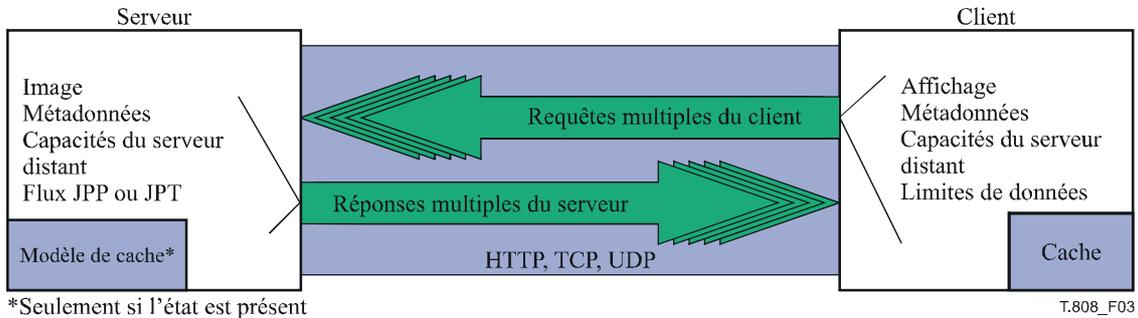


Figure 3 – Vue d'ensemble du protocole JPIP

Ce protocole peut être utilisé lors de plusieurs transports différents, comme représenté dans la Figure 4. La présente Recommandation | Norme internationale contient des annexes informatives sur l'utilisation du protocole JPIP au-dessus des protocoles HTTP et TCP et offre des suggestions relatives à d'autres exemples de réalisation.

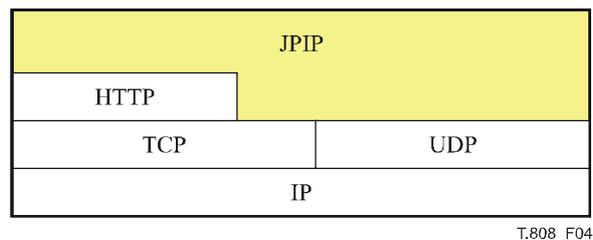


Figure 4 – Empilement du protocole JPIP

Des dispositions ont été incluses au sujet de l'extension du protocole JPIP afin de prendre en charge les normes actuelles de codage JPEG 2000, à savoir la Rec. UIT-T T.802 | ISO/CEI 15444-3: Images JPEG 2000 animées, et l'ISO/CEI 15444-6: Format de fichier d'image composite, ainsi que les futures parties de la norme JPEG 2000 (actuellement JP3D, JPSEC et JPWL).

6.2 Objet

La présente Recommandation | Norme internationale définit la syntaxe et les méthodes requises tant pour le client que pour le serveur. Chaque annexe définit un composant qui est requis afin de réaliser l'interopérabilité et la fonctionnalité entre le client et le serveur distant lors de plusieurs transports. Chaque annexe peut être une exigence du client, une exigence du serveur distant, ou une exigence des deux. Ces annexes sont décrites ci-dessous.

- L'Annexe A décrit les flux en mode pavé et en mode district qui sont requis pour le client comme pour le serveur. Celui-ci est tenu de produire des flux JPP et JPT conformes et de pouvoir interpréter les flux JPP et JPT téléchargés vers l'amont. Le client est tenu d'interpréter et de décoder correctement ces flux. Il est également chargé de produire des flux conformes lors d'un téléchargement amont d'imagerie partielle vers le serveur distant.
- L'Annexe B décrit la session et la modélisation du cache d'une session client/serveur; elle est requise pour le client comme le serveur.
- L'Annexe C définit la syntaxe d'une requête du client. Celui-ci doit produire des requêtes conformes et le serveur distant doit être en mesure d'interpréter et de satisfaire toutes les requêtes conformes.
- L'Annexe D définit la syntaxe de la réponse du serveur distant. Celui-ci doit produire des réponses conformes et le client doit être en mesure de les interpréter.
- L'Annexe E définit la syntaxe et les méthodes permettant de télécharger vers l'amont une image partielle au moyen des systèmes qui utilisent à cette fin le protocole JPIP.

- Les Annexes F, G et H définissent les méthodes et procédures pour les interactions client/serveur en protocole JPIP en présence de plusieurs protocoles de transport différents.
- L'Annexe I définit la syntaxe des informations d'indexation contenues dans une boîte JPEG 2000, qui peuvent être utilisées par un client et par un serveur afin d'accéder plus efficacement à l'imagerie et aux données qui y sont associées.
- L'Annexe J définit comment la présente Recommandation | Norme internationale peut être étendue au moyen de l'enregistrement.
- L'Annexe K décrit plusieurs exemples d'utilisation de la présente Recommandation | Norme internationale pour plusieurs applications différentes.

7 Conformité

La conformité à la présente Recommandation | Norme internationale par un client signifie que les requêtes JPIP de ce client sont bien structurées, valides et conformes à la requête du client JPIP qui est définie par la présente Recommandation | Norme internationale. Les clients doivent prendre en charge tous les types normatifs de requête.

La conformité à la présente Recommandation | Norme internationale par un serveur distant signifie que les réponses JPIP de ce serveur sont bien structurées, valides et conformes à la signalisation de réponse de serveur JPIP qui est définie par la présente Recommandation | Norme internationale. Les serveurs distants doivent prendre en charge tous les types normatifs de requête.

Bien qu'il soit prévu que la présente Recommandation | Norme internationale doit être implémentée de façon que les données iconographiques soient demandées au moyen de requêtes JPIP efficaces sur la base des exigences applicatives du côté client, aucun comportement conforme n'est défini.

De même, les données iconographiques devraient être desservies sur la base d'une réponse JPIP efficace des serveurs distants – minimisant la quantité de données desservies en dehors des intérêts signalés par le client et des données redondantes déjà en possession du client. Cependant, aucun comportement conforme n'est défini.

Il est attendu que les applications de serveur distant puissent réduire l'efficacité par l'envoi de données additionnelles ou de données redondantes, selon la qualité de service du réseau. De telles décisions d'implémentation sont propres à chaque application et confèrent au système JPIP un haut niveau d'utilité. Cependant, la présente Recommandation | Norme internationale ne définit pas la conformité applicable à la configuration de ces décisions d'implémentation.

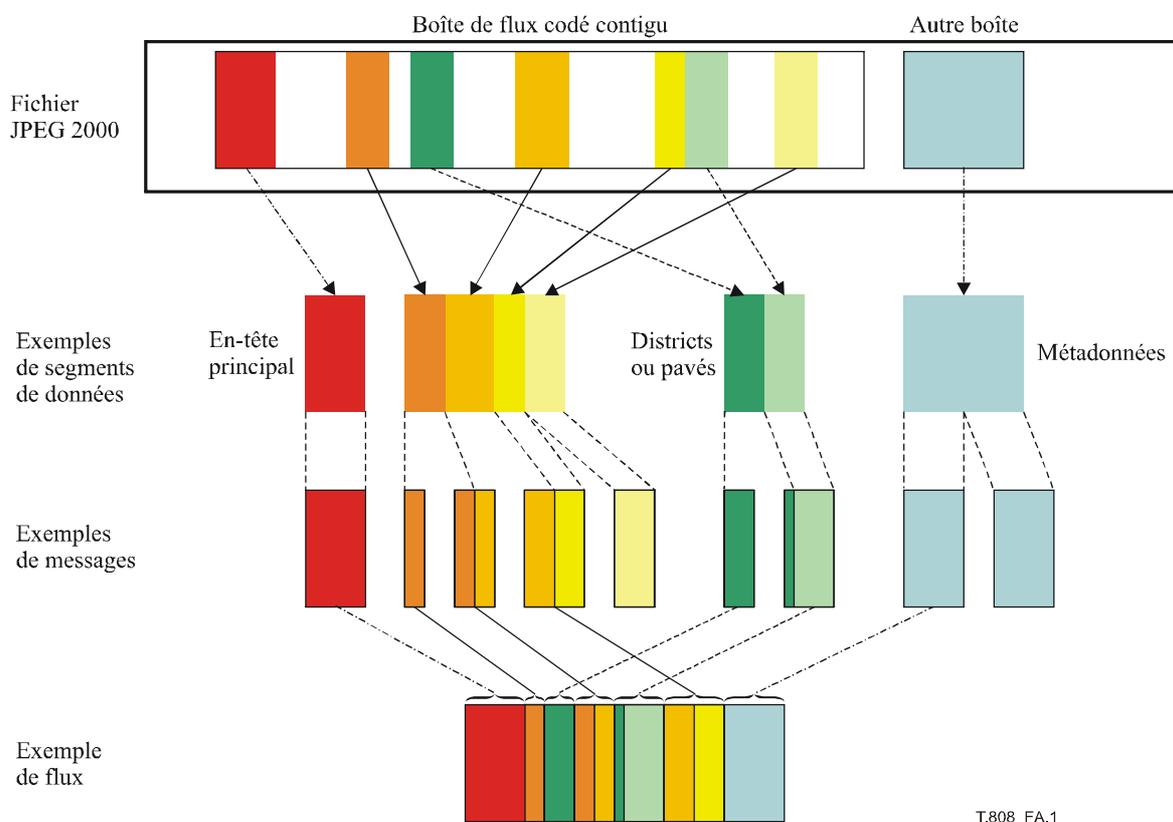
Annexe A

Types de média à flux JPP et JPT

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

A.1 Introduction

Les flux JPP et JPT sont des types de média permettant de présenter dans un ordre arbitraire les flux à codage JPEG 2000 et les données de format de fichier correspondantes. Chaque type de média se compose d'une séquence de messages concaténés dans laquelle chaque message contient une portion d'un même segment de données précédé par un en-tête de message. Les segments de données contiennent de telles portions de représentation d'image comprimée JPEG 2000 qu'il est possible de construire un flux qui représente complètement les informations présentes dans un fichier ou flux codé JPEG 2000. Chaque message est complètement autodéscriptif, de façon que la séquence de messages puisse être fermée à un point quelconque et que ces messages puissent être remis en ordre sous réserve de contraintes minimales sans perdre leur signification. C'est pourquoi les types de média à flux JPP et JPT sont utiles pour les serveurs JPIP. Le protocole JPIP a été particulièrement conçu en fonction de ces types de média. Cette Annexe définit les types de média à flux JPP et JPT sans référence au protocole JPIP.



T.808_FA.1

Figure A.1 – Exemples de relations entre fichier JPEG 2000, segments de données JPIP et flux JPIP (d'après G.J. Colyer et R.A. Clark, IEEE Trans. Consumer Electronics, 49 (2003), p. 850-854)

La Figure A.1 est un exemple montrant la relation entre les flux binaires issus d'un fichier JPEG 2000, des segments de données JPIP et un flux JPIP. Cette figure montre la couleur de l'en-tête principal, codée en rouge, 2 districts avec des paquets codés en nuances d'orange-jaune et de vert, ainsi qu'une boîte de métadonnées codée en bleu. Les messages autodéscriptifs JPIP sont formés à partir de ces segments de données puis concaténés afin de constituer un flux JPIP.

Un flux JPIP se compose d'un ou de plusieurs messages JPIP concaténés. Chaque message JPIP se compose d'un en-tête et d'un corps. L'en-tête fournit des informations descriptives permettant d'identifier le segment de données correspondant. Le corps est constitué de données issues de ce segment de données. A moins qu'une autre signalisation ne soit fournie, le message est la concaténation de l'en-tête avec le corps.

NOTE – Dans la présente Recommandation | Norme internationale, tous les exemples fournis forment des messages binaires par concaténation d'un en-tête et d'un corps. La question de savoir si une autre signalisation est fournie pour l'en-tête et pour le corps dépend de la réalisation du mode de transport et de l'application. Par exemple, une signalisation auxiliaire avec protection variable contre les erreurs peut être implémentée dans les applications de communication sans fil.

A.2 Structure d'en-tête de message

A.2.1 Généralités

Chaque message représente une portion exactement égale à un segment de données. L'en-tête de message se compose d'une séquence de segments de longueur variable alignés à l'octet (VBAS). Chaque segment VBAS se compose d'une séquence d'octets qui ont tous, sauf le dernier, un bit de poids fort (bit 7) égal à 1, comme indiqué dans la Figure A.2. Les 7 bits de poids faible de chaque octet contenu dans le VBAS sont concaténés afin de constituer un flux binaire qui est utilisé de différentes façons pour différents segments VBAS.

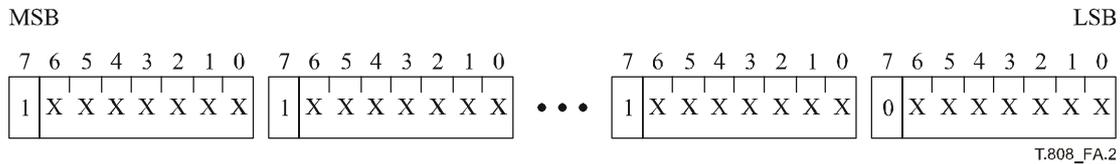


Figure A.2 – Structure des segments VBAS

L'en-tête de message sert à identifier le segment de données, avec sa série d'octets particuliers, qui est représenté par le corps du message. Les en-têtes de message peuvent prendre une forme indépendante et une forme dépendante. La forme indépendante est de type long, où les en-têtes de message sont complètement autodéscriptifs: leur interprétation est indépendante de tout autre en-tête de message. Les en-têtes de message facultatifs à forme dépendante plus courte font usage des informations contenues dans les en-têtes de messages antérieurs. Leur décodage dépend du message précédent. Les applications peuvent choisir d'utiliser les en-têtes de message de forme longue: ces messages peuvent ensuite être réorganisés dans un ordre quelconque. En variante, les applications peuvent utiliser les en-têtes de message de forme plus courte, qui dépendent en fait de précédents en-têtes de message: ces messages sont plus courts mais vont créer des résultats erronés s'ils ne sont pas organisés en séquence correcte lors de leurs décodage. La question de savoir si l'ordre séquentiel des messages reçus peut ou non être considéré comme fiable et s'il convient – dans ce cas – de faire usage de la forme courte des en-têtes de message, dépend d'une décision prise au niveau de l'application.

L'en-tête de message se compose des segments VBAS suivants (les segments VBAS facultatifs sont identifiés par l'utilisation de crochets):

Identificateur de segment [, Class] [, CSn], Msg-Offset, Msg-Length [, Aux]

L'existence des segments VBAS "Class" et "CSn" est déterminée par examen de l'identificateur de segment VBAS. L'existence du segment VBAS "Aux" est déterminée par le segment VBAS "Class" actuel, ou par le précédent segment VBAS "Class" s'il n'y a aucun segment VBAS "Class" dans les en-têtes de message actuels.

L'identificateur de segment VBAS remplit plusieurs rôles. Les bits 6 et 5 du premier octet de l'identificateur de segment VBAS, étiqueté 'b' dans la Figure A.3, indiquent si les segments VBAS "Class" et "CSn" sont présents dans l'en-tête de message. Le Tableau A.1 définit les valeurs binaires et leur signification.

Le bit 4 du premier octet de l'identificateur de segment VBAS, étiqueté 'c' dans la Figure A.3, indique si le message considéré contient ou non le dernier octet du segment de données associé: la valeur '0' signifie qu'il ne contient pas le dernier octet du segment de données; la valeur '1' indique qu'il contient le dernier octet du segment de données. La réception d'un message avec ce bit activé permet la détermination de la longueur du segment de données complet, bien qu'elle n'implique pas que le flux JPP ou JPT complet contienne suffisamment de messages pour rassembler tous les octets extraits de ce segment de données.

Les 4 bits restants du premier octet et les 7 bits de poids faible de tous les octets restant contenus dans l'identificateur de segment VBAS (étiqueté 'd' dans la Figure A.3) forment un "identificateur-dans-la-classe" qui sert à identifier de façon unique le segment de données dans sa classe, comme décrit en A.2.3.

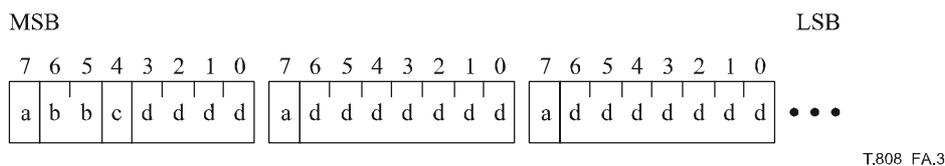


Figure A.3 – Structure des segments VBAS à identificateur de segment

Tableau A.1 – Identificateur de segment: indication supplémentaire de segment VBAS

Bits indicateurs 'bb'	Signification
00	Interdit.
01	Aucun segment VBAS "Class" ou "CSn" n'est présent dans l'en-tête de message.
10	Un segment VBAS "Class" est présent mais un segment "CSn" n'est pas présent dans l'en-tête de message.
11	Les segments VBAS "Class" et "CSn" sont tous les deux présents dans l'en-tête de message.

Le segment VBAS "Class", s'il est présent, offre un identificateur de classe de message qui est un entier non négatif, formé par concaténation des 7 bits de poids faible de chaque octet du segment VBAS, dans l'ordre gros-boutiste. Si le segment VBAS "Class" n'est pas présent, l'identificateur de classe de message est inchangé par rapport à celui qui était associé au message précédent. Si le segment VBAS "Class" n'est pas présent et qu'il n'y ait aucun message précédent, l'identificateur de classe de message est 0. Les identificateurs de classe de message valides sont décrits en A.2.2.

S'il est présent, le segment VBAS "CSn" désigne l'indice (à partir de 0) du flux codé auquel le segment de données appartient. L'indice du flux codé est formé par concaténation des 7 bits de poids faible de chaque octet du segment VBAS, dans l'ordre gros-boutiste. Si le segment VBAS "CSn" n'est pas présent, l'indice du flux codé est inchangé à partir du message précédent. Si le segment VBAS "CSn" n'est pas présent et qu'il n'y ait aucun message précédent, l'indice du flux codé est 0.

Les segments VBAS "Msg-Offset" et "Msg-Length" représentent chacun des valeurs non négatives d'entier, formées par concaténation des 7 bits de poids faible de chaque octet contenu dans le segment VBAS, dans l'ordre gros-boutiste. L'entier "Msg-Offset" désigne le décalage des données dans ce message à partir du début du segment de données. L'entier "Msg-Length" désigne le nombre total d'octets contenus dans le corps du message.

Un segment VBAS "Aux" peut être présent. Sa présence et la signification de celle-ci sont déterminées par l'identificateur de classe de message qui a été découvert dans l'identificateur de segment VBAS, comme expliqué en A.2.2. S'il est présent, le segment VBAS "Aux" représente une valeur non négative d'entier, formée par concaténation des 7 bits de poids faible de chaque octet contenu dans le VBAS, dans l'ordre gros-boutiste.

NOTE – Les informations contenues dans le segment VBAS "Aux" ne peuvent pas affecter la longueur du corps du message.

A.2.2 Identificateurs de classe de message

Les identificateurs de classe de message définis par la présente Recommandation | Norme internationale sont les entiers non négatifs représentés dans le Tableau A.2. L'interprétation des classes de segment de données auxquelles ils se rapportent est décrite en A.3. Toutes les autres valeurs d'identificateur de classe de message sont réservées et les messages associés devraient être omis par les décodeurs ne reconnaissant pas cette valeur.

Les identificateurs de classe sont choisis de façon qu'un segment VBAS "Aux" ne soit présent que si l'identificateur est impair. Cette propriété permet d'analyser correctement la sémantique des en-têtes de message non reconnus et d'en omettre le contenu.

Les messages étendus et non étendus à segments de données de district ont exactement la même interprétation car ils se rapportent exactement aux mêmes segments de données de district. Les messages étendus de district comprennent un segment VBAS "Aux" qui désigne le nombre de paquets complets (couches de qualité) qui seront disponibles pour le district si les octets contenus dans ce message ont été combinés avec tous les octets précédents du même district. Si ce message contient également le dernier octet du segment de données, le segment VBAS "Aux" indique le nombre total de couches de qualité qui est associé au district dans le flux codé original. Sinon, le segment VBAS "Aux" indique la couche de qualité à laquelle appartient l'octet qui suit immédiatement le dernier octet contenu dans le message. Les informations contenues dans le segment VBAS "Aux" peuvent être utiles à certains clients.

Tableau A.2 – Identificateurs de classe pour différentes classes de message de segment de données

Identificateur de classe	Classe de message	Classe de segment de données	Type de flux
0	Message de segment de données de district	Segment de données de district	Flux JPP seulement
1	Message étendu de segment de données de district	Segment de données de district	Flux JPP seulement
2	Message de segment de données d'en-tête de pavé	Segment de données d'en-tête de pavé	Flux JPP seulement
4	Message de segment de données de pavé	Segment de données de pavé	Flux JPT seulement
5	Message étendu de segment de données de pavé	Segment de données de pavé	Flux JPT seulement
6	En-tête principal message de segment de données	Segment de données d'en-tête principal	Flux JPP et JPT
8	Message de segment de métadonnées	Segment de métadonnées	Flux JPP et JPT

Les messages étendus et non étendus de segment de données de pavé ont exactement la même interprétation car ils se rapportent exactement aux mêmes segments de données de pavé. Les messages étendus de pavé comprennent un segment VBAS "Aux" qui désigne le plus petit nombre n tel que, dans tous les composants pour lesquels le niveau de résolution ($N_L - n$) n'est pas négatif, ce niveau ($N_L - n$) et tous les niveaux de résolution inférieurs ont été complétés quand les octets contenus dans ce message sont combinés avec tous les octets précédents du même pavé, où N_L est le nombre de niveaux de décomposition, qui peut varier par composant. Si aucun niveau de résolution de composant n'a été complété, la valeur du segment VBAS "Aux" est égale à l'unité plus la valeur maximale de N_L dans tous les composants. La valeur zéro est atteinte quand toutes les résolutions ont été complétées dans tous les composants. Etant donné que les résolutions n'apparaissent pas nécessairement en ordre dans un pavé, certains niveaux de résolution supérieurs à la valeur signalée par le VBAS peuvent avoir été complétés, mais cela ne peut pas être déterminé à partir de l'en-tête de message. Les informations contenues dans le segment VBAS "Aux" peuvent être utiles à certains clients.

A.2.3 Identificateurs-dans-la-classe

Les 4 bits de poids faible du premier octet et les 7 bits de poids faible de tous les autres octets extraits de l'identificateur de segment VBAS sont concaténés dans l'ordre gros-boutiste afin de constituer un même mot, possédant $7k-3$ bits, où k est le nombre d'octets contenus dans le VBAS. Ce mot représente un entier non signé qui sert à identifier de façon unique le segment de données dans sa classe et dans son flux codé. Le § A.3 décrit les diverses classes de segment de données, de même que les identificateurs correspondants dans la classe.

A.3 Segments de données

A.3.1 Introduction

Les segments de données contiennent des portions de données de fichier ou de flux codé JPEG 2000. Ces portions peuvent être fondées sur des éléments d'imagerie, comme des données en mode district, des données en mode pavé et des en-têtes. Elles peuvent également être fondées sur des métadonnées. Chaque segment de données est traité, quel que soit son contenu, en tant que flux binaire individuel.

A.3.2 Segments de données de district

A.3.2.1 Format d'un segment de données de district

Les segments de données de district n'apparaissent que dans le type de média à flux JPP. Chaque segment de données de district correspond à un même district dans un même flux codé. L'identificateur-dans-la-classe est défini par l'équation (A-1).

$$I = t + (c + s \times \text{num_components}) \times \text{num_tiles} \quad (\text{A-1})$$

où:

- I est l'identificateur unique du district dans son flux codé
- t est l'indice (à partir de 0) du pavé auquel le district appartient
- c est l'indice (à partir de 0) du composant d'image auquel le district appartient
- s est un numéro séquentiel qui désigne le district dans son élément de pavé.

Dans chaque élément de pavé, les districts se font attribuer des nombres séquentiels contigus, *s*, comme suit. Tous les districts du plus bas niveau de résolution (celui qui contient seulement les échantillons de sous-bande LL) sont d'abord ordonnés en séquence, à partir de 0, conformément à un ordre de balayage. Les districts extraits de chaque niveau de résolution successif sont ordonnés en séquence tour à tour, de nouveau conformément à un ordre de balayage dans leur niveau de résolution.

Il s'ensuit qu'un identificateur de district de valeur 0 se rapporte au district du coin gauche supérieur extrait de la sous-bande LL du composant d'image 0 contenu dans le pavé 0.

Chaque segment de données de district correspond à la chaîne d'octets formée par concaténation de tous les paquets de flux codé, complètement équipés de tous les en-têtes de paquet applicables qui appartiennent au district. Il est concevable que les en-têtes de paquet soient imbriqués dans des segments marqueurs PPM ou PPT, qui doivent alors appartenir à des segments de données à en-tête principal ou à en-tête de pavé, auquel cas le segment de données de district ne contiendra que des corps de paquet. En tout état de cause, le flux de données de district devrait coïncider avec le segment d'octets contigu qui sera découvert dans un flux à codage JPEG 2000 possédant une des séquences de progression subordonnées à chaque couche (CPRL, PCRL ou RPCL).

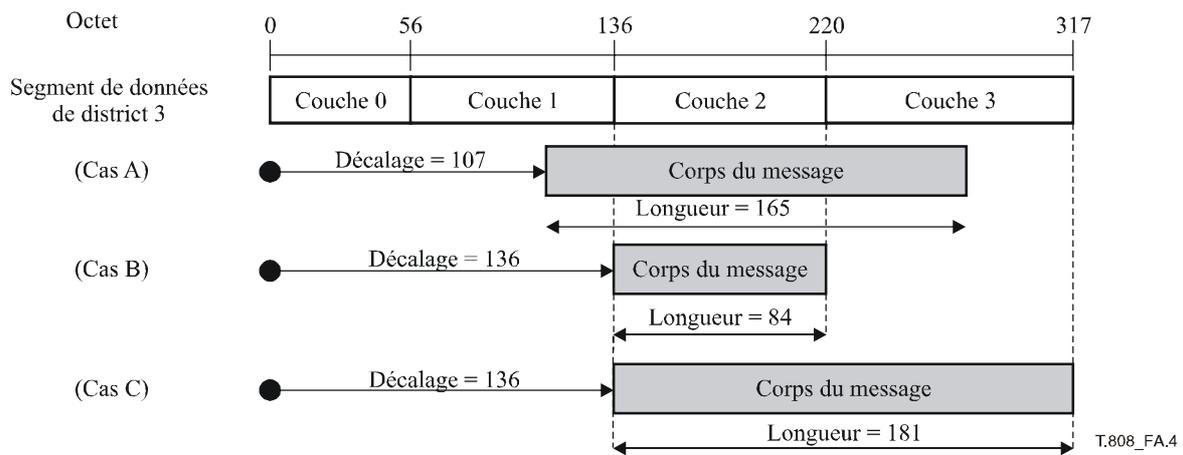


Figure A.4 – Exemple de segment de données de district

A.3.2.2 Exemple de segment de données de district (pour information)

La Figure A.4 montre un exemple de segment de données de district (identificateur-dans-la-classe 3) avec 4 couches de qualité (ou paquets).

Pour les cas A, B et C, l'en-tête de message est représenté ci-dessous sur la base de structures de messages étendus et non étendus de segment de données de district. Les données soulignées désignent le segment VBAS "Aux" permettant d'identifier le nombre de couches qui sont complétées par le message.

(Cas A)

En-tête non étendu: 00100011 01101011 10000001 00100101 xxxxxxxx ...

Le bit initial 0 indique qu'un seul octet est utilisé dans l'identificateur de segment VBAS. Les deux bits suivants ("01") indiquent qu'aucun segment VBAS de type "Class" ou "CSn" n'est présent. Le prochain bit "0" indique que le segment de données n'est pas complété par ce message. Les bits restants du premier octet ("0011") indiquent que l'identificateur de segment est 3. Le premier bit du second octet indique que le segment VBAS de type "Msg-Offset" n'utilise qu'un seul octet. Les 7 bits suivants ("1101011") signifient que le décalage est 107. Le premier bit du 3ème octet indique que cet octet et au moins le prochain octet font tous les deux partie du segment VBAS "Msg-Length". Le bit 0 commençant le 4ème octet indique que celui-ci est le dernier octet du segment VBAS "Msg-Length". Ainsi, tous les bits de poids faible à partir des 3ème et 4ème octets sont concaténés afin de déterminer la longueur. Dans ce cas, "0000001 0100101" = 165.

En-tête étendu: 01000011 00000001 01101011 10000001 00100101 00000011 xxxxxxxx ...

(Cas B)

En-tête non étendu: 00100011 10000001 00001000 01010100 xxxxxxxx ...

En-tête étendu: 01000011 00000001 10000001 00001000 01010100 00000011 xxxxxxxx ...

(Cas C)

En-tête non étendu: 00110011 10000001 00001000 10000001 00110101 xxxxxxxx ...

En-tête étendu: 01010011 00000001 10000001 00001000 10000001 00110101 00000100 xxxxxxxx ...

Noter que, comme les données de réponse contiennent le dernier octet du segment de données indiqué dans le cas C, l'identificateur de segment VBAS signale qu'il s'agit d'un message "complété".

A.3.3 Segments de données d'en-tête de pavé

Les segments de données d'en-tête de pavé n'apparaissent que dans le type de média à flux JPP. Pour les segments de données appartenant à cette classe, l'identificateur-dans-la-classe contient l'indice (à partir de 0) du pavé auquel le segment de données se rapporte. Ce segment de données se compose de marqueurs et de segments marqueurs pour le pavé n . Il ne doit pas contenir de segment de marqueur SOT. L'inclusion de marqueurs SOD est facultative. Ce segment de données peut être formé à partir d'un flux codé légal, par concaténation de tous les segments marqueurs – sauf SOT et POC – dans tous les en-têtes se rapportant au pavé n .

A.3.4 Segments de données de pavé

Les segments de données de pavé ne doivent être utilisés qu'avec le type de média à flux JPT. Pour les segments de données appartenant à cette classe, l'identificateur-dans-la-classe est l'indice (à partir de 0) du pavé auquel le segment de données appartient. Chaque segment de données de pavé correspond à la chaîne d'octets formée par concaténation de tous les éléments de pavés appartenant au pavé, dans l'ordre, complets avec leur marqueur SOT, SOD et tous les autres segments marqueurs correspondants.

A.3.5 Segment de données d'en-tête principal

Les deux types de média JPP et JPT utilisent le segment de données d'en-tête principal. Pour les segments de données appartenant à la classe des en-tête principaux de flux codé (variantes complétées ou non complétées), l'identificateur-dans-la-classe doit être 0. Ce segment de données se compose d'une liste concaténée de tous les marqueurs et segments marqueurs contenus dans l'en-tête principal, à partir du marqueur SOC. Il ne contient aucun marqueur de type SOT, SOD ou EOC.

A.3.6 Segment de métadonnées

A.3.6.1 Introduction aux segments de métadonnées

Les deux types de média, JPP et JPT, utilisent des segments de métadonnées qui servent à acheminer des métadonnées à partir de la cible logique qui contient le(s) flux codé(s) dont des éléments peuvent être référencés par d'autres segments de données associés au flux JPP ou JPT. Aux fins de la présente Recommandation | Norme internationale, le terme "métadonnées" se rapporte à tout ensemble de "boîtes" extraites d'un fichier de la famille JPEG 2000. L'indice du flux codé doit être ignoré dans tout message qui contient l'identificateur de classe de segment de métadonnées.

A la différence des identificateurs numériques utilisés pour d'autres types de segment de données, les identificateurs de segment de métadonnées ne s'appliquent pas algorithmiquement à certaines structures ou à certains décalages d'octet d'un format de fichier. Le serveur distant peut choisir tout identificateur numérique pour un segment de métadonnées particulier. La seule et unique exception à ce qui précède est que le segment de métadonnées contenant la racine de la cible logique doit recevoir un identificateur de valeur 0.

NOTE – Le mécanisme d'affectation dépend de l'implémentation, mais il est suggéré, à titre d'information, que les serveurs distants attribuent des identificateurs de segment utilisant des nombres consécutifs.

A.3.6.2 Subdivision en segments de métadonnées d'une cible logique contenant un fichier JPEG 2000

Toutes les métadonnées pourraient théoriquement être incluses dans le segment de métadonnées 0. Dans ce cas, toutes les boîtes extraites de la cible logique appartiendraient au segment de métadonnées 0, apparaissant dans leur ordre original. Etant donné que le format des fichiers de la famille JPEG 2000 n'est pas autre chose qu'une séquence de boîtes, cela signifie en fait que le segment de métadonnées 0 se composera de la totalité de la cible logique. Plus généralement, il est cependant utile de subdiviser la cible logique en fragments pouvant être transmis de façon gérable. Cela permet à des serveurs d'image d'omettre délibérément les portions de la cible logique qui ne sont pas actuellement requises par un client. A cette fin, le protocole JPIP définit un nouveau type de boîte spécial, appelé "boîte générique", qui sert à identifier la taille et le type d'une boîte à partir de la cible logique, tout en pointant sur un autre segment de données englobant le contenu de cette boîte. Les génériques sont également en mesure de représenter des flux codés à partir de la cible logique. Ce point est particulièrement important compte tenu du fait que les données comprimées représentées par un quelconque flux codé peuvent être acheminées progressivement au moyen des autres types de segment de données (segments de données d'en-tête et segments de données de district ou à pavé).

Formellement, le segment de métadonnées 0 se compose de toutes les boîtes extraites de la cible logique, apparaissant dans leur ordre original, sauf qu'un générique peut remplacer une boîte quelconque. La boîte générique contient l'en-tête original de la boîte qui a été remplacée, ainsi que l'identificateur du segment de métadonnées qui englobe le contenu de cette boîte, non compris l'en-tête proprement dit. Chaque segment de métadonnées, autre que le segment de métadonnées 0, doit se composer du contenu d'une certaine boîte, dont l'en-tête apparaît dans le générique qui se rapporte à ce segment de données. Ce contenu de boîte peut lui-même comprendre des sous-boîtes, dont l'une quelconque peut être remplacée par d'autres génériques.

Le système de couleurs ci-après sera utilisé dans les illustrations des exemples de segment de métadonnées (Figure A.5).

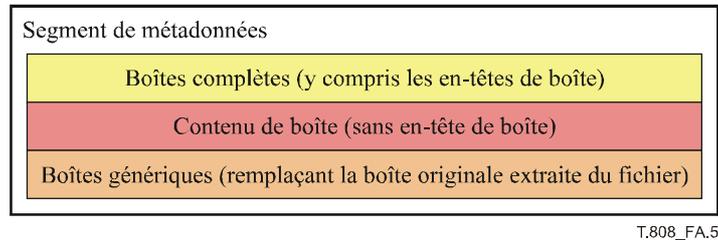


Figure A.5 – Exemple de système colorimétrique de segment de métadonnées

Par exemple, considérons un simple fichier JP2 avec la structure de boîte suivante (Figure A.6).

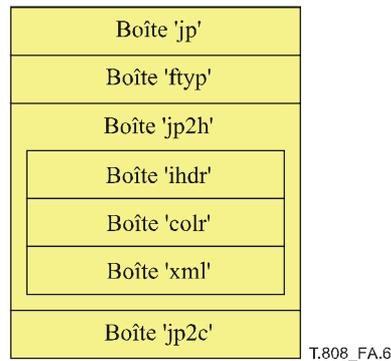


Figure A.6 – Echantillon de fichier JP2

Ce fichier peut être subdivisé en trois segments de métadonnées: l'un afin de représenter le niveau sommital du fichier original (segment de données 0); un autre afin de représenter la boîte d'en-tête JP2; et le troisième afin de représenter le flux codé. Cette subdivision est représentée dans la Figure A.7.

Alors que le contenu de tout segment de métadonnées doit être celui de la boîte ou du fichier représenté par ce segment, les données réellement contenues dans cette boîte ou dans ce fichier peuvent théoriquement varier selon le type de boîte. Par exemple, dans le segment de métadonnées 1 de la Figure A.7, représentant le contenu de la boîte d'en-tête de fichier JP2, ce contenu est en fait une série d'autres boîtes complètes, car cette boîte d'en-tête JP2 est une superboîte. Aucune donnée autre que la série de ces boîtes complètes ne peut être découverte dans le segment de métadonnées 1, car il n'y a aucune autre donnée dans la boîte d'en-tête JP2. En revanche, les données se trouvant à l'intérieur du segment de métadonnées 2 sont le contenu brut de la boîte de flux codé contigu, sans en-têtes de boîte, parce que cette boîte n'est pas une superboîte.

Un point particulièrement intéressant à noter à partir de l'exemple de la Figure A.7 est que l'accès à des données de flux codé peut être fourni de deux façons. Le second segment générique sert à remplacer la boîte de flux codé contigu (jp2c) dans le fichier original. Il désigne le segment de métadonnées 2 comme englobant le contenu original de cette boîte, c'est-à-dire le flux codé non traité proprement dit. Afin de faciliter les descriptions dans la présente Recommandation | Norme internationale, cette représentation doit être appelée "flux codé non traité". Les flux codés non traités sont desservis à partir de segments de métadonnées.

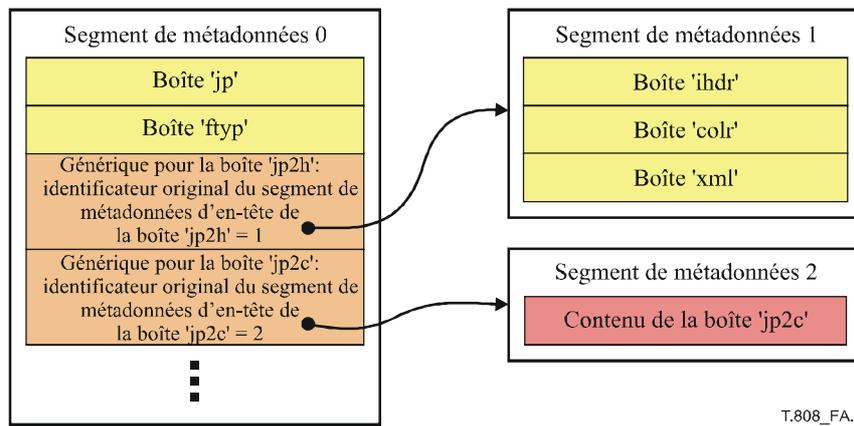


Figure A.7 – Echantillon de fichier JP2 subdivisé en trois segments de métadonnées

Le générique peut également offrir un identificateur de flux codé. Tous les segments de données appartenant aux classes de segment de données d'en-tête principal, d'en-tête de pavé, de district ou de pavé, possédant ce même identificateur de flux codé, acheminent des données comprimées associées au flux codé qui a été découvert dans le segment de métadonnées 2. Afin de faciliter les descriptions dans la présente Recommandation | Norme internationale, cette représentation est dite "à flux codé incrémentiel". Les flux codés incrémentiels sont desservis à partir de ces segments de données.

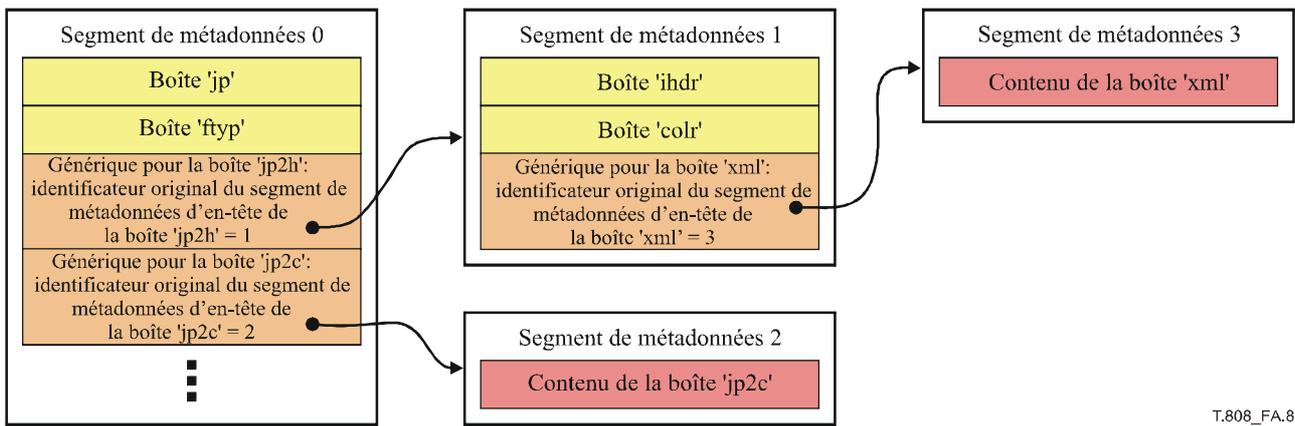
En général, les génériques qui font référence à des données de flux codé peuvent le faire soit en faisant référence à un segment distinct de métadonnées (flux codé non traité), ou en fournissant un identificateur de flux codé (flux codé incrémentiel), ou par ces deux méthodes. Même si celles-ci sont fournies, les données de flux JPP ou JPT disponibles chez un client ou agent de rendu d'image pourraient n'être que le contenu du flux codé non traité, ou n'être que les données extraites du flux codé incrémentiel. En outre, si les deux versions du même flux codé (non traité et incrémentiel) sont disponibles, il n'y a aucune garantie que ces deux représentations auront des paramètres de codage compatibles. Seuls les échantillons d'image reconstruits, qui sont associés aux deux représentations, sont garantis cohérents.

Il est également possible d'utiliser des boîtes génériques afin d'associer de multiples flux codés à une même boîte originale. L'interprétation d'une telle association dépend de la boîte qui doit être remplacée. Une étude plus approfondie de ce sujet apparaît en A.3.6.4.

Dans l'exemple simple de la Figure A.7, les boîtes génériques apparaissent seulement au niveau sommital du fichier, dans le segment de métadonnées 0. Comme déjà noté cependant, des génériques peuvent servir à remplacer une boîte quelconque, dans tout segment de métadonnées. Cela permet de décomposer des fichiers complexes de façon hiérarchique. En tant que tel, un même fichier original peut être encapsulé dans diverses structures de segment de métadonnées, selon la façon dont les génériques sont utilisés. **Cependant, un même flux JPP ou JPT ne doit adopter qu'une seule encapsulation de ce type.** Dans les applications client-serveur, le serveur distant va généralement déterminer une structure de segment de métadonnées appropriée au fichier, en attribuant un identificateur unique au flux résultant et en utilisant la même structure de segment de métadonnées dans toutes les communications avec tous les clients qui font référence à ce même identificateur unique.

Quand un générique relocalise une boîte dans un nouveau segment de métadonnées, l'en-tête de cette boîte (champs LBox, TBox et XBox) est mémorisé, sans modification, dans la boîte générique. Si un client ou agent de rendu a besoin d'appliquer des boîtes particulières selon leurs décalages de fichier originaux, il peut le faire au moyen des en-têtes de boîte originaux qui apparaissent dans les boîtes génériques. Ces informations permettent en fin de compte d'appliquer un quelconque emplacement dans le fichier original à un emplacement particulier dans un segment particulier de métadonnées, si le contenu de ce segment de données existe. Ce point est important car certains fichiers de la famille JPEG 2000 contiennent des boîtes qui font référence à d'autres boîtes selon leur emplacement dans le fichier.

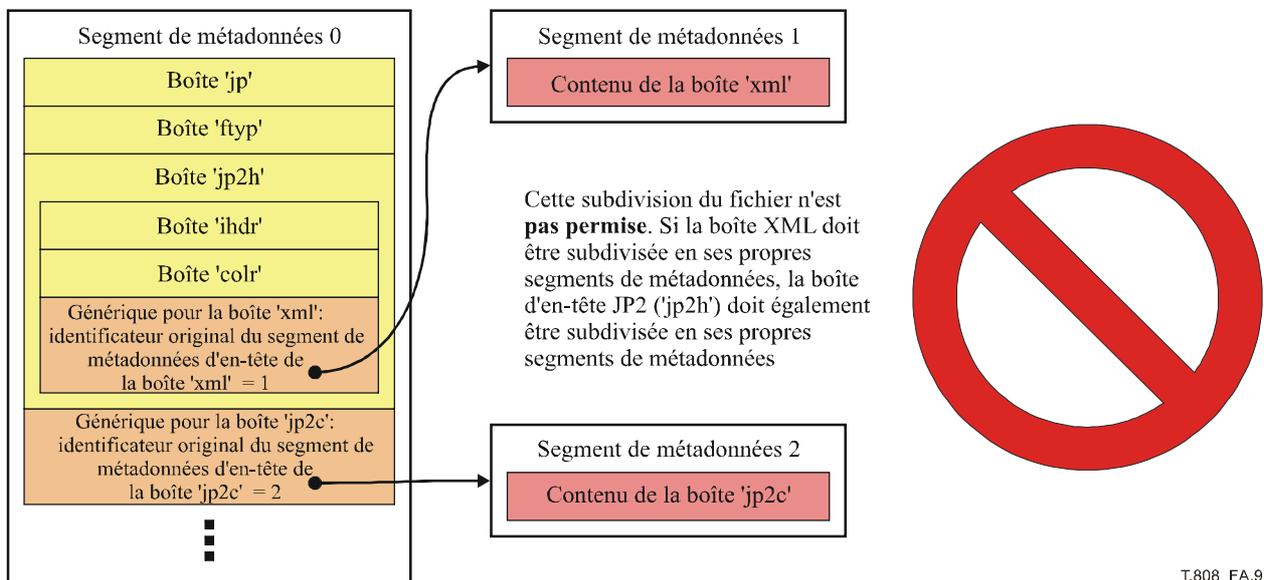
Alors qu'une liberté considérable existe quant à la meilleure façon de subdiviser un fichier en segments de métadonnées, il existe la restriction suivante: une boîte générique quelconque qui apparaît dans un segment de métadonnées doit remplacer une boîte de niveau sommital dans ce segment de données. De même, chaque fois qu'une sous-boîte doit être remplacée par un générique, sa superboîte immédiatement englobante doit résider dans son propre segment de métadonnées. Par exemple, dans l'échantillon de fichier représenté par la Figure A.6, les données XML contenues dans la boîte d'en-tête JP2 peuvent être placées dans un segment de données distinct des autres boîtes. Cela permet à un serveur distant de n'acheminer que les segments de données qui sont réellement requis pour le décodage et l'affichage de l'image, à moins que des données XML ne soient explicitement demandées. Une structure de segment de données appropriée est représentée dans la Figure A.8.



T.808_FA.8

Figure A.8 – Superboîte avec segment de métadonnées référencé

Il ne serait pas légal, cependant, que la boîte d'en-tête JP2 soit laissée dans le segment de métadonnées 0, comme représenté dans la Figure A.9.



T.808_FA.9

Figure A.9 – Subdivision illégale du fichier en segments de métadonnées

NOTE – Une façon équivalente d'exprimer cette même restriction est la suivante. Chaque fois qu'un générique remplace une sous-boîte, un générique doit également remplacer sa boîte englobante. Cette restriction garantit qu'il est toujours possible à un client ou à un agent de rendu de récupérer les longueurs et les emplacements des boîtes originales dans le fichier, même si certaines de ces boîtes ne sont pas comprises par ce client.

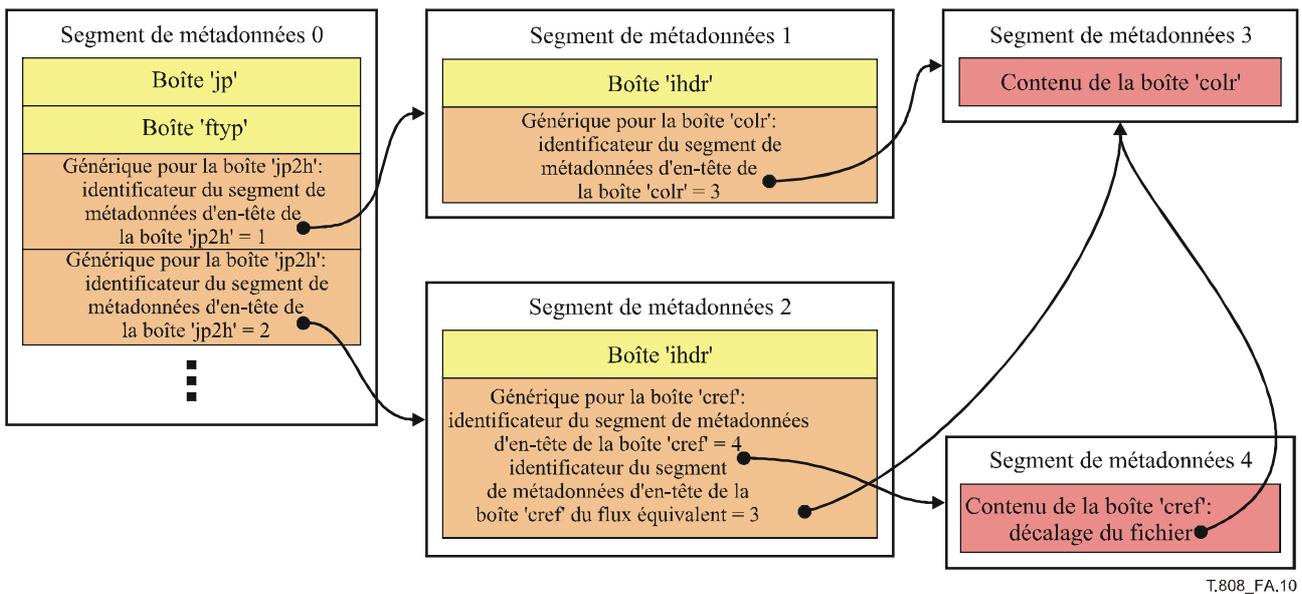
En plus de la fourniture du contenu original d'une boîte dans un segment distinct de métadonnées, les flux JPP et JPT sont également autorisés à offrir des représentations en variante de cette boîte, qui ne figuraient pas explicitement dans le fichier original. Ces représentations en variante sont appelées "flux équivalents". Par exemple, le fichier original pourrait contenir une boîte de renvoi interne dont la boîte de liste de fragments recueille un ou plusieurs fragments du fichier afin de reconstituer une boîte de spécification de l'espace chromatique. Alors qu'un client ou agent de rendu devrait être en mesure de suivre les pointeurs de fichier applicables afin de reconstruire la boîte de spécification de l'espace chromatique, une représentation plus pratique du flux JPP ou JPT pourrait contenir un générique qui se rapporte à un segment de données contenant la boîte de spécification de l'espace chromatique reconstruite sous forme de flux équivalent. A cette fin, le générique contient un en-tête de boîte pour le flux équivalent, ainsi que l'identificateur du segment de métadonnées qui détient le contenu de la boîte de flux équivalent.

L'exemple suivant (représenté dans la Figure A.10) décrit l'utilisation de flux équivalents pour des boîtes de renvoi interne. Dans ce cas, le segment de données qui contient le contenu du flux équivalent est également référencé comme englobant le contenu original d'une autre boîte. Bien que cela soit susceptible d'être une situation commune lorsque le fichier original contenait des boîtes de renvoi interne, il n'est pas nécessaire que le flux équivalent pointe sur un segment

de métadonnées qui est connecté à la hiérarchie de fichiers originale. Le contenu de la boîte de flux équivalent peut être créé à partir de zéro ou peut se rapporter à un contenu qui existait initialement dans d'autres fichiers. Cela permet d'encapsuler entièrement, dans un même flux JPP ou JPT, des boîtes de renvoi interne dont la liste de fragments fait référence à d'autres fichiers ou adresses URL.

Les flux équivalents peuvent être utilisés dans toute situation où le serveur distant peut créer une variante du contenu d'une boîte qui offre certains avantages au client; ils ne sont pas limités à fournir un accès à des données faisant l'objet d'un renvoi explicite.

En plus de son pointage sur des données de boîte réelles ou équivalentes, une boîte générique peut pointer sur un ou plusieurs flux codés où la boîte remplacée est équivalente à ces flux codés. Par exemple, la boîte de flux codé contigu peut être remplacée par une boîte générique qui se rapporte à l'identificateur du flux codé incrémentiel contenu dans cette boîte de flux codé contigu. Un autre exemple consisterait à remplacer la boîte de décalage de fragment contenue dans un fichier JPEG 2000 par un générique qui spécifie une série tabulaire d'identificateurs de flux codé. Ces identificateurs de flux codé se rapportent aux flux codés qui sont pointés par la boîte de décalage de fragment.

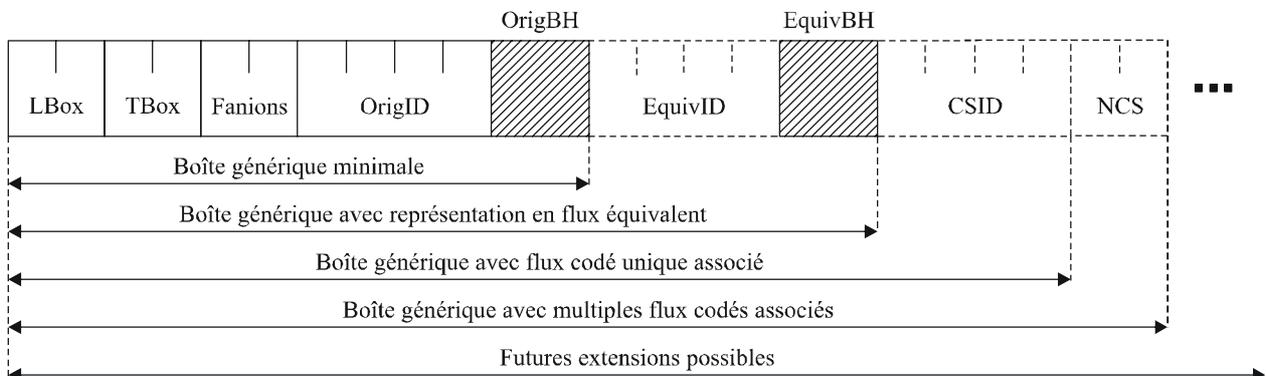


T.808_FA.10

Figure A.10 – Exemple de l'utilisation de flux équivalents

A.3.6.3 Format de boîte générique

La Figure A.11 montre le format d'une boîte générique, y compris l'en-tête de boîte (à la différence de la définition de la plupart des boîtes présentées dans l'Annexe I et dans d'autres parties de la présente Recommandation | Norme internationale). Le format est spécifié de cette façon afin de souligner le fait que l'utilisation du champ de longueur dans l'en-tête d'une boîte générique est plus restrictive que pour d'autres boîtes.



T.808_FA.11

Figure A.11 – Structure de boîte générique

ISO/CEI 15444-9:2005 (F)

LBox: champ de longueur gros-boutiste de 4 octets, normalisé pour une boîte. La valeur ne doit pas être 1 pour une boîte générique, c'est-à-dire que le champ XLBox ne doit pas être présent.

TBox: champ de type de boîte de 4 octets, normalisé pour une boîte. La valeur de type pour une boîte générique doit être 'phld' (0x7068 6c64).

Fanions: ce champ spécifie quels éléments de la boîte générique contiennent des données valides. Ce champ est codé comme un entier gros-boutiste de 4 octets. Les valeurs légales pour le champ de fanions sont spécifiées dans le Tableau A.3.

OrigID: ce champ spécifie l'identificateur du segment de métadonnées englobant le contenu de la boîte originale référencée par cette boîte générique. Il est codé comme un entier non signé gros-boutiste de 8 octets.

OrigBH: ce champ spécifie l'en-tête original (LBox, TBox et XLBox, selon les besoins) de la boîte originale référencée par cette boîte générique. La longueur de ce champ est de 8 octets si le champ LBox de l'en-tête de la boîte originale n'est pas égal à 1, et de 16 octets si ce n'est pas le cas.

EquivID: ce champ spécifie l'identificateur du segment de métadonnées qui contient une forme de flux équivalent au contenu de cette boîte. Ce champ est codé comme un entier non signé gros-boutiste de 8 octets.

EquivBH: ce champ spécifie l'en-tête de boîte de flux équivalent (LBox, TBox et XLBox selon les besoins) de la boîte référencée par cette boîte générique. La longueur de ce champ est de 8 octets si le champ LBox de l'en-tête de la boîte de flux équivalent n'est pas égal à 1, et de 16 octets sinon.

CSID: ce champ spécifie l'identificateur du premier flux codé associé à la boîte remplacée. C'est l'identificateur qui est associé à tous les segments de données d'en-tête, de district et/ou de pavé servant à communiquer progressivement le contenu du premier flux codé associé à la boîte remplacée. Ce champ est codé comme un entier non signé gros-boutiste de 8 octets.

NCS: ce champ spécifie le nombre de flux codés contenus dans la série tabulaire de flux codés qui est équivalente à la boîte remplacée. Les valeurs d'identificateur de ces flux codés s'étendent consécutivement à partir de la valeur spécifiée par le champ CSID. Ce champ est codé comme un entier non signé gros-boutiste de 4 octets.

ExtendedBoxList: ce champ n'est pas spécifiquement représenté dans la Figure A.11. Le champ NCS peut être suivi par une séquence de boîtes contenant des informations étendues provenant du serveur distant. L'existence d'une boîte quelconque suivant le champ NCS doit être spécifiée au moyen d'un bit dans le champ de fanions. Cependant, aucune boîte étendue ni aucun fanion binaire additionnel n'est défini par la présente Recommandation | Norme internationale. Les clients doivent ignorer toute boîte contenue dans le champ ExtendedBoxList qui n'est pas comprise.

Une valeur binaire de "x" dans le Tableau A.3 indique que la valeur spécifiée inclut des cas où ce bit est réglé à "1" ou à "0". Les bits indiqués par la lettre "y" ne sont pas utilisés par la présente Recommandation | Norme internationale: ils doivent être réglés à 0 par les serveurs distants et être ignorés par les clients.

Il n'est pas nécessaire que la totalité des champs définis pour une boîte générique apparaissent dans chaque boîte générique. Comme suggéré par les flèches dans la Figure A.11, si aucun identificateur de flux codé équivalent ou incrémentiel de boîte n'est fourni, la boîte peut être terminée à la fin du champ OrigBH. De même, si aucun identificateur de flux codé incrémentiel n'est fourni, la boîte peut être terminée à la fin du champ EquivBH et si au plus un seul identificateur de flux codé incrémentiel est fourni, la boîte peut être terminée à la fin du champ CSID.

Tableau A.3 – Valeurs légales pour le champ de fanions "Flags" d'une boîte générique

Valeur	Signification
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xxx1	L'accès au contenu original de cette boîte est fourni au moyen du segment de métadonnées spécifié dans le champ OrigID.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xxx0	Aucun accès au contenu original de cette boîte n'est fourni et la valeur du champ OrigID doit être ignorée.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xx1x	Une boîte de flux équivalent est fournie, dont le contenu se trouve dans le segment de métadonnées spécifié par le champ EquivID.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xx0x	Aucune boîte de flux équivalent n'est fournie et la valeur de chaque champ EquivID ou EquivBH doit être ignorée.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy 01xx	L'accès à l'image représentée par cette boîte est fourni par un même flux codé incrémentiel, qui est identifié par le champ CSID. La valeur du champ NCS doit être traitée comme si elle avait été réglée à "1" quelle que soit la valeur réelle de ce champ.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy 11xx	L'accès à l'image représentée par cette boîte est fourni par un ou plusieurs flux codés incrémentiels, comme spécifié par les champs CSID et NCS.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy x0xx	Ce générique ne donne pas accès à une image représentant la boîte originale en tant que flux codé incrémentiel; les champs CSID et NCS doivent être ignorés.
Autres valeurs	Champ réservé pour utilisation par l'ISO.

A.3.6.4 Référencement de flux codés incrémentiels avec des génériques

Chaque fois que des segments de données d'en-tête, de district ou de pavé existent, leur identificateur de flux codé doit apparaître dans un segment de métadonnées approprié, situé dans une boîte générique. La seule exception à cette exigence concerne les flux à codage JPEG 2000 sans format enroulé, qui ne sont pas imbriqués dans un format de fichier de la famille JPEG 2000.

Les valeurs d'identificateur de flux codé qui apparaissent dans la boîte générique correspondante doivent être conformes à toutes les exigences imposées par le format de fichier englobant. Par exemple, les fichiers JPX attribuent formellement un numéro séquentiel à chaque flux codé qui apparaît au niveau sommital du fichier, soit au moyen d'une boîte de flux codé contigu ou au moyen d'une boîte de table de fragments. Le premier flux codé de niveau sommital contenu dans la cible logique doit avoir un identificateur de flux codé égal à 0; le prochain flux doit avoir un identificateur de flux codé égal à 1; et ainsi de suite.

Les génériques qui font référence à de multiples identificateurs de flux codé ne peuvent être utilisés que lorsque la signification de ces flux codés est bien définie par le type de la boîte qui doit être remplacée.

A.3.6.5 Utilisation de boîtes génériques avec les fichiers MJ2

La présente Recommandation | Norme internationale ne définit que deux types de boîtes appropriées aux génériques avec fichiers de type *Motion JPEG 2000* (MJ2). Spécifiquement, soit la boîte de décalage de fragment ('stco') ou la boîte de décalage de long fragment ('co64') peut être remplacée par une boîte générique qui désigne de multiples identificateurs de flux codé.

Chaque piste vidéo contenue dans un fichier MJ2 contient exactement 1 boîte de décalage de fragment (soit 'stco' ou 'co64') qui, en combinaison avec la boîte d'échantillons jusqu'au fragment ('stsc'), sert à identifier les emplacements de toutes les boîtes de flux codé contigu qui appartiennent à cette piste vidéo. Si la boîte de décalage de fragment est remplacée par un générique qui offre un ou plusieurs identificateurs de flux codé, il doit y avoir exactement 1 identificateur de flux codé pour chaque boîte de flux codé contigu dans la piste vidéo. Si la boîte d'entrée d'échantillon visuel ('mjp2') indique un décompte de champs égal à 2, il doit y avoir $2N$ identificateurs de flux codé dans l'étendue fournie par la boîte générique, où N est le nombre d'échantillons vidéo (c'est-à-dire que N est le nombre de trames). Sinon, il doit y avoir seulement N identificateurs de flux codé dans l'étendue fournie par la boîte générique. Les identificateurs de flux codé doivent être ordonnés en séquence par numéro d'échantillon (numéro de trame) et par numéro de champ dans chaque échantillon.

NOTE – Pour les fichiers MJ2 contenus dans une représentation par flux JPP ou JPT, il n'y a aucune nécessité que le flux achemine le contenu de la boîte originale de décalage de fragment, de la boîte d'échantillons jusqu'au fragment ('stsc'), ou de la boîte de longueur d'échantillon ('stsz'). Ces informations d'indexation peuvent être régénérées au besoin si la représentation du flux est convertie en fichier MJ2.

A.4 Conventions relatives à l'analyse sémantique et à la remise de flux JPP et JPT (pour information)

Les boîtes génériques apportent, aussi bien aux clients qu'aux serveurs, un complément de flexibilité et une certaine probabilité d'ambiguïté dans la façon dont ils analysent ou remettent des flux JPP et JPT. Un serveur distant peut choisir de partitionner des boîtes originales à partir d'un fichier de la famille JPEG 2000 afin d'obtenir des segments de métadonnées utilisant l'une quelconque d'une large étendue de stratégies, en introduisant des boîtes génériques à des points appropriés. Le serveur distant doit faire cela de façon cohérente, afin que les segments de données associés à un flux JPP ou JPT aient le même contenu nominal pour tous les clients qui accèdent à la même cible logique (éventuellement qualifiée par un identificateur unique de cible), chaque fois qu'ils y accèdent.

Ce qui est cependant plus important, est que les boîtes génériques permettent aux serveurs distants de construire un même flux JPP ou JPT dont les segments de données offrent de multiples représentations en variante du même contenu original. Cela peut se produire quand un écoulement de flux équivalent est identifié dans un générique, et/ou quand un identificateur de flux codé incrémentiel est identifié dans un générique. Dans ces cas, une boîte originale pourrait être rendue disponible dans un segment de métadonnées, tout en étant également rendue disponible en tant que flux équivalent dans un autre segment de métadonnées, et/ou également en étant rendue disponible en tant que flux codé incrémentiel au moyen des segments de données d'en-tête, de district ou de pavé. Alors qu'ils pourraient distribuer le contenu de tous les segments de données qui représentent une boîte originale, les serveurs distants seront, pour des causes d'efficacité, censés distribuer seulement des informations suffisantes pour acheminer le contenu original, à moins qu'ils ne soient explicitement invités à distribuer des segments de données redondants. Quand ils sont confrontés à des représentations multiples d'une boîte originale, les analyseurs de flux JPP ou JPT du côté client pourraient choisir d'ignorer la totalité des représentations sauf une seule. La convention prévue avec le client devrait avoir une influence importante sur le choix du segment de métadonnées que le serveur distant choisit d'envoyer réellement à un client.

En considération de ce qui précède, la présente Recommandation | Norme internationale recommande les conventions suivantes:

- Sauf argument contraire, un serveur doit partir du principe que l'analyseur du client va analyser une boîte de flux équivalent, de préférence à la boîte originale, si la présence des deux types de boîte a été signalée au client par des génériques.
- Sauf argument contraire, un serveur doit partir du principe que l'analyseur du client va utiliser la représentation par flux codé incrémentiel (segments de données d'en-tête, de district ou de pavé) de préférence à un flux codé non traité si la présence des deux types de boîte a été signalée au client par des génériques.

A.5 Conventions relatives à l'interopérabilité avec des flux JPP ou JPT (pour information)

Cette convention décrit le format de fichier d'échange pour flux JPP ou JPT, ci-après appelés respectivement *jpp-file* et *jpt-file*. Un tel fichier peut contenir les données JPEG 2000 reçues à partir d'une session JPIP (le cache du client par exemple), ou un sous-ensemble de ces données. Il est possible à un autre client du protocole JPIP de lire et d'utiliser ce fichier parce que les flux JPP et JPT sont des types de média autodescriptifs.

Ces fichiers sont formés par concaténation de messages de flux JPT ou JPP. Ils peuvent par exemple être formés par la simple concaténation de tous les messages de ce type qui ont été reçus par un client dans une même session ou à partir de multiples sessions. Une situation améliorée serait celle où des clients auraient produit un flux JPT ou JPP légal au moyen d'un même en-tête de message et d'un même message par segment de données.

Il est recommandé que les extensions ".jpp" et ".jpt" soient utilisées pour ces fichiers et, au besoin, que le nom du fichier contienne une référence à un jeton applicable de cible ou d'identificateur de cible JPIP.

Cette convention ne spécifie pas l'implémentation ni la structure du cache pour un client. Par exemple, un client peut utiliser une base de données afin d'implémenter la fonction de cache plutôt qu'un système de cache fondé sur des fichiers.

Annexe B

Sessions, canaux, modèle de cache et ensembles de modèles

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

B.1 Requêtes émises au cours d'une session par opposition aux requêtes sans description d'état

Le protocole JPIP fait une claire distinction entre deux types de requête différents: les requêtes sans description d'état et celles qui appartiennent à une session.

L'objet des sessions consiste à réduire la quantité de communications explicites qui sont requises entre le client et le serveur. Dans une session, le serveur est censé mémoriser les capacités et les préférences du client, fournies dans de précédentes requêtes, de façon que ces informations n'aient pas besoin d'être envoyées dans toutes les requêtes. Plus important encore, le serveur distant conservera normalement un journal des informations qu'il a déjà envoyées au client en réponse à de précédentes requêtes, de façon que ces informations n'aient pas besoin d'être retransmises en réponse à de futures requêtes. Ce journal persistera pendant la durée d'une session. Sauf instruction explicitement contraire, le serveur distant peut partir du principe que le client antémémorise les réponses à toutes les requêtes émises dans une session. Le serveur peut également modéliser sa mémoire cache en n'envoyant que les portions des données ou métadonnées d'image comprimée que le client ne détient pas encore dans son cache.

Les requêtes sans description d'état ne sont associées à aucune session et doivent donc être entièrement autonomes. Il convient de noter que le terme "sans description d'état" s'applique seulement au serveur distant et non au client. Comme pour les sessions, le client devrait généralement mettre en mémoire cache les réponses à de précédentes requêtes associées à la même cible logique. Les clients qui émettent de multiples requêtes sans description d'état pour la même cible devraient généralement inclure dans chaque requête des informations sur le contenu de leur cache, de façon à éviter la transmission de données redondantes. Ainsi, les avantages apportés par les sessions sont des requêtes plus petites, moins complexes et/ou moins de données redondantes dans les réponses du serveur distant. L'avantage des communications sans description d'état est que le serveur distant n'a pas besoin de conserver les informations d'état entre les requêtes; autrement dit, le même serveur local n'a pas besoin, en fin de compte, de desservir toutes les requêtes visant une même image cible provenant d'un même client.

B.2 Canaux et sessions

Les éléments suivants sont associés à chaque session:

- Une ou plusieurs cibles logiques (habituellement des fichiers d'image), dont le contenu ne change pas au cours de la session.
- Un même type de retour de données d'image pour chaque cible logique associée à la session.
- Pour chaque cible logique associée à la session, un modèle du contenu du cache client doit être conservé chaque fois que le type de retour des données est soit "flux JPP" ou "flux JPT". Noter cependant que ce modèle n'a pas besoin de parfaitement refléter l'état réel du cache du client. Les règles régissant la maintenance des modèles de mémoire cache sont décrites en B.3.
- Un ou plusieurs canaux JPIP. Les clients peuvent généralement ouvrir de multiples canaux au cours de la même session. Chaque canal JPIP peut être associé à un canal distinct de transport sous-jacent (par exemple une connexion TCP distincte), bien que ce cas ne soit pas probable. De multiples canaux permettent aux clients d'émettre des requêtes simultanées pour de multiples régions d'image, dans l'hypothèse que le serveur distant satisfera ces requêtes de concert. Les canaux permettent également une attribution intelligente de la largeur de bande entre différents types de requêtes, soit dans une même image cible ou dans des cibles multiples.
- Lorsque des canaux multiples sont associés à la même cible logique, le modèle de cache de la session s'applique à tous les canaux. De multiples clients peuvent ouvrir des canaux JPIP à l'intérieur de la même session, bien que cela puisse avoir des effets secondaires indésirables si ces canaux se rapportent à la même cible logique.

Les éléments suivants sont associés à chaque canal:

- Une même cible logique (habituellement un fichier d'image).
- Un identificateur attribué par le serveur distant qui doit être inclus dans chaque requête. Le protocole JPIP ne définit pas d'identificateur de session distinct, car l'identificateur de canal est suffisant pour associer la requête à sa session.
- Un enregistrement des capacités et préférences du client, qui peut être ajusté au moyen de champs de requête appropriés.

- Dans la mesure où le serveur distant met en file d'attente les requêtes, il devrait offrir une file d'attente distincte pour chaque canal JPIP.

Il y a une correspondance univoque entre requête et réponse du client dans un canal. Différents canaux JPIP peuvent être sur le même canal de transport ou sur différents canaux de transport. Les requêtes qui utilisent différents canaux JPIP peuvent arriver de façon asynchrone au serveur distant si des canaux de transport distincts servent à transporter les requêtes. Les réponses qui utilisent différents canaux JPIP peuvent arriver de façon asynchrone au client si des canaux de transport distincts servent à transporter les réponses. La prise en charge de canaux multiples est à la discrétion du serveur distant. Cependant, le champ de requête de vitesse d'acheminement et les préférences relatives à la largeur de bande maximale et à la tranche spectrale devraient guider le serveur distant.

B.3 Gestion du modèle de cache

Comme déjà noté, une des principales fonctions d'une session est la modélisation du cache du client par le côté serveur distant. Sauf indication explicitement contraire, le serveur distant **peut** partir du principe que le client a antémémorisé toutes les informations envoyées en réponse à des requêtes au cours de la session: ces informations **n'ont pas besoin** d'être retransmises. Noter cependant que le serveur distant n'est pas obligé de conserver un modèle de cache complet ni même un quelconque modèle de cache car des données redondantes **peuvent** être transmises en réponse à des requêtes.

En plus de l'impact des données transmises, des instructions explicites de manipulation du modèle de cache, contenues dans les requêtes émises par un client, peuvent mettre à jour le modèle de cache du serveur distant. Ces instructions doivent être traitées avant de déterminer les données qui devraient être renvoyées au client en réponse à sa requête. Il y a deux types d'instructions de manipulation du modèle de cache: les instructions additives et les instructions soustractives.

Les instructions additives de manipulation du modèle de cache servent à augmenter le modèle de cache du serveur distant, en ajoutant au modèle existant des segments de données ou des portions de segments de données. Ces instructions permettent à un client de signaler au serveur distant des informations qu'il a reçues lors d'une session précédente, ou d'utiliser de précédentes requêtes sans description d'état. Un serveur distant **devrait** essayer d'exploiter toute instruction additive de manipulation du modèle de cache qui pourrait apparaître dans les requêtes émises par un client. Cependant, les serveurs distants ne sont pas obligés de conserver un modèle complet du cache, de sorte qu'un serveur distant peut négliger, totalement ou partiellement, les instructions additives de manipulation du modèle de cache.

Les instructions soustractives servent à supprimer des segments de données, ou des portions de segments de données à partir du modèle de cache du serveur distant. Un client pourrait émettre des instructions soustractives de manipulation du modèle de cache afin d'informer le serveur distant qu'il n'a pas antémémorisé ou a rejeté certaines données qui ont été déjà envoyées par le serveur distant. Celui-ci est par ailleurs libre de partir du principe que le client a antémémorisé toutes les données transmises pendant la session. Le serveur distant **doit** supprimer, de tout modèle de cache (complet ou non) qu'il tient à jour, toutes les informations identifiées par une instruction soustractive de manipulation du modèle de cache.

Les requêtes JPIP en mode session ont des effets secondaires qui peuvent affecter la réponse à de futures requêtes. Il en est de même des requêtes qui contiennent des instructions de manipulation du modèle de cache – car les effets de la manipulation du modèle de cache sont rémanents. En outre, les effets secondaires d'une requête arrivant dans un canal JPIP sont répercutés dans la réponse à d'éventuelles requêtes qui pourraient appartenir à un autre canal JPIP associé à la même cible logique. Cela vient du fait qu'il n'y a qu'un seul modèle de cache pour chaque cible logique dans une session.

B.4 Interrogation et manipulation des ensembles de modèles

Lorsqu'une cible logique associée à une session contient un grand nombre de flux codés (par exemple une cible vidéo), ou qu'un client reste longtemps connecté, la modélisation partielle du cache devient une stratégie de plus en plus probable pour les réalisations concrètes de serveur distant. Il devient également de plus en plus probable que les clients seront dans l'impossibilité de mettre en mémoire cache toutes les informations envoyées par le serveur distant. Afin d'éviter des insuffisances de communication dans de telles circonstances, le concept de champ "mset" (ensemble de modèles) est introduit. "mset" est l'ensemble des flux codés pour lesquels le contenu du cache client doit être modélisé par le serveur distant.

Dans une requête quelconque, le client peut donner l'ordre au serveur distant de limiter son champ "mset" à un ensemble particulier de flux codés. Cela permet pratiquement aux clients de rejeter des flux codés entiers à partir de leur cache sans courir le risque que le serveur distant produise des réponses incomplètes à de futures requêtes concernant ces flux codés.

Les requêtes "mset" se traduisent également par des réponses de serveur qui indiquent l'ensemble réel de flux codés pour lequel des informations de modèle de cache doivent être conservées. Cela permet aux clients de déterminer si des

instructions de manipulation du modèle de cache, qui se rapportent à une variété de flux codés, seront ou non négligées par le serveur distant.

En l'absence de toute manipulation ou interrogation explicite d'un ensemble "mset", le client ne peut que partir du principe que l'ensemble "mset" du serveur distant contient tous les flux codés pour lesquels des données de réponse à sa requête seront produites. Comme les serveurs distants ont généralement le droit de limiter le domaine d'application d'une requête du client à un plus petit nombre de flux codés que celui qui a été originellement spécifié, il n'y a aucune garantie que l'ensemble "mset" du serveur distant inclura tous les flux codés mentionnés dans une requête, à moins que celle-ci n'ait mentionné qu'un seul flux codé. Ces questions sont expliquées plus en détail en C.8.6.

Annexe C

Requête émise par un client

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

C.1 Syntaxe des requêtes

C.1.1 Introduction

Cette annexe décrit tous les éléments pouvant être contenus dans une requête JPIP. Chaque paragraphe principal décrit un groupe de champs et de valeurs possibles pour ces champs. En général, une requête se composera de champs extraits de plusieurs groupes, mais certains groupes sont incompatibles. D'autre part, dans chaque groupe, certains champs de requête sont incompatibles. Certaines requêtes légales par ailleurs peuvent ne pas être valides lors d'une utilisation dans certaines situations (par exemple sessions), bien que cela ne soit pas indiqué par la syntaxe du formalisme BNF. Finalement, même avec une requête légale, un serveur distant ne peut pas prendre en charge tous les champs de requête possibles ou leurs combinaisons.

C.1.2 Structure des requêtes

La requête JPIP se compose des champs suivants:

- Champ d'identification de cible
- Champs de gestion de session et de canal
- Champs de requête de fenêtre de visualisation
- Champ de métadonnées
- Champs de requête de limitation de données
- Champs de requête de commande de serveur distant
- Champs de requête de gestion de cache
- Champs de requête de téléchargement amont
- Champs de capacité et de préférence du client

Les éléments contenus dans la requête doivent être envoyés conformément au protocole de transport choisi. En protocole HTTP par exemple, les requêtes sont exprimées par les caractères énumérés dans la syntaxe du formalisme BNF, les paramètres multiples sont accolés au moyen d'un caractère "&" et les requêtes peuvent faire partie du champ d'interrogation d'une requête GET, ou du corps d'une requête POST. Voir les Annexes F, G et H pour de plus amples détails.

NOTE – Les caractères réservés d'un identificateur URI peuvent faire l'objet d'un échappement. Par exemple, "request=a:b" dans une adresse URL de commande GET en protocole HTTP se traduira par "request=a%3Ab", où le caractère réservé ':' des adresses URL fait l'objet d'un échappement vers le groupe '%3A'.

```

jpip-request-field = target-field
                    / channel-field
                    / view-window-field
                    / metadata-field
                    / data-limit-field
                    / server-control-field
                    / cache-management-field
                    / upload-field
                    / client-cap-pref-field

target-field       = target           ; C.2.2
                   / subtarget       ; C.2.3
                   / tid              ; C.2.4

channel-field      = cid              ; C.3.2
                   / cnew            ; C.3.3
                   / cclose          ; C.3.4
                   / qid             ; C.3.5

```

view-window-field	= fsiz	; C.4.2
	/ roff	; C.4.3
	/ rsiz	; C.4.4
	/ comps	; C.4.5
	/ stream	; C.4.6
	/ context	; C.4.7
	/ srates	; C.4.8
	/ roi	; C.4.9
	/ layers	; C.4.10
metadata-field	= metareq	; C.5.2
data-limit-field	= len	; C.6.1
	/ quality	; C.6.2
server-control-field	= align	; C.7.1
	/ wait	; C.7.2
	/ type	; C.7.3
	/ drates	; C.7.4
cache-management-field	= model	; C.8.1
	/ tpmodel	; C.8.3
	/ need	; C.8.4
	/ tpneed	; C.8.5
	/ mset	; C.8.6
upload-field	= upload	; C.9.1
client-cap-pref-field	= cap	; C.10.1
	/ pref	; C.10.2
	/ csf	; C.10.3

C.1.3 Restrictions relatives à la combinaison des champs de requête

Chaque type du champ de requête JPIP doit apparaître au plus une fois dans une requête isolée.

En général, les requêtes visant des données d'image (requêtes de fenêtre de visualisation) peuvent être combinées avec des requêtes visant des métadonnées additionnelles. Cependant, il y a des restrictions sur la façon dont les champs de requête peuvent être combinés.

Le champ de requête de téléchargement amont ne doit pas être combiné avec les formes `metadata-field`, `data-limit-field`, ou `server-control-field`.

C.2 Champs d'identification de cible

C.2.1 Introduction aux cibles logiques

Chaque requête JPIP est orientée vers une représentation spécifique d'une certaine ressource nommée originale ou vers une certaine portion de cette ressource. Celle-ci peut être un fichier ou un objet physiquement stocké, ou quelque chose qui est créé virtuellement par le serveur distant sur demande.

La représentation spécifique, qu'il s'agisse de la forme codée originale, d'une forme transcodée, d'une certaine série d'octets ou de la ressource entière, est désignée par le terme de *cible logique*. Celle-ci est spécifiée au moyen de trois champs de requête: Identificateur de cible, Cible et Sous-Cible.

Le champ de requête Cible spécifie la ressource nommée originale vers laquelle la requête est dirigée. Il est spécifié au moyen d'une forme `chemin`, qui pourrait être une simple chaîne ou un identificateur URI. Si le champ Cible n'est pas spécifié et que la requête soit transportée par le protocole HTTP, alors la requête JPIP doit être dirigée vers la ressource spécifiée au moyen du composant de chemin de l'adresse URL de cette requête JPIP. Cette ressource nommée originale peut être un fichier réel ou un autre objet physiquement mémorisé dans le serveur distant; ou bien elle peut être quelque chose que le serveur distant crée en réponse à une requête JPIP.

Le champ de requête Sous-Cible spécifie la série spécifique d'octets de la ressource nommée originale (spécifiée au moyen du champ de requête Cible) vers laquelle la requête est dirigée. Si le champ de requête Sous-Cible n'est pas spécifié, la requête est dirigée vers la série entière d'octets de la ressource originale.

Le champ de requête d'Identificateur de cible peut servir à spécifier plus précisément un codage particulier de la ressource lorsque le client et le serveur distant ont déjà échangé des données à partir de cette ressource. Par exemple, le serveur distant peut avoir déjà fourni au client une version transcodée du fichier sur la base des informations fournies et

ISO/CEI 15444-9:2005 (F)

des conditions propres à une requête précédente. Si ce client a préservé les données déjà transmises dans son cache, il souhaitera continuer à recevoir des données au moyen du même transcodage de façon qu'il puisse continuer à utiliser les données contenues dans le cache. L'identificateur de cible est une chaîne d'identification définie par le serveur distant, à laquelle celui-ci a déjà associé cette représentation spécifique de cette même ressource nommée originale, ou a associé une série d'octets d'une certaine ressource nommée, originale et particulière.

Si un client spécifie à la fois la ressource nommée originale (soit au moyen du champ de requête Cible ou au moyen du composant de chemin de l'adresse URL de la requête JPIP) et l'identificateur de cible, le serveur distant doit vérifier s'il peut ou non répondre à cette requête de la même façon que quand il avait originellement assigné cet identificateur de cible à cette ressource. Si le serveur distant ne peut pas répondre de la même façon, il doit utiliser un en-tête de réponse de type JPIP-tid afin d'informer le client d'un nouvel identificateur de cible, à partir de quoi le client saura qu'il doit impérativement rejeter toutes les données déjà mises en mémoire cache.

Si une cible logique doit être desservie par des messages à flux JPP ou JPT, les segments de données associés doivent rester cohérents dans toutes les réponses qui sont émises au cours de la même session. Lorsque le serveur distant, ou un serveur distant associé, émet également un identificateur de cible, les segments de données doivent rester cohérents dans toutes les réponses émises avec le même identificateur de cible, qu'elles soient ou non émises au cours de la même session.

Si cette requête fait partie d'une session et qu'un identificateur de canal ait été assigné par le serveur distant, le client peut spécifier cet identificateur de canal au moyen du champ de requête d'identificateur de canal au lieu de spécifier le champ de Cible, de Sous-Cible ou d'Identificateur de cible. Si la cible logique est spécifiée à la fois au moyen d'une combinaison des champs de Cible, de Sous-Cible et d'Identificateur de cible et au moyen du champ de requête d'identificateur de canal, alors le serveur distant doit répondre par une erreur.

Les exemples suivants montrent la spécification des cibles logiques:

EXEMPLE 1: pour l'adresse URL de requête JPIP ci-après,

`http://one.jpeg.org/imageserver.cgi?target= http%3A%2F%2Fone.jpeg.org%2Fimages%2Fpicture.jp2&fsiz=200,200`, la cible logique est la série entière d'octets contenue dans l'identificateur URI `http://one.jpeg.org/images/picture.jp2`, relatif au répertoire du document racine dans le serveur distant.

EXEMPLE 2: pour l'adresse URL de requête JPIP ci-après,

`http://one.jpeg.org/imageserver.cgi? target= http%3A%2F%2Fone.jpeg.org%2Fimages%2Fpicture.jp2&tid=4384-5849-af4d-3dca&fsiz=200,200`, la cible logique est la série entière d'octets contenue dans l'identificateur URI `http://one.jpeg.org/images/picture.jp2` relatif au répertoire du document racine dans le serveur distant, avec une représentation spécifiée par l'Identificateur de cible défini par le serveur distant: 4384-5849-af4d-3dca.

EXEMPLE 3: pour l'adresse URL de requête JPIP ci-après,

`http://one.jpeg.org/imageserver.cgi?target= http%3A%2F%2Fone.jpeg.org%2Fimages%2Fpicture.jp2&subtarget=1038-13458&fsiz=200,200`, la cible logique est l'étendue d'octets commençant par l'octet 1038 et tous les octets jusqu'à et y compris l'octet 13458, contenue dans l'identificateur URI "`http://one.jpeg.org/images/picture.jp2`" relatif au répertoire du document racine dans le serveur distant.

EXEMPLE 4: pour l'adresse URL de requête JPIP ci-après, `http://one.jpeg.org/imageserver.cgi?cid=1234-5849-af4d-3dca&fsiz=200,200`, la cible logique est la ressource à laquelle le serveur distant s'est associé avec le canal ayant l'identificateur 1234-5849-af4d-3dca.

EXEMPLE 5: pour l'adresse URL de requête JPIP ci-après, `http://one.jpeg.org/images/picture.jp2?fsiz=200,200`, la cible logique est la série entière d'octets contenue dans le fichier "`images/picture.jp2`", relatif au répertoire du document racine dans le serveur distant.

EXEMPLE 6: pour l'adresse URL de requête JPIP ci-après, `http://one.jpeg.org/images/picture.jp2?subtarget=1038-13458&fsiz=200,200`, la cible logique est l'étendue d'octets commençant par l'octet 1038 et tous les octets jusqu'à et y compris l'octet 13458, contenue dans le fichier "`images/picture.jp2`" relatif au répertoire du document racine dans le serveur distant.

C.2.2 Cible ("target")

```
target = "target" "=" PATH
```

Ce champ sert à spécifier la ressource nommée originale (souvent le nom d'un fichier dans le serveur distant). Si le champ de requête Cible fait défaut, alors la ressource nommée originale est déterminée autrement.

C.2.3 Sous-Cible ("subtarget")

```
subtarget = "subtarget" "=" byte-range
byte-range = UINT-RANGE
```

Ce champ peut servir à qualifier la ressource nommée originale au moyen de la spécification d'une série d'octets. La cible logique doit être interprétée comme la série d'octets indiquée de la ressource nommée originale.

Les limites inférieure et supérieure de l'étendue d'octets fournie sont incluses et 0 se rapporte au premier octet du fichier de cible.

C.2.4 Identificateur de cible (tid)

```
tid = "tid" "=" target-id
target-id = TOKEN
```

Ce champ peut servir à fournir une chaîne "target-id" qui a déjà été produite par le serveur distant afin d'identifier formellement la cible logique qui doit faire l'objet d'un accès, y compris tout transcodage discrétionnaire effectué par le serveur distant. Le nom de la cible logique n'est pas nécessairement unique et ne correspond pas nécessairement à un seul codage de son contenu, tandis que la chaîne "target-id", associée au nom de la ressource originale et à la série d'octets, devrait identifier formellement à la fois l'imagerie et son codage.

Si la valeur de la chaîne "target-id" est "0", la cible logique est spécifiée au moyen du composant de chemin de Cible, de Sous-Cible et d'adresse URL du protocole JPIP et le client va demander explicitement que le serveur distant l'informe de l'identificateur "target-id" assigné, s'il en existe un. Le serveur distant doit inclure un en-tête identificateur de cible dans sa réponse à toutes les requêtes émises par un client avec une chaîne "target-id" de valeur égale à "0".

La longueur de la chaîne "target-id" ne doit pas dépasser 255 caractères.

C.3 Champs de travail avec des sessions et des canaux

C.3.1 Introduction

Une requête doit être sans description d'état à moins qu'une des conditions suivantes, ou les deux, n'apparaissent:

- La requête contient un champ d'identificateur de canal valide.
- La requête contient un champ de nouveau canal (voir ci-dessous) et la réponse du serveur distant contient un en-tête de réponse de nouveau canal avec un identificateur "channel-id" nouvellement émis.

Voir B.2, qui donne plus de détails sur les sessions et les canaux.

C.3.2 Identificateur de canal (cid)

```
cid = "cid" "=" channel-id
channel-id = TOKEN
```

- Ce champ sert à associer la requête à un certain canal JPIP et donc à la session à laquelle ce canal appartient.

C.3.3 Nouveau canal (cnew)

```
cnew = "cnew" "=" 1#transport-name
transport-name = TOKEN
```

Ce champ sert à demander un nouveau canal JPIP. Si aucun champ de requête d'identificateur de canal n'est présent, la requête vise une nouvelle session. Sinon, la requête vise un nouveau canal au cours de la même session en tant que canal identifié par le champ de requête d'identificateur de canal.

La chaîne de valeur désigne les noms d'un ou de plusieurs protocoles de transport que le client est disposé à accepter. La présente Recommandation | Norme internationale ne définit que les noms des protocoles de transport, "http" et "http-tcp", bien qu'il soit prévu que d'autres protocoles de transport, comme "udp", puissent être définis ailleurs. Les détails relatifs à l'utilisation du protocole JPIP au-dessus du protocole de transport "http" apparaissent dans l'Annexe F, alors que les détails de l'utilisation du protocole JPIP au-dessus du protocole de transport "http-tcp" apparaissent dans l'Annexe G.

ISO/CEI 15444-9:2005 (F)

Si le serveur distant est disposé à ouvrir un nouveau canal au moyen d'un des protocoles de transport indiqués, il doit renvoyer le jeton d'identificateur de nouveau canal au moyen de l'en-tête de réponse de nouveau canal (voir D.2.3). Dans ce cas, la requête actuelle est la première dans le nouveau canal.

Un client peut ouvrir un canal vers une nouvelle cible logique au cours de la même session. A cette fin, la requête du client doit identifier aussi bien un identificateur existant de canal qu'une cible logique. Lors de l'ouverture d'un nouveau canal vers la cible logique qui est associée à un canal existant, il n'est pas nécessaire de spécifier la cible logique explicitement.

Si le serveur distant n'est pas disposé à ouvrir de nouveau canal, il ne doit pas renvoyer d'en-tête de réponse de nouveau canal, mais la requête doit être desservie comme si le champ de requête de nouveau canal n'avait pas été inclus. C'est-à-dire qu'une requête qui spécifie un identificateur existant de canal doit être traitée comme une requête émise à l'intérieur de ce canal, alors qu'une requête qui ne contient aucun champ de requête d'identificateur de canal doit être traitée comme une requête sans description d'état. Si la requête de nouveau canal désigne une autre cible logique que celle qui est associée à l'identificateur de canal existant, le serveur distant ne sera pas en mesure de satisfaire cette requête sans soit émettre un identificateur de nouveau canal ou renvoyer un code d'erreur.

EXEMPLE 1: "target=nice.jp2&cnew=http" demande le premier canal d'une nouvelle session vers l'image "nice.jp2" au moyen du transport "http". Si aucun canal n'est assigné par le serveur distant, la requête sera traitée comme étant sans description d'état.

EXEMPLE 2: "cid=013ac8&cnew=http-tcp" demande un nouveau canal au cours de la même session, qui est associée à l'identificateur de canal 013ac8. Le nouveau canal vise à utiliser le transport "http-tcp" et se rapporte à la même cible logique que l'identificateur de canal 013ac8. Un même modèle de cache est partagé par ces canaux. Si aucun canal n'est assigné par le serveur distant, la requête sera traitée comme si le champ de nouveau canal de requête avait été omis.

EXEMPLE 3: "target=nice.jp2&cid=013ac8&cnew=http" demande un nouveau canal au cours de la même session, qui est associée à l'identificateur de canal "013ac8". Le nouveau canal vise à utiliser le protocole de transport "http". La cible logique associée au nouveau canal est différente de celle qui était associée à l'identificateur de canal "013ac8" et un modèle de cache distinct est utilisé pour le nouveau canal. Les modèles de cache pour les deux cibles sont associés à cette session commune.

C.3.4 Fermeture de canal (cclose)

```
cclose = "cclose" "=" ("*" / 1#channel-id)
```

Ce champ sert à fermer un ou plusieurs canaux ouverts à une session. Si le champ de valeur contient un ou plusieurs jetons d'identificateur de canal, ceux-ci doivent tous appartenir à la même session. Dans ce cas, le champ de requête d'identificateur de canal n'est pas nécessaire, mais s'il est fourni, il doit également désigner un canal appartenant à la même session.

Si le champ de valeur est "*", tous les canaux associés à la session seront fermés. Dans ce cas, la session doit être identifiée par l'inclusion d'un champ de requête d'identificateur de canal.

Le serveur distant doit terminer sa réponse concernant tout canal spécifié dans la demande de fermeture de canal avant de fermer réellement ce canal.

C.3.5 Identificateur de requête (qid)

```
qid = "qid" "=" UINT
```

Ce champ sert à spécifier une valeur d'identificateur de requête. Chaque canal a sa propre file d'attente de requêtes, avec son propre compteur d'identificateurs de requête. Les requêtes qui sont reçues dans un canal quelconque (ce qui est indiqué par la valeur de l'identificateur de canal) doivent être traitées dans l'ordre de leurs valeurs d'identificateur de requête, où le champ d'identificateur de requête est utilisé. Le serveur distant peut traiter, dans l'ordre de leur arrivée, les requêtes qui ne contiennent pas de champ d'identificateur de requête. Cependant, il ne doit pas traiter de requête qui arrive avec une valeur d'identificateur de requête égale à n tant qu'il n'a pas traité toutes les requêtes ayant une valeur d'identificateur de requête inférieure à n associée au même canal, sauf si $n=0$. Le client ne doit pas émettre de requête qui spécifie la même valeur d'identificateur de requête qu'une autre requête associée au même canal et ne doit pas émettre d'identificateur de requête qui soit plus petit que l'un quelconque des identificateurs de requête déjà émis dans ce canal.

C.4 Champs de requête de fenêtre de visualisation

C.4.1 Mappage des requêtes de fenêtre de visualisation vers des résolutions et régions d'image à flux codé

L'objet du protocole JPIP consiste à fournir des portions d'image JPEG 2000 et les métadonnées associées en réponse à des requêtes issues d'un client. Cette opération est effectuée au moyen d'une séquence de requêtes et de réponses.

Concernant la portion d'image, les données demandées peuvent être inférieures à l'image complète en termes de longueur de trame, de région, de qualité et/ou de composants d'image.

Dans le cas le plus simple, la portion d'image en question est définie directement par rapport à la grille de référence en haute résolution du (des) flux à codage JPEG 2000 identifié(s) dans la requête, et non par rapport à la grille échantillonnée d'un certain composant d'image. Plus généralement, cependant, les clients peuvent demander des objets d'image de très haut niveau (par exemple des couches de composition en format JPX ou des pistes vidéo en format MJ2) au moyen du champ de requête de contexte de flux codé (voir C.4.7). Dans ce cas, la portion d'image demandée peut devoir être soumise à une transformation de coordonnées afin de déterminer la portion de chaque flux codé associé qui doit être demandée. Ces transformations de coordonnées sont décrites en C.4.7 et doivent être comprises conformément à la description ci-après des régions d'image à flux codé.

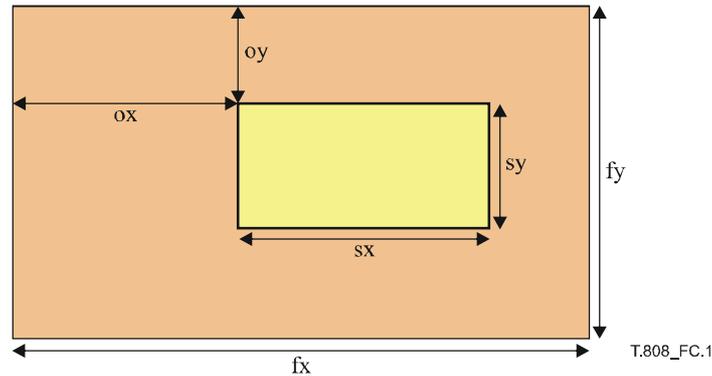


Figure C.1 – Recherche de région dans une image

Les régions d'image par flux codé sont décrites au moyen de 3 paramètres bidimensionnels, comme représenté dans la Figure C.1. Les paramètres de taille (s_x et s_y) et les paramètres de décalage (o_x et o_y) spécifient la largeur et la hauteur de la région d'image recherchée dans un flux codé, ainsi que le coin supérieur gauche de cette région, par rapport à une image entière qui a la longueur de trame indiquée (f_x et f_y).

EXEMPLE 1: Un client qui souhaite remplir un écran d'affichage à 640×480 pixels avec l'image entière pourrait faire une requête comme suit "`fsiz=640,480&rsiz=640,480&roff=0,0`". Noter que cette opération peut être effectuée quelle que soit la taille originale de l'image (et même sans la connaître).

Quand aucune des résolutions d'image disponibles dans le flux à codage JPEG 2000 ne correspond exactement à la longueur de trame demandée, les données d'image renvoyées peuvent être plus grandes ou plus petites que la longueur de trame demandée et peuvent même différer par leur format d'écran. Le serveur distant doit déterminer une résolution appropriée d'image à flux codé, indiquée par les paramètres de taille f_x' et f_y' ; il doit également déterminer une région appropriée dans le flux codé, indiquée par les paramètres s_x' , s_y' , o_x' et o_y' , comme représenté dans la Figure C.2. Bien que le client puisse spécifier le sens d'arrondissement, dans le cadre du champ de requête de longueur de trame, le client doit être disposé à traiter les données renvoyées qui ne correspondent pas exactement aux paramètres demandés.

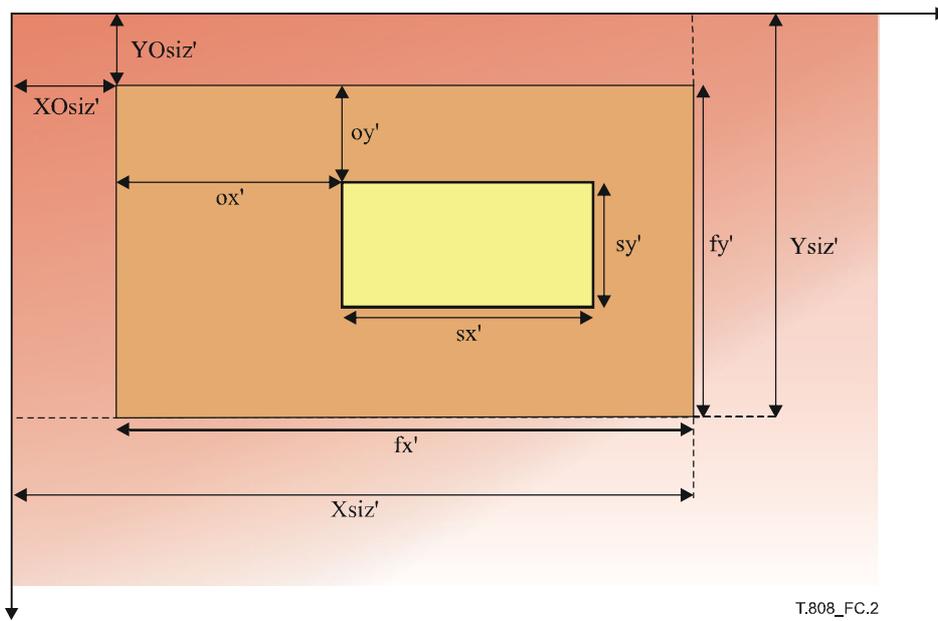


Figure C.2 – Recherche de région par rapport à la grille de référence sous-échantillonnée

Comme représenté dans la Figure C.2, la grandeur de la résolution appropriée de l'image par flux codé est indiquée par $fx' = Xsiz' - XOsiz'$ et $fy' = Ysiz' - YOsiz'$, où les termes $XOsiz'$, $YOsiz'$, $Xsiz'$ et $Ysiz'$ sont calculés au moyen de l'équation (C-1).

$$XOsiz' = \left\lceil \frac{XOsiz}{2^r} \right\rceil; \quad YOsiz' = \left\lceil \frac{YOsiz}{2^r} \right\rceil; \quad Xsiz' = \left\lceil \frac{Xsiz}{2^r} \right\rceil; \quad Ysiz' = \left\lceil \frac{Ysiz}{2^r} \right\rceil \quad (C-1)$$

où:

r est déterminé par le serveur distant afin de correspondre à la taille d'image demandée (fx et fy) aussi étroitement que possible, sous réserve de toutes préférences d'arrondissement fournies au moyen du champ de requête de longueur de trame.

Ici, les termes $XOsiz$, $YOsiz$, $Xsiz$ et $Ysiz$ sont extraits du segment correspondant de marqueur SIZ du flux codé. Il est naturel d'interpréter r comme étant le nombre des plus hauts niveaux de transformation DWT qui ont été rejetés: en fait, r doit impérativement être un entier non inférieur à 0. Cependant, la valeur de r n'est pas limitée par le nombre de niveaux de transformation DWT qui ont servi à compresser un quelconque élément de pavé dans le flux codé.

Une fois que la longueur de trame appropriée, fx' et fy' , a été découverte, la taille de la région, sx' et sy' , ainsi que le décalage, ox' et oy' , associés à la région d'image à flux codé, sont déterminés par l'équation (C-2).

$$ox' = \left\lfloor ox \cdot \frac{fx'}{fx} \right\rfloor; \quad oy' = \left\lfloor oy \cdot \frac{fy'}{fy} \right\rfloor; \quad sx' = \left\lfloor (sx + ox) \cdot \frac{fx'}{fx} \right\rfloor - ox'; \quad sy' = \left\lfloor (sy + oy) \cdot \frac{fy'}{fy} \right\rfloor - oy' \quad (C-2)$$

EXEMPLE 2: Supposons que la longueur de trame demandée soit de 128×128 et que l'image par flux codé sur la grille de référence en haute résolution soit décrite par $XO_{siz}=127$, $X_{siz}=648$, $YO_{siz}=0$ et $Y_{siz}=504$. Supposons également que 3 niveaux de transformation en ondelette existent pour tous les composants d'image contenus dans le flux codé. Les tailles disponibles d'image par flux codé sont donc:

$$\begin{aligned}
 521 \times 504 & \left(\left[\frac{648}{1} \right] - \left[\frac{127}{1} \right] \text{ by } \left[\frac{504}{1} \right] - 0 \right) \\
 260 \times 252 & \left(\left[\frac{648}{2} \right] - \left[\frac{127}{2} \right] \text{ by } \left[\frac{504}{2} \right] - 0 \right) \\
 130 \times 126 & \left(\left[\frac{648}{4} \right] - \left[\frac{127}{4} \right] \text{ by } \left[\frac{504}{4} \right] - 0 \right) \\
 65 \times 63 & \left(\left[\frac{648}{8} \right] - \left[\frac{127}{8} \right] \text{ by } \left[\frac{504}{8} \right] - 0 \right)
 \end{aligned}$$

Ainsi, si la requête vise une plus grande longueur de trame (la valeur du paramètre `round-direction` est `round-up`) la longueur de trame renvoyée sera 260×252 . Si la requête vise une plus petite longueur de trame (la valeur du paramètre `round-direction` est `round-down`), alors une longueur de trame de 65×63 sera utilisée. Noter que, comme dans cet exemple, les longueurs de trame disponibles de l'image par flux codé ne sont généralement pas des puissances de 2 exactes.

Le sous-échantillonnage d'un composant d'image, comme spécifié par XR_{siz} et YR_{siz} , n'a aucun effet sur l'interprétation de la région ni sur la résolution d'image demandée, dans tout flux codé demandé.

EXEMPLE 3: Une requête visant une région à 256×256 à partir du coin supérieur gauche d'une image à 512×512 peut être effectuée par:

`fsiz=512,512&rsiz=256,256`

Supposons que le flux codé contienne une image sous-échantillonnée dans les composants 1 et 2 mais non dans le composant 0. Spécifiquement, supposons que $X_{siz}=1024$, $Y_{siz}=1024$, $XO_{siz}=0$, $YO_{siz}=0$ et $XR_{siz}^0=1$, $YR_{siz}^0=1$, $XR_{siz}^1=2$, $YR_{siz}^1=2$, $XR_{siz}^2=2$ et $YR_{siz}^2=2$. Le serveur distant laissera de côté le plus haut niveau de résolution de ces trois composants et renverra suffisamment de pavés ou de districts pour offrir 256×256 échantillons du composant 0, mais seulement 128×128 échantillons des composants 1 et 2. Le client possède donc les données nécessaires pour afficher le coin supérieur gauche à la moitié de la taille de l'image complète mais encore sous-échantillonnée. Si le client souhaite afficher des composants de chrominance non sous-échantillonnés, il pourra émettre une requête additionnelle, comme la suivante:

`fsiz=1024,1024&rsiz=512,512&comps=1,2`

Le serveur distant renverra alors suffisamment de données pour offrir 256×256 échantillons des composants 1 et 2, qui pourront être combinés avec les données du composant 0 déjà reçues afin d'obtenir une image non sous-échantillonnée mais de taille réduite de moitié.

Si les trois composants avaient été sous-échantillonnés, le serveur distant n'offrirait que 128×128 échantillons des trois composants pour la requête originale (`fsiz=512,512&rsiz=256,256`) car la résolution d'image et les régions d'image sont évaluées par rapport à la grille de référence de chaque flux codé demandé.

C.4.2 Longueur de trame (`fsiz`)

`fsiz = "fsiz" "=" fx ", " fy ["", " round-direction]`

`fx = UINT`

`fy = UINT`

`round-direction = "round-up" / "round-down" / "closest"`

Ce champ sert à identifier la résolution associée à la fenêtre de visualisation demandée. Les valeurs `fx` et `fy` spécifient les dimensions de la résolution d'image recherchée. La valeur du paramètre `round-direction` spécifie comment une résolution disponible d'image par flux codé doit être choisie pour chaque flux codé demandé, si la résolution d'image demandée n'est pas disponible dans ce flux codé. La longueur de trame demandée est appliquée sur une résolution d'image à flux codé, suivant la procédure décrite en C.4.1, éventuellement avec l'adjonction de transformations de coordonnées, demandées au moyen d'un champ de requête de contexte de flux codé (voir C.4.7). Un

client qui souhaite régler le nombre exact d'échantillons reçus pour un certain composant d'image peut devoir augmenter la longueur de trame demandée, comme expliqué en C.4.1. Les options relatives au paramètre "round-direction" (sens d'arrondissement) défini par la présente Recommandation | Norme internationale sont décrites dans le Tableau C.1.

Tableau C.1 – Options relatives au paramètre "round-direction"

Round-direction	Signification
"round-up"	Pour chaque flux codé demandé, la plus petite résolution d'image par flux codé dont la largeur et la hauteur sont toutes les deux supérieures ou égales à la taille spécifiée doit être choisie. S'il n'y en a aucune, alors la plus grande résolution disponible d'image par flux codé doit être utilisée.
"round-down"	Pour chaque flux codé demandé, la plus grande résolution d'image par flux codé dont la largeur et la hauteur sont toutes les deux inférieures ou égales à la taille spécifiée doit être choisie. Il s'agit de la valeur par défaut quand le paramètre "round-direction" n'est pas spécifié.
"closest"	Pour chaque flux codé demandé, la résolution d'image par flux codé qui est la plus proche ("closest") de la taille spécifiée dans la zone (où zone = $fx \times fy$) doit être choisie. Lorsque les deux résolutions d'image par flux codé ont des zones qui sont équidistantes de $fx \times fy$, la plus grande des deux doit être choisie.

Si le champ de requête de longueur de trame est omis d'une requête de fenêtre de visualisation et que le paramètre "metadata-only" ne soit pas spécifié dans un champ de requête de métadonnées (voir C.5.1), la fenêtre de visualisation demandée ne contient ni données d'image comprimée ni en-tête spécifique de pavé, mais comprend bien toutes les autres informations d'en-tête (flux codé et format de fichier) qui auraient été renvoyées si le client avait inclus le champ de requête de longueur de trame. Voir au C.5.1 de plus amples détails sur les informations relatives au format de fichier (métadonnées) qui sont implicitement demandées en même temps que la requête de fenêtre de visualisation.

C.4.3 Décalage (roff)

roff = "roff" "=" ox ", " oy

ox = UINT

oy = UINT

Ce champ sert à identifier le coin gauche supérieur (décalage) de la région spatiale associée à la fenêtre de visualisation demandée; si ce champ n'est pas présent, les décalages reviennent par défaut à 0. Le déplacement effectif d'une région d'image par flux codé à partir du coin gauche supérieur de l'image, à la résolution effective d'image par flux codé choisie par le serveur distant, est obtenu suivant la procédure décrite en C.4.1, éventuellement avec l'adjonction de transformations de coordonnées, demandées au moyen d'un champ de requête de contexte de flux codé (voir C.4.7).

L'utilisation du champ de décalage n'est valide que conjointement avec le champ de requête de longueur de trame.

Si une région d'image par flux codé spécifié, utilisant le champ de taille de région et/ou de décalage, se révèle vide (sans zone), la réponse du serveur distant ne devrait pas inclure de données d'image comprimée pour ce flux codé. En particulier, les réponses de type flux JPP ou JPT ne devraient contenir aucun message faisant référence à des segments de données de district, de pavé ou d'en-tête de pavé, contenus dans ce flux codé. Le serveur distant peut, à sa discrétion, choisir de renvoyer les messages contenant un en-tête principal ou un segment de métadonnées qui auraient été renvoyés en réponse à une requête ayant omis le champ de requête de longueur de trame.

C.4.4 Taille de région (rsiz)

rsiz = "rsiz" "=" sx ", " sy

sx = UINT

sy = UINT

Ce champ sert à identifier l'étendue horizontale et verticale (taille) de la région spatiale associée à la fenêtre de visualisation demandée; si ce champ n'est pas présent, la région s'étend jusqu'au coin inférieur droit de l'image. Les dimensions effectives d'une région d'image à flux codé, à la résolution effective d'image par flux codé choisie par le serveur distant, sont calculées suivant la procédure décrite en C.4.1, éventuellement avec l'adjonction de transformations de coordonnées demandées au moyen d'un champ de requête de contexte de flux codé (voir C.4.7). Une région d'image par flux codé qui est demandée n'a pas nécessairement besoin d'être entièrement contenue dans le flux codé; auquel cas le serveur distant prend simplement l'intersection entre la région disponible d'image par flux codé et la région demandée.

L'utilisation du champ de requête de taille de région n'est valide que conjointement avec le champ de requête de longueur de trame.

La région d'image par flux codé peut être vide, par exemple si s_x ou s_y avait la valeur zéro. Si elle est vide, alors la réponse du serveur distant ne devrait pas inclure de données d'image comprimée pour ce flux codé. En particulier, les réponses de type flux JPP ou JPT ne devraient contenir aucun message faisant référence à des segments de données de district, de pavé ou d'en-tête de pavé dans ce flux codé. Le serveur distant peut, à sa discrétion, choisir de renvoyer les messages contenant un en-tête principal ou un segment de métadonnées qui auraient été renvoyés en réponse à une requête ayant omis le champ de requête de longueur de trame.

C.4.5 Composants (comps)

```
comps = "comps" "=" 1#UINT-RANGE
```

Ce champ sert à identifier les composants d'image qui doivent être inclus dans la fenêtre de visualisation demandée; si ce champ n'est pas présent, la requête est interprétée comme incluant tous les composants disponibles d'image de tous les flux codés identifiés au moyen du champ de requête de flux codé, ainsi que tous les composants applicables de tous les flux codés demandés au moyen du champ de requête de contexte de flux codé (voir C.4.7). Ces composants "applicables" sont ceux qui sont impliqués dans la reproduction des entités d'image (par exemple des couches de composition en format JPX ou des pistes vidéo en format MJ2) qui sont spécifiées au moyen du champ de requête de flux codé.

Les valeurs contenues dans ce champ de requête représentent les indices des composants d'image recherchés. Les indices de composant d'image commencent à 0 et ont la signification qui leur est attribuée par la syntaxe du flux à codage JPEG 2000, comme décrit dans la Rec. UIT-T T.800 | ISO/CEI 15444-1. Il convient cependant de noter que ce sont les composants qui sont obtenus par décodage et par transformation inverse en ondelette des données comprimées, avant l'application de la transformation inverse RCT ou ICT de ces composants. Dans les flux codés conformes à la Rec. UIT-T T.801 | ISO/CEI 15444-2, les composants identifiés ici le sont comme "composants spatiaux", c'est-à-dire obtenus par décodage et transformation inverse en ondelette des données comprimées, avant l'application de toute transformation inverse de composants multiples, transformation conditionnelle de composant, ou transformation en ondelette de composants multiples.

Les composants inexistant dans l'un quelconque des flux codés demandés doivent être négligés.

C.4.6 Flux codé (stream)

```
stream = "stream" "=" 1#sampled-range
sampled-range = UINT-RANGE [":" sampling-factor]
sampling-factor = UINT
```

Ce champ sert à identifier quel(s) flux codé(s) appartient (appartiennent) à la fenêtre de visualisation demandée. Si ce champ est omis et si le(s) flux codé(s) ne peut (peuvent) pas être déterminé(s) autrement, la valeur par défaut est l'unique flux codé ayant l'identificateur 0. Noter que le champ de requête de contexte de flux codé (voir C.4.7) offre un moyen supplémentaire de demander des flux codés.

Pour les cibles de la famille JPEG 2000, les indices de flux codé sont ceux qui sont imbriqués dans la boîte générique correspondante qui apparaît dans le segment approprié de métadonnées, comme décrit en A.3.6. Pour les formats de fichier qui ont impliqué des identificateurs de flux codé, ces identificateurs devraient s'accorder avec les indices utilisés ici.

Lorsqu'une étendue de flux codés est identifiée, l'absence de limite supérieure signifie que l'étendue s'étend à tous les flux codés ayant des identificateurs plus grands. Lorsqu'une limite supérieure est fournie, elle fournit l'identificateur absolu du dernier flux codé contenu dans l'étendue.

Qu'une limite supérieure soit ou non fournie, une étendue de flux codés peut être qualifiée par un facteur d'échantillonnage additionnel qui, s'il est fourni, doit être un entier strictement positif, F . L'étendue contient alors tous les identificateurs de flux codé $L+Fk$ qui se trouvent dans l'étendue non qualifiée, où L est l'identificateur du premier flux codé dans l'étendue. L'indice du client du flux codé recherché est k , qui est un entier de forme UINT.

C.4.7 Contexte du flux codé (context)

```

context = "context" "=" 1#context-range
context-range = jpx1-context-range / mj2t-context / reserved-context
jpx1-context-range = "jpx1" "<" jpx-layers ">" [ "[" jpx1-geometry "]" ]
jpx-layers = sampled-range
jpx1-geometry = "s" jpx-iset "i" jpx-inum
jpx-iset = UINT
jpx-inum = UINT
mj2t-context = "mj2t" "<" mj2-track ">" [ "[" mj2t-geometry "]" ]
mj2-track = NONZERO ["+" "now" ]
mj2t-geometry = "track" / "movie"
reserved-context = 1*( TOKEN / "<" / ">" / "[" / "]" / "-" / ":" / "+" )

```

Ce champ peut servir à demander des flux codés indirectement au moyen des entités d'image "de niveau supérieur". La présente Recommandation | Norme internationale définit des contextes correspondant à des couches de composition en format JPX (une couche de composition JPX peut impliquer un ou plusieurs flux codés) et à des pistes vidéo en format MJ2; cependant, le mécanisme est conçu de façon à pouvoir faire l'objet d'une extension.

Si un champ de requête de contexte de flux codé est fourni, la fenêtre de visualisation demandée contient chacun des flux codés qui sont associés au(x) contexte(s) demandé(s), en plus de tout flux codé demandé au moyen du champ de requête de flux codé.

Le corps d'un champ de requête de contexte de flux codé se compose d'une ou de plusieurs valeurs du paramètre "context-range". Chaque paramètre "context-range" est associé à un ensemble de flux codés qui peuvent être déterminés par le serveur distant. Un paramètre "context-range" (étendue du contexte) peut également identifier des transformations par remappage de coordonnées, qui doivent être appliquées aux paramètres de longueur de trame, de taille de région et de décalage afin de déterminer la résolution d'image par flux codé et la région d'image par flux codé pour chacun des flux codés associés à cette étendue de contexte. Lorsque le serveur distant est disposé à traiter un paramètre "context-range" (étendue du contexte), il doit identifier les flux codés qui sont associés à cette étendue de contexte au moyen d'un en-tête de réponse concernant le contexte de flux codé.

La présente Recommandation | Norme internationale définit deux types spécifiques d'étendue de contexte, qui sont destinés à répondre aux besoins des formats de fichier JPX et MJ2. Le premier de ces types d'étendue de contexte, "jpx1-context-range", sert à identifier une ou plusieurs couches de composition en format JPX. Les indices des couches de composition associées à un paramètre "jpx1-context-range" sont fournis sous la forme d'une étendue échantillonnée "sampled-range", conformément à la même sémantique que les étendues de flux codé échantillonné indiquées dans le champ de requête de flux codé. Lorsqu'un paramètre "jpx1-context-range" est traité par le serveur distant, les flux codés appartenant à la (aux) couche(s) de composition correspondante(s) doivent être identifiés dans un en-tête de réponse concernant le contexte de flux codé.

Un paramètre "jpx1-context-range" peut identifier une transformation facultative par remappage de coordonnées, à utiliser pour calculer la résolution d'image par flux codé et la région d'image par flux codé pour chacun de ses flux codés. Cette transformation par remappage de coordonnées est déterminée par deux entiers non négatifs, jpx-iset et jpx-inum. Ensemble, ces deux entiers identifient une certaine instruction de composition dans une boîte de composition (comp) en format JPX, découverte dans le domaine d'application de la cible logique. L'instruction spécifique en question est située dans la boîte de jeu d'instructions (iset) dont la position ordinale (à partir de 0) dans la boîte de composition est indiquée par la valeur du paramètre "jpx-iset". La valeur du paramètre "jpx-inum" donne la position ordinale (à partir de 0) de l'instruction contenue dans cette boîte de jeu d'instructions. L'interprétation de ces indices est indépendante des décomptes de répétition qui peuvent apparaître dans une boîte de composition en format JPX.

Quand des valeurs "jpx-iset" et "jpx-inum" sont traitées par le serveur distant, les paramètres demandés de longueur de trame et de région, fx, fy, sx, sy, ox et oy, doivent d'abord être appliqués sur les paramètres modifiés de longueur de trame et de région, fx", fy", sx", sy", ox" et oy" au moyen des expressions contenues dans l'équation C-3. Ces paramètres modifiés de région doivent être calculés séparément pour chaque flux codé demandé et doivent ensuite être utilisés à la place des paramètres fx, fy, sx, sy, ox et oy lors de la détermination de la résolution d'image par flux codé et de la région d'image par flux codé suivant la procédure décrite en C.4.1.

$$\begin{aligned}
f_x'' &= \left[f_x \cdot \frac{X_{R_{reg}}}{X_{S_{reg}}} \cdot \frac{W_{t_{inst}}}{W_{S_{inst}}} \cdot \frac{W_{cod}}{W_{comp}} \right]; & f_y'' &= \left[f_y \cdot \frac{Y_{R_{reg}}}{Y_{S_{reg}}} \cdot \frac{H_{t_{inst}}}{H_{S_{inst}}} \cdot \frac{H_{cod}}{H_{comp}} \right] \\
s_x'' &= \min \{ (o_x + s_x), x_{lim} \} - \max \{ o_x, x_{min} \} \\
s_y'' &= \min \{ (o_y + s_y), y_{lim} \} - \max \{ o_y, y_{min} \} \\
o_x'' &= \max \{ o_x, x_{min} \} - \left[\left(X_{O_{inst}} - \left(X_{C_{inst}} - \frac{X_{O_{reg}}}{X_{S_{reg}}} \right) \cdot \frac{W_{t_{inst}}}{W_{S_{inst}}} \right) \cdot \frac{f_x}{W_{comp}} \right] \\
o_y'' &= \max \{ o_y, y_{min} \} - \left[\left(Y_{O_{inst}} - \left(Y_{C_{inst}} - \frac{Y_{O_{reg}}}{Y_{S_{reg}}} \right) \cdot \frac{H_{t_{inst}}}{H_{S_{inst}}} \right) \cdot \frac{f_y}{H_{comp}} \right] \\
x_{min} &= \left[X_{O_{inst}} \cdot \frac{f_x}{W_{comp}} \right]; & y_{min} &= \left[Y_{O_{inst}} \cdot \frac{f_y}{H_{comp}} \right] \\
x_{lim} &= \left[\left(X_{O_{inst}} + W_{t_{inst}} \right) \cdot \frac{f_x}{W_{comp}} \right]; & y_{lim} &= \left[\left(Y_{O_{inst}} + H_{t_{inst}} \right) \cdot \frac{f_y}{H_{comp}} \right]
\end{aligned} \tag{C-3}$$

Noter que la région modifiée de fenêtre de visualisation, définie par s_x'' , s_y'' , o_x'' et o_y'' , peut éventuellement se trouver légèrement à gauche ou au-dessus de l'origine. C'est-à-dire que o_x'' et/ou o_y'' peut être négatif. Toute portion de la région de fenêtre de visualisation qui se trouve à gauche ou au-dessus de l'origine devrait être ignorée lors de la détermination de la région d'image par flux codé suivant la procédure décrite en C.4.1.

Si les valeurs des paramètres $jpx\text{-iset}$ et $jpx\text{-inum}$ ne sont pas fournies, les paramètres modifiés de région à utiliser à la place de f_x , f_y , s_x , s_y , o_x et o_y sont indiqués par les expressions contenues dans l'équation C-4. Comme précédemment, ces paramètres modifiés doivent être utilisés lors de la détermination de la résolution d'image par flux codé et de la région d'image à flux codé, suivant la procédure indiquée en C.4.1.

$$\begin{aligned}
f_x'' &= \left[f_x \cdot \frac{X_{R_{reg}}}{X_{S_{reg}}} \cdot \frac{W_{cod}}{W_{reg}} \right]; & f_y'' &= \left[f_y \cdot \frac{Y_{R_{reg}}}{Y_{S_{reg}}} \cdot \frac{H_{cod}}{H_{reg}} \right] \\
o_x'' &= o_x - \left[\frac{X_{O_{reg}}}{X_{S_{reg}}} \cdot \frac{f_x}{W_{reg}} \right]; & o_y'' &= o_y - \left[\frac{Y_{O_{reg}}}{Y_{S_{reg}}} \cdot \frac{f_y}{H_{reg}} \right] \\
s_x'' &= s_x; & s_y'' &= s_y
\end{aligned} \tag{C-4}$$

Le second type d'étendue de contexte décrit par la présente Recommandation | Norme internationale, "mj2t-context", permet aux clients de demander des pistes spécifiques à partir d'un fichier MJ2. L'identificateur de piste "mj2-track" doit impérativement être un entier strictement positif, car 1 est le plus petit identificateur de piste admissible qui soit autorisé dans un fichier MJ2. Si un identificateur de piste mj2 contient le suffixe facultatif "+now", le contexte mj2t se compose de tous les flux codés appartenant à la piste vidéo de format MJ2, à partir du flux codé dont l'instant de capture correspond à celui auquel la requête est reçue. Cela est utile quand la source est un flux vidéo en direct. Sinon, le serveur distant peut associer le suffixe "+now" à tout flux codé qu'il juge utile. Si le suffixe "+now" n'est pas inclus, le contexte mj2 se compose de tous les flux codés appartenant à la piste vidéo de format MJ2.

Un paramètre "mj2t-context" peut spécifier une transformation par remappage de coordonnées, à utiliser lors du calcul de résolutions d'image par flux codé et de régions d'image par flux codé pour chacun de ses flux codés. Si ce champ n'est pas présent, les paramètres de longueur de trame et de région fournis au moyen des champs de requête de longueur de trame, décalage et taille de région doivent être interprétés directement, conformément à la procédure décrite en C.4.1. Sinon, un des deux types de transformation de coordonnées doit être demandé, tel qu'il est identifié par l'apparition d'un des jetons "track" ou "movie".

Lorsque le jeton "track" est spécifié, les champs de requête de longueur de trame, décalage et taille de région doivent être utilisés afin d'identifier une taille de présentation recherchée et une région rectangulaire recherchée dans le plus petit rectangle englobant qui puisse contenir la présentation de piste, à cette taille de présentation recherchée. Les transformations géométriques décrites par la boîte d'en-tête de piste MJ2 (tkhd) doivent être appliquées afin de déterminer une résolution et une région d'image correspondantes dans chaque flux codé associé à la piste.

Lorsque le jeton "movie" est spécifié, les champs de requête de longueur de trame, de décalage et de taille de région doivent être utilisés afin d'identifier une taille recherchée pour la séquence cinématographique entière (éventuellement

composée) qui est reproduite et afin d'identifier une région rectangulaire recherchée dans le plus petit rectangle englobant qui contient le jeton "movie", à cette taille recherchée. Les transformations géométriques décrites par la boîte d'en-tête de piste MJ2 (tkhd) doivent être combinées avec les transformations géométriques décrites par la boîte d'en-tête de séquence cinématographique (mvhd) et doivent être appliquées afin de déterminer une résolution et une région d'image correspondantes dans chaque flux codé associé à la piste.

Si un serveur distant est dans l'impossibilité d'appliquer l'une quelconque des transformations géométriques de contexte mj2t décrites ci-dessus, il offre une chaîne modifiée de contexte mj2t dans son en-tête de réponse concernant le contexte de flux codé.

NOTE 1 – L'utilisation du champ de requête de contexte de flux codé ainsi que du champ de requête de flux codé peut se traduire par de multiples demandes de flux codé avec différentes transformations géométriques des champs de requête de longueur de trame, de taille de région et de décalage. Si cela se produit, de multiples portions d'image disjointes ou superposées dans ce flux codé seront effectivement demandées.

NOTE 2 – Les expressions contenues dans l'équation C-4 peuvent de même être obtenues par le réglage suivant: $X_{S_{comp}}=W_{S_{inst}}=W_{t_{inst}}=W_{reg}$, $Y_{S_{comp}}=H_{S_{inst}}=H_{t_{inst}}=H_{reg}$ et $XO_{inst}=YO_{inst}$ dans l'équation C-3 quand les limites relatives aux paramètres s_x , s_y , o_x et o_y ne sont pas bornées par X_{lim} , X_{min} , Y_{lim} , Y_{min} .

EXEMPLE 1: "context=jpxl<0-4:2>[s5i2]"

Dans ce cas, le serveur distant est invité à renvoyer les flux codés qui sont utilisés par des couches de composition en format JPX 0, 2 et 4, par le remappage de la longueur de trame et de la région d'image demandées conformément aux réglages géométriques représentés par la troisième instruction de la sixième boîte de jeu d'instructions contenue dans la boîte de composition (les fichiers JPX ont au plus une seule boîte de composition).

EXEMPLE 2: "stream=0&context=mj2t<1+now>[track]"

Dans ce cas, le serveur distant est invité à renvoyer le flux codé 0, ainsi que tous les flux codés appartenant à la première piste d'un fichier MJ2, à partir du flux codé dont l'instant d'échantillonnage correspond à l'instant actuel. En outre, le serveur distant est invité à remapper les longueurs de trame et région d'image demandées conformément aux réglages géométriques décrits dans la boîte d'en-tête de piste, en négligeant tous réglages géométriques additionnels qui pourraient être décrits dans la boîte d'en-tête de séquence cinématographique.

C.4.8 Fréquence d'échantillonnage (srate)

```
srate = "srate" "=" streams-per-second
streams-per-second = UFLOAT
```

Si ce champ est fourni, les flux codés qui appartiennent à la fenêtre de visualisation sont obtenus par sous-échantillonnage de ceux qui sont mentionnés par le champ de requête de flux codé, en plus de ceux qui sont étendus à partir de valeurs du paramètre "context-range" contenues dans le champ de requête de contexte de flux codé (voir C.4.7), de façon à réaliser une fréquence d'échantillonnage moyenne non supérieure à la valeur du nombre de flux par seconde. Cela n'est possible que si les flux codés ont des informations temporelles associées (par exemple s'ils appartiennent à une cible logique conforme au format de fichier MJ2).

Ce champ de requête ne sert qu'à déterminer quels flux codés devraient être considérés comme appartenant à la fenêtre de visualisation. Le serveur distant doit explorer tous les flux codés qui seraient sinon inclus dans la fenêtre de visualisation, en rejetant si nécessaire des flux codés afin de garantir que la séparation moyenne entre les temps des sources de flux codé n'est pas inférieure à l'inverse de la valeur du nombre de flux par seconde. La présente Recommandation | Norme internationale ne prescrit pas d'algorithme pour le sous-échantillonnage, ni une interprétation précise du terme "séparation moyenne".

Si aucune information de rythme de la source n'est disponible, la fenêtre de visualisation se composera de tous les flux codés identifiés au moyen du champ de requête de flux codé et du champ de requête de contexte de flux codé, mais ce champ de requête peut néanmoins affecter l'interprétation d'un champ de requête de vitesse d'acheminement, s'il est présent.

C.4.9 ROI (roi)

```
roi = "roi" "=" region-name
region-name = 1*(DIGIT / ALPHA / "_")
/ "dynamic"
```

Ce champ spécifie la région spatiale recherchée de l'image au moyen d'un nom plutôt qu'au moyen de coordonnées. Le mappage entre le nom de région et une certaine région spatiale de l'image peut provenir de plusieurs endroits; il peut être défini dans une boîte de description de région ROI se trouvant dans la cible logique, ou être défini dans le cadre de l'implémentation du serveur distant proprement dit.

Une valeur du paramètre "region-name" égale à "dynamic" (région ROI dynamique) est réservée à la représentation d'une région non constante de l'image qui est appliquée sur une région spatiale indépendamment de toutes les requêtes. Le serveur distant peut utiliser toute information sur le client et tout autre paramètre de requête quand il détermine quelles régions spatiales il va fournir en réponse à cette requête particulière. Par exemple, si le serveur distant est informé du fait que l'affichage physique chez le client est très petit, il peut choisir d'offrir seulement la région d'avant-plan de l'image à une résolution plus élevée plutôt que l'entière région de l'image à une résolution inférieure. Les serveurs distants ne sont pas tenus de prendre en charge les régions ROI dynamiques.

Si un champ de région ROI existe et que le serveur distant soit informé de la façon de traiter la requête de région ROI, alors le champ de région ROI a priorité sur le champ de requête de décalage et sur les champs de taille de région, qui doivent être ignorés par le serveur distant. Si un champ de région ROI existe, mais que le serveur distant ne sache pas comment le traiter pour une cause quelconque, ce serveur distant doit ignorer le champ de région ROI et utiliser les champs de décalage et de taille de région. Si ces champs sont omis, les valeurs par défaut de ces champs doivent être utilisées.

Si le client spécifie une longueur de trame ainsi qu'une région ROI et si le serveur distant interprète la région ROI spécifiée, la valeur du champ de requête de longueur de trame détermine la résolution d'image à laquelle la région ROI est demandée.

C.4.10 Couches (layers)

```
layers = "layers" "=" UINT
```

Ce champ peut servir à limiter le nombre de couches de qualité du flux codé qui appartiennent à la requête de fenêtre de visualisation. Par défaut, toutes les couches disponibles sont à examiner. La valeur spécifie le nombre de couches de qualité initiales qui sont à examiner. Le serveur distant ne devrait pas essayer d'augmenter d'éventuels segments de données de district au-delà de la limite de couche applicable. Le serveur distant ne devrait pas essayer d'augmenter d'éventuels segments de données de pavé au-delà du point auquel tout le contenu restant se trouve au-delà de la limite de couche applicable. Compte tenu de l'ordre des données dans un pavé, il peut être nécessaire que le serveur distant ne renvoie des données, situées au-delà de la limite de la couche demandée, que pour des requêtes de flux en format JPT.

C.5 Champs de requête de métadonnées

C.5.1 Métadonnées demandées implicitement avec des requêtes de fenêtre de visualisation

Le champ de requête de flux codé et le champ de requête de contexte de flux codé désignent un ou plusieurs flux codés qui sont associés à la fenêtre de visualisation demandée. Même si aucun de ces champs de requête n'est présent, la fenêtre de visualisation est associée à au moins 1 flux codé, comme mentionné en C.4.6. En outre, comme noté en C.4.2, même si le champ de requête de longueur de trame est omis, la fenêtre de visualisation demandée contient au moins l'en-tête de flux codé principal pour chaque flux codé demandé. La seule exception à cette règle se présente lorsque le paramètre "metadata-only" (métadonnées seulement) est spécifié dans un champ de requête de métadonnées (voir C.5.2). A cette exception près, le client va donc demander implicitement toutes les boîtes de métadonnées qui pourraient être requises à partir du format de fichier, le cas échéant, afin d'utiliser l'imagerie représentée par les flux codés demandés. Afin de garantir l'interopérabilité entre client et composants de serveur distant, ce paragraphe désigne un ensemble minimal de métadonnées que les serveurs distants doivent considérer comme étant implicitement demandées en même temps que la fenêtre de visualisation. Lorsque le serveur distant est informé de l'existence d'éléments additionnels de métadonnées applicables, il peut les remettre également.

Pour les fichiers en formats JP2 et JPX, les éléments de métadonnées suivants doivent être considérés comme étant demandés en même temps que la fenêtre de visualisation:

- a) Le contenu entier du segment de métadonnées 0.
- b) Le contenu entier de chacune des boîtes suivantes, chaque fois qu'elles sont découvertes au niveau sommital du fichier:
 - 1) signature de format JP2 ("jP");
 - 2) type de fichier ("ftyp");
 - 3) exigences du lecteur ("rreq");
 - 4) composition ("comp").
- c) Tous les en-têtes de sous-boîte immédiate à partir de chacune des superboîtes suivantes:
 - 1) toute boîte à en-tête JP2 ("jp2h");
 - 2) toute boîte à en-tête de flux codé ("jpch") associée à un flux codé demandé;
 - 3) toute boîte à en-tête de couche de composition ("jplh") associée à une couche de composition JPX demandée au moyen du champ de requête de contexte de flux codé.

- d) Le contenu entier de chacune des boîtes suivantes, chaque fois que ces boîtes sont découvertes dans une des superboîtes mentionnées ci-dessus:
- 1) en-tête d'image ("ihdr");
 - 2) bits par composant ("bpcc");
 - 3) palette ("pclr");
 - 4) mappage de composant ("cmap");
 - 5) définition de canal ("cdef");
 - 6) résolution ("res");
 - 7) enregistrement de flux codé ("creg");
 - 8) opacité ("opct").
- e) Pour les fichiers JP2, les fichiers compatibles avec le format JP2 et les fichiers JPX, une ou plusieurs boîtes de description d'espace chromatique ("colr") associées à chaque flux codé ou à chaque couche de composition JPX demandée au moyen du champ de requête de contexte de flux codé, comme suit:
- 1) Si le serveur distant est en mesure de déterminer exactement quelle boîte est préférée, il devrait n'envoyer que cette boîte, même si cela signifie le non-envoi de la première boîte pour les fichiers JP2 ou pour les fichiers compatibles avec le format JP2 (par exemple si la seconde boîte applique une méthode quelconque du consortium ICC et si les préférences relatives à l'espace chromatique spécifient que le client préfère les boîtes à méthode quelconque du consortium ICC). Si le serveur distant n'est pas en mesure de déterminer exactement quelle boîte est préférée, il devrait envoyer la totalité de la première boîte de description d'espace chromatique.
 - 2) Pour toutes les boîtes non envoyées, le serveur distant devrait envoyer une portion du contenu de boîte de façon que le client puisse déterminer s'il y aura lieu de demander ultérieurement une autre spécification d'espace chromatique.
 - Pour les boîtes énumérées, le serveur distant devrait envoyer au moins les 7 premiers octets du contenu de boîte (au moins jusqu'au champ "enumCS").
 - Pour les boîtes d'espace chromatique définies par le vendeur, le serveur distant devrait envoyer au moins les 19 premiers octets du contenu de boîte (au moins jusqu'au champ "VCLR").
 - Pour les boîtes d'espace chromatique à méthode ICC restreinte ou quelconque, le serveur distant devrait envoyer au moins les 3 premiers octets du contenu de boîte (au moins les champs "METH", "APPROX" et "PREC").

Le serveur distant est invité à renvoyer un préfixe initial de chaque segment de métadonnées qui contient l'une quelconque des métadonnées mentionnées ci-dessus, en allant du premier octet du segment de métadonnées jusqu'à la fin de toutes les métadonnées demandées par rapport à ce segment de métadonnées. En conséquence, la quantité réelle de métadonnées renvoyées par le serveur distant peut dépendre de la façon particulière dont la cible logique a été partitionnée dans le segment de métadonnées. Une analyse de ces questions peut être consultée en A.3.6.2.

La présente Recommandation | Norme internationale ne donne pas d'avis sur ce qui constitue les métadonnées MJ2 implicites dans les requêtes de fenêtre de visualisation. Cela pourra cependant être défini dans une future norme.

C.5.2 Requête de métadonnées (metareq)

```

metareq = "metareq" "=" 1#("[" 1$(req-box-prop) "]" [root-bin] [max-depth])
                                     [metadata-only]

req-box-prop = box-type [limit] [metareq-qualif] [priority]

limit = ":" (UINT / "r")

metareq-qualif = "/" 1*("w" / "s" / "g" / "a")

priority = "!"

root-bin = "R" UINT

max-depth = "D" UINT

metadata-only = "!!"
    
```

Ce champ spécifie quelles métadonnées sont recherchées en réponse à cette requête, en plus de toutes les métadonnées requises afin que le client décode ou interprète les données d'image demandées (voir C.5.1). La chaîne de valeur

contenue dans ce champ de requête est une liste de requêtes indépendantes; cependant, le serveur distant peut manipuler les requêtes comme un groupe et il peut y avoir un chevauchement entre les requêtes.

Chaque requête est relative au segment de données spécifié par la valeur de son paramètre "root-bin": si cette valeur n'est pas spécifiée, la racine est le segment de métadonnées 0. La requête se rapporte seulement aux données contenues ou désignées (au moyen de boîtes génériques) dans ce segment particulier de données.

Si une valeur est spécifiée pour "max-depth", alors seules les boîtes contenues dans le segment radical de métadonnées, ainsi que les niveaux non supérieurs à "max-depth" dans la hiérarchie des fichiers situés au-dessous de ces boîtes, sont demandés. Si une valeur pour "max-depth" n'est pas spécifiée, il n'y a aucune limite à la profondeur de la hiérarchie des fichiers pour cette requête.

La portion "req-box-prop" de la requête spécifie une liste de types de boîte qui sont recherchés par le client. La chaîne spéciale "*" peut être substituée au type de boîte, auquel cas tous les types de boîte sont impliqués. Chaque type de boîte (ou "*") peut être suivi par toute combinaison de trois attributs: une valeur limite, un qualificateur de requête de métadonnées ("metareq") et un fanion de priorité.

La valeur limite spécifie le type d'information et la proportion du contenu de boîte que le client va demander pour ce type de boîte. Le paramètre "limit" prend la forme d'un caractère de deux points superposés, suivi par une valeur (la valeur limite), qui doit être soit un entier non signé ou le caractère "r".

Si la valeur limite est un entier n plus grand que zéro, alors le serveur distant est invité à renvoyer seulement les n premiers octets du contenu des boîtes correspondantes de ce type de boîte, en plus des en-têtes de boîte. Si la valeur limite est 0, alors seuls les en-têtes des boîtes de ce type sont demandés. Si la valeur limite n'est pas spécifiée, alors le client va demander le contenu entier de toutes les boîtes de ce type qui correspondent à d'autres aspects de la requête, que les boîtes de ce type soient ou non des superboîtes. De même, dans le cas d'une valeur limite numérique ou non spécifiée concernant une superboîte, le serveur distant est invité à offrir la quantité de données demandées par la valeur limite, que la hiérarchie contenue dans cette superboîte soit ou non plus profonde que celle qui aurait été atteinte sur la base des valeurs des paramètres "root-bin" et "max-depth" et quel que soit le type des sous-boîtes découvertes dans la superboîte.

Si la valeur limite est "r", alors le serveur distant est invité à envoyer l'en-tête, mais non le contenu, de toute boîte du type indiqué, ainsi que de toutes ses sous-boîtes hiérarchiquement inférieures (quel que soit leur type), jusqu'à profondeur maximale spécifiée dans la requête. C'est en fait une requête visant un squelette de cette portion de la hiérarchie des boîtes. Si le serveur distant est dans l'impossibilité de déterminer qu'une boîte est une superboîte, ce serveur pourrait ne pas être en mesure de descendre hiérarchiquement dans les sous-boîtes de cette boîte, de sorte qu'il pourrait ne pas satisfaire complètement à certaines requêtes de métadonnées. Les serveurs distants devraient être en mesure de reconnaître le statut de la superboîte de toutes les boîtes définies par les formats de fichier qu'ils sont destinés à prendre en charge.

Alors qu'une valeur limite de "r" signifie que le client va demander un squelette de la structure de boîte, composée des en-têtes de boîte, la subdivision de la cible logique en segments de métadonnées peut forcer le serveur distant à renvoyer des données supplémentaires, y compris le contenu de certaines boîtes et les en-têtes et/ou le contenu d'autres boîtes non demandées, parce que le serveur distant est invité à renvoyer tous les octets extraits du début de chaque segment de métadonnées qui contient les octets de boîte demandés, jusqu'au dernier octet de boîte demandé.

Le qualificateur "metareq-qualifier" prend la forme d'un caractère "/" suivi par un ou plusieurs fanions "g", "s", "w" et "a". Chaque fanion désigne un contexte à partir duquel les boîtes qui correspondent à la requête doivent être appelées. L'interprétation de chacun de ces contextes est fournie dans le Tableau C.2. Si plusieurs fanions sont fournis, la réunion des contextes correspondants doit être prise. Si aucun paramètre "metareq-qualifier" n'est fourni, la réunion des contextes "g", "s" et "w" doit être utilisée. Aux fins de la clarification, noter que les contextes "g", "s" et "w" sont mutuellement exclusifs, mais que leur réunion est généralement plus petite que le contexte universel "a".

Si le fanion de priorité est spécifié, alors le client va demander que les boîtes de type "box-type" qui correspondent à d'autres éléments de la requête soient acheminées avec une priorité plus élevée que les données d'image.

Pour tout type de boîte non spécifié dans la liste "req-box-prop", aucune donnée n'est demandée pour les boîtes de ce type.

Si le paramètre "metadata-only" est spécifié à la fin du champ de requête de métadonnées, le client va demander que la réponse du serveur distant consiste seulement de métadonnées, sans aucune donnée d'image ni aucun en-tête de flux codé, que des champs de requête de fenêtre de visualisation (tels que des champs de longueur de trame) aient ou non été utilisés. Pour les types de retour en flux JPP ou JPT, cela signifie que les messages JPIP renvoyés seront tous les messages de segment de métadonnées.

EXEMPLE 1: "metareq=[*]R31D4"

Dans ce cas, le serveur distant est invité à renvoyer le contenu entier de toutes les boîtes qu'il découvre dans le contenu du segment 31. Bien qu'une restriction sur la profondeur recherchée ait été spécifiée, le serveur distant doit ignorer cette restriction parce que le contenu de ces boîtes n'a pas été limité au moyen du paramètre "limit".

EXEMPLE 2: "metareq=[*:r,drep]R31D4"

Les caractères " *:r " signifient que le serveur distant a été invité à renvoyer des en-têtes de boîte pour toutes les boîtes contenues dans le segment de métadonnées 31 et dans tous les segments référencés par des génériques contenus dans ce segment, jusqu'à une profondeur de 4 niveaux à partir du contenu du segment 31, mais non compris le contenu de ces boîtes. La valeur supplémentaire "drep" du paramètre "req-box-prop" spécifie que le serveur distant est invité à renvoyer le contenu entier de toute boîte "drep" se trouvant dans le segment de métadonnées 31 ainsi que tous les segments référencés par des génériques dans ce segment, jusqu'à une profondeur de 4 niveaux à partir du contenu du segment 31.

EXEMPLE 3: "metareq=[drep]R31D4"

Dans ce cas, le serveur distant est encore invité à renvoyer le contenu entier de toute boîte "drep" qu'il découvre dans le contenu du segment 31 ou de tous les segments référencés par ce segment, jusqu'à une profondeur de quatre niveaux à partir du contenu du segment 31. Cependant, étant donné qu'aucune autre boîte n'a été spécifiée, le serveur distant est invité à n'envoyer qu'autant d'autres données que nécessaire afin de spécifier la position, dans la hiérarchie des fichiers, de toute boîte "drep" par rapport à la boîte contenue dans le segment de métadonnées 31.

Quelles que soient les spécifications de boîte fournies au moyen du champ de requête de métadonnées, le serveur distant peut envoyer d'autres données, soit parce qu'il a déterminé que ces autres données sont requises par le client afin de décoder ou interpréter les données d'image demandées, ou parce que le serveur distant a déjà subdivisé la cible logique en segments de données utilisant différents critères et parce que des données additionnelles doivent être envoyées afin d'offrir une vue cohérente et significative du segment de métadonnées pour cette cible logique.

Tableau C.2 – Fanions qualificateurs d'une requête de métadonnées

Fanion	Interprétation
"w"	Ce contexte de requête de métadonnées contient toutes les boîtes qui sont réputées associées à une certaine région spatiale d'image contenue dans un ou plusieurs flux codés qui appartiennent à la fenêtre de visualisation, lorsque la région spatiale, les résolutions et les composants d'image auxquels les boîtes se rapportent, sont en intersection avec ceux de la fenêtre de visualisation. Une telle association pourrait, par exemple, être établie par une boîte de type "asoc" contenue dans un fichier JPX.
"s"	Ce contexte de requête de métadonnées contient toutes les boîtes qui sont réputées associées à un ou plusieurs flux codés qui appartiennent à la fenêtre de visualisation, ou à un ou plusieurs des contextes demandés de flux codé (par exemple des couches de composition en format JPX ou des pistes vidéo en format MJ2), lorsque ces boîtes ne sont pas seulement associées à des régions spatiales particulières. Une telle association pourrait être établie par une boîte de type "asoc" contenue par exemple dans un fichier JPX.
"g"	Ce contexte de requête de métadonnées contient toutes les boîtes qui se rapportent à la fenêtre de visualisation demandée, compte tenu des flux codés demandés et de leurs contextes, à l'exclusion des boîtes qui sont incluses dans les contextes de requête de métadonnées "w" et "s".
"a"	Ce contexte de requête de métadonnées contient toutes les boîtes de la cible logique, sans exception (Note).
NOTE – Ce contexte de requête de métadonnées est approprié aux requêtes qui visent à rechercher la structure de fichier indépendamment de la fenêtre de visualisation.	

C.6 Champs de requête de limitation de données

C.6.1 Longueur maximale de réponse (len)

len = "len" "=" UINT

Ce champ spécifie une restriction sur la quantité de données que le client souhaite voir envoyées par le serveur distant en réponse à cette requête. L'unité doit être les octets. Si ce champ n'est pas présent, le serveur distant devrait envoyer des données d'image au client jusqu'à ce que toutes les données correspondantes aient été envoyées, qu'une limite de qualité ait été atteinte (voir C.6.2), ou que la réponse soit interrompue par l'arrivée d'une nouvelle requête qui ne comporte pas de champ de requête "Wait" avec une valeur égale à "yes" (voir C.7.2). Le client devrait utiliser la valeur "len=0" s'il nécessite des en-têtes de réponse sans aucune donnée de réponse.

C.6.2 Qualité (quality)

quality = "quality" "=" (1*2DIGIT / "100") ; 0 à 100

Ce champ peut servir à limiter la transmission de données à un certain niveau de qualité (compris entre 0 pour la plus basse qualité et 100 pour la plus haute) associé à l'image. Les limites de qualité sont difficiles à formuler de façon fiable et le serveur distant peut ignorer cette requête en répondant par une valeur de "-1" (voir D.2.16). Néanmoins, ce champ sert au client à offrir une indication de la qualité d'image maximale qui pourra être recherchée. Le facteur qualité peut tenter d'approcher la qualité ad hoc couramment utilisée afin de commander la compression JPEG. Le client devrait s'attendre que la longueur des données renvoyées ne décroîtra pas en proportion inverse de l'augmentation de la qualité, c'est-à-dire que l'augmentation de la valeur de qualité correspondra généralement à l'augmentation de la longueur des données renvoyées.

NOTE – Si un serveur distant prend en charge cette requête et si deux clients différents font des requêtes identiques à la même cible avec la même valeur de qualité, par exemple "quality=80", le serveur distant devrait avoir une politique cohérente d'implémentation lors du renvoi de données à partir de segments de données.

C.7 Champs de requête de commande de serveur distant

C.7.1 Alignement (align)

align = "align" "=" ("yes" / "no")

Ce champ spécifie si les données de réponse du serveur distant doivent être alignées sur des frontières naturelles. La valeur par défaut est "no". Si la valeur est "yes", tout message de flux JPT ou JPP acheminé en réponse à cette requête et traversant une quelconque "frontière naturelle" doit aboutir à toute "frontière naturelle" subséquente. Les frontières naturelles de chaque type de segment de données sont énumérées dans le Tableau C.3. Un message est considéré comme traversant une frontière naturelle s'il contient le dernier octet avant la frontière et le premier octet après. Par exemple, un segment de données de district traverse une frontière naturelle s'il contient le dernier octet d'un paquet et le premier octet du prochain paquet. Il y a lieu de bien noter que les messages de réponse alignés ne sont pas réellement tenus d'aboutir à une frontière naturelle à moins qu'ils ne la traversent. C'est-à-dire, par exemple, que la réponse peut comporter des paquets partiels à partir de districts, ce qui peut être nécessaire si une limite d'octet prioritaire empêche la remise de paquets complets.

Tableau C.3 – Frontières de verrouillage fondées sur le type de segment

Type de segment	Frontière naturelle
Segment de données de district	Fin d'un paquet (une seule frontière pour chaque couche de qualité).
Segment de données de pavé	Fin d'un pavé (une seule frontière pour chaque pavé).
Segment de données d'en-tête de pavé	Fin du segment (une seule frontière).
Segment de données d'en-tête principal	Fin du segment (une seule frontière).
Segment de métadonnées	Fin d'une boîte au niveau sommital du segment de données (une seule frontière pour chaque boîte).

C.7.2 Attente (wait)

wait = "wait" "=" ("yes" / "no")

Ce champ sert à indiquer si le serveur distant doit achever une réponse à la requête précédente. Si la valeur du champ est "yes", le serveur distant doit complètement satisfaire la précédente requête dans la ressource de canal spécifiée au moyen du champ d'identificateur de canal, avant de commencer à satisfaire cette requête.

Si la valeur de ce champ est "no", le serveur distant peut achever progressivement le traitement d'éventuelles requêtes précédentes dans la même ressource de canal (spécifiée au moyen du champ d'identificateur de canal) avant l'achèvement de la réponse et peut commencer à répondre à cette nouvelle requête. Dans ce contexte, un "achèvement progressif" implique que le serveur distant doit au moins terminer le message en cours.

La valeur par défaut de ce champ est "no".

C.7.3 Type de retour d'image (type)

type = "type" "=" 1#image-return-type

image-return-type = media-type / reserved-image-return-type

media-type = TOKEN "/" TOKEN *(";" parameter)

ISO/CEI 15444-9:2005 (F)

```
reserved-image-return-type = TOKEN * ( ";" parameter )
parameter = attribute "=" value
attribute = TOKEN
value = TOKEN
```

Ce champ sert à indiquer le type (ou les types) des données de réponse demandées. Un serveur distant non disposé à offrir l'un quelconque des types de retour demandés doit émettre une réponse d'erreur.

La valeur du champ de requête de type de retour d'image doit être soit un type de média (défini dans le document RFC 2046) ou un des types réservés de retour d'image définis dans le Tableau C.4.

Tableau C.4 – Types légaux de retour d'image

Type	Interprétation
"flux JPP"	Flux JPP tel que défini dans l'Annexe A. Un "flux JPP" peut facultativement être suivi par la chaîne ";ptype=ext", auquel cas le type de retour demandé est tel que tous les en-têtes de message de segment de données de district aient la forme étendue (voir A.2.2).
"flux JPT"	Flux JPT tel que défini dans l'Annexe A. Un "flux JPT" peut facultativement être suivi par la chaîne ";ttype=ext", auquel cas le type de retour demandé est tel que tous les en-têtes de message de segment de données de pavé aient la forme étendue (voir A.2.2).
"raw"	Le client va demander l'acheminement sans changement de la séquence entière des octets contenus dans la cible logique.
Autres valeurs	Champ réservé pour utilisation par l'ISO.

Si le champ de requête "type" est omis, le type de retour devrait être déterminé par un autre moyen.

Dans une session, c'est-à-dire une session dont les requêtes impliquent un champ de requête d'identificateur de canal, la valeur du paramètre de retour doit être conservée dans les réponses successives à des requêtes de données ou de métadonnées d'image qui correspondent à la même cible logique.

NOTE 1 – D'autres types de média iconographique (par exemple jp2, jpeg, tiff, png), si disponibles, peuvent être fournis par un serveur distant au titre d'un service de transcodage avec fonctionnalité de protocole JPIP.

NOTE 2 – Pour le type de retour de flux codé sans changement, les données de réponse devraient se composer de l'entité demandée en totalité. Un grand nombre des autres champs de requête du client n'auront alors aucune signification et seront ignorés par un serveur distant.

C.7.4 Vitesse d'acheminement (drate)

```
drate = "drate" "=" rate-factor
rate-factor = UFLOAT
```

Ce champ sert à spécifier la vitesse d'acheminement de divers flux codés. Si ce champ est fourni, le serveur distant doit remettre les données appartenant aux divers flux codés de la fenêtre de visualisation, conformément à un programme ordonné en séquence temporelle. Les flux codés qui appartiennent à la fenêtre de visualisation sont tous ceux qui ont été identifiés au moyen du champ de requête de flux codé et du champ de requête de contexte de flux codé, éventuellement sous-échantillonnés conformément au champ de requête de fréquence d'échantillonnage.

De façon à donner une signification à ce champ de requête, les informations temporelles doivent être associées aux divers flux codés se trouvant dans la fenêtre de visualisation. Si ces flux codés appartiennent à un fichier MJ2, les informations temporelles sont fournies par ce fichier. Le fichier MJ2 offre un mappage entre chaque flux codé et un temps nominal de lecture, qui est identifié ici comme étant le "temps de source".

Si les flux codés ne contiennent pas d'informations temporelles de source, mais que le champ de requête de fréquence d'échantillonnage soit présent, le serveur distant doit partir du principe que les flux codés se trouvant dans la fenêtre de visualisation ont des temps de source qui sont séparés par l'inverse de la valeur contenue dans le champ de requête de fréquence d'échantillonnage.

Si les flux codés n'ont pas d'informations temporelles de source et que le champ de requête de fréquence d'échantillonnage ne soit pas présent, le serveur distant doit partir du principe que les flux codés se trouvant dans la fenêtre de visualisation ont des temps de source qui sont séparés par exactement 1 s.

Le champ de requête de vitesse d'acheminement offre un facteur de normalisation entre les vitesses de remise et d'envoi à la source. Si le facteur de vitesse d'acheminement est indiqué comme ayant la valeur 1, le serveur distant devrait essayer de remettre les flux codés au client à la vitesse suggérée par leur temps de source, compte tenu du fait que ces temps de source pourraient ne pas nécessairement être réguliers. Plus généralement, si le facteur de vitesse

d'acheminement est F , le serveur distant devrait essayer de remettre les flux codés au client à une vitesse qui est F fois plus rapide que celle qui est suggérée par leur temps de source.

Si le serveur distant est dans l'impossibilité de remettre toutes les données correspondant à chaque flux codé à la vitesse demandée (par exemple à cause de contraintes sur la largeur de bande), ce serveur devrait remettre seulement une partie des données de chaque flux codé, de façon à éviter de violer la vitesse d'acheminement demandée. La portion des données de chaque flux codé qui n'est pas acheminée peut dépendre de la valeur du paramètre "view-window-pref" fournie dans un champ de requête de préférences du client (voir C.10.2). Si la préférence est "progressive" ou si aucune préférence de ce genre n'est identifiée, le serveur distant devrait essayer de remettre une qualité d'image maximale et uniforme dans la fenêtre de visualisation, sous réserve de la contrainte relative à la vitesse d'acheminement. Si une valeur du paramètre "view-window-pref" égale à "fullwindow" a été fournie, le serveur distant pourrait tronquer d'une autre façon la représentation associée à chaque flux codé. En tout état de cause, le comportement devrait être semblable à celui qui aurait résulté de l'émission par le client d'une succession de requêtes visant tour à tour chacun des flux codés considérés, à la vitesse d'acheminement.

Si le serveur distant est en mesure de remettre toutes les données correspondant à chaque flux codé, à la vitesse demandée, il devrait mettre à l'état de repos la connexion selon ce qui est nécessaire afin de garantir que la vitesse d'acheminement ne sera pas dépassée.

Si ce champ n'est pas fourni et si une valeur du paramètre "view-window-pref" égale à "fullwindow" n'a pas été spécifiée, le serveur distant devrait essayer de mettre en séquence les données correspondantes de façon à augmenter progressivement la qualité de tous les flux codés uniformément.

C.8 Champs de requête de gestion de cache

C.8.1 Modèles (model)

C.8.1.1 Généralités

```

model = "model" "=" 1#model-item
model-item = [codestream-qualifier ","] model-element
model-element = ["-"] bin-descriptor
bin-descriptor = explicit-bin-descriptor ; C.8.1.2
                / implicit-bin-descriptor ; C.8.1.3
codestream-qualifier = "[" 1$(codestream-range) "]"
codestream-range = first-codestream-id ["-"] [last-codestream-id]
first-codestream-id = UINT
last-codestream-id = UINT

```

Ce champ peut être utilisé en mode session ou par requêtes sans description d'état. Une requête en mode session est toute requête qui contient un champ d'identificateur de canal, car des canaux sont associés à une session gérée par le serveur distant. Le champ "model" contient un ou plusieurs descripteurs de segment, dont chacun désigne un segment de données, ou une étendue de segments de données, au sujet desquels des informations de cache sont signalées. Dans le cas des requêtes émises au cours d'une session, ces informations de cache servent à mettre à jour le modèle du serveur distant dans le cache du client. Il n'y a qu'un seul modèle de cache pour chaque cible logique associée à la session. Dans le cas d'une requête sans description d'état, le modèle du serveur distant du cache client est vide au début de la requête, mais est mis à jour par le champ "model" (s'il en existe un) avant que le serveur distant formule sa réponse. Toutes les informations de modèle de cache sont rejetées à l'issue du traitement d'une requête sans description d'état par le serveur distant.

Deux formes sont fournies pour les valeurs de descripteur de segment afin de faciliter l'échange efficace des informations de modèle de cache. Ces formes sont dites "explicite" et "implicite". Elles sont décrites dans le paragraphe suivant. Les clients peuvent émettre des requêtes en utilisant l'une ou l'autre forme et peuvent au besoin mélanger les deux formes de descripteur de segment dans un même champ de requête de type "model".

Si un descripteur de segment est précédé par un symbole "-", il est considéré comme étant soustractif. Sinon, il est considéré comme étant additif. Un descripteur de segment soustractif informe le serveur distant que les données correspondantes devraient être supprimées du modèle du serveur distant contenu dans le cache client. La suppression d'éléments du modèle de cache signifie que le serveur distant ne doit pas partir du principe que le client possède déjà ces éléments. Les valeurs de descripteur de segment sont traitées en ordre.

ISO/CEI 15444-9:2005 (F)

Un descripteur de segment additif (c'est-à-dire non précédé par le symbole "-") informe le serveur distant de données que le client possède déjà dans son cache. Le serveur distant peut ajouter ces informations à son modèle de cache et peut partir du principe que le client possède déjà les données indiquées.

Le champ "model" peut faire référence à des segments de données qui n'appartiennent pas à la fenêtre de visualisation recherchée, telle qu'identifiée par d'autres champs de requête (longueur de trame, taille de la région, décalage, etc.). Si cela se produit, la manipulation du modèle de cache pourrait ne pas affecter la réponse à la requête actuelle, mais pourrait néanmoins affecter la réponse à de futures requêtes (à moins que la requête ne soit sans description d'état).

Chaque fois que la liste des éléments de modèle contient un qualificateur de flux codé, tous les éléments de modèle subséquents doivent être ajoutés à – ou (selon le cas) soustraits de – tous les flux codés dont les identificateurs sont énumérés par le qualificateur de flux codé. Les qualificateurs de flux codé peuvent être répartis dans la liste afin de modifier progressivement l'ensemble des flux codés qui doivent être affectés par les éléments de modèle subséquents. Tout élément de modèle qui n'est pas précédé par un qualificateur de flux codé s'applique au premier flux codé demandé au moyen d'un champ de requête de flux codé. Si aucun champ de requête de flux codé n'est présent, les valeurs d'élément de modèle qui ne sont pas précédées par un qualificateur de flux codé doivent se rapporter au flux codé 0, qu'un champ de requête de contexte de flux codé soit ou non inclus. Si l'identificateur du dernier flux codé n'est pas présent, mais que le tiret de qualificateur le soit, alors cela doit signifier que l'identificateur du premier flux codé et tous les flux codés subséquents sont inclus.

Les requêtes émises au cours d'une session ne doivent pas comprendre de qualificateur de flux codé qui se rapporte à plus d'un seul flux codé.

NOTE 1 – Le serveur distant devrait essayer d'exploiter des instructions additives de manipulation du modèle de cache, mais il est libre d'en négliger certaines ou la totalité au prix éventuel de l'efficacité du transport. Les clients devraient être informés du fait que des serveurs distants pourraient être tout à fait susceptibles de négliger des instructions additives de manipulation du modèle de cache qui se rapportent à des segments de données appartenant à des flux codés qui ne seront pas desservis par la requête actuelle. Afin de supprimer de telles incertitudes lorsque de multiples flux codés sont impliqués, le champ de requête "mset" peut servir à déterminer l'ensemble des flux codés qui vont être modélisés.

NOTE 2 – La manipulation en mode session d'un modèle de cache du serveur distant affecte généralement la réponse, aussi bien à la requête actuelle qu'aux requêtes futures. En outre, tous les canaux d'une session qui sont associés à une même cible logique partagent le même modèle de cache. Ainsi, les champs "model" contenus dans des requêtes qui arrivent en utilisant un canal (champ d'identificateur de canal) peuvent affecter la réponse à des requêtes qui arrivent au moyen d'un canal différent. Il est important de noter que les requêtes qui utilisent différents canaux JPIP (différentes valeurs d'identificateur de canal) peuvent arriver de façon asynchrone au serveur distant, si des canaux TCP distincts servent à transporter la requête soit directement à partir du client ou indirectement par un serveur intermédiaire. Les clients devraient entreprendre toute action nécessaire afin de garantir que les instructions de manipulation de leur modèle de cache restent significatives à la lumière de ces considérations.

C.8.1.2 Forme explicite

```
explicit-bin-descriptor = explicit-bin
                          [ ":" (number-of-bytes / number-of-layers) ]

explicit-bin = codestream-main-header-bin
              / meta-bin
              / tile-bin
              / tile-header-bin
              / precinct-bin

number-of-bytes = UINT

number-of-layers = %x4c UINT ; "L"

codestream-main-header-bin = %x48 %x6d ; "Hm"

meta-bin = %x4d bin-uid ; "M"

tile-bin = %x54 bin-uid ; "T"

tile-header-bin = %x48 bin-uid ; "H"

precinct-bin = %x50 bin-uid ; "P"

bin-uid = UINT / "*"

```

Les valeurs de descripteur de segment qui se rapportent explicitement à des segments de données sont des types suivants: M (segment de métadonnées), Hm (segments de données d'en-tête principal), H (segments de données d'en-tête de pavé), P (segments de données de district) ou T (segments de données de pavé). Les descripteurs de segment explicites identifient le(s) segment(s) de données correspondant(s) dans les flux codés applicables, en utilisant soit un identificateur unique évalué par un entier ou un caractère générique: "*". La seule exception à cela est le segment de données d'en-tête principal de flux codé, dont le descripteur de segment est "Hm". Pour toutes les autres

classes de segment de données, l'identificateur unique est identique à la valeur communiquée par l'identificateur-dans-la-classe qui est contenu dans les en-têtes de message en flux JPP et/ou JPT (voir Annexe A).

Le caractère générique, "*", ne doit être utilisé que dans les requêtes sans description d'état. Lorsqu'il est utilisé, le descripteur de segment se rapporte simultanément à tous les segments de données de la classe correspondante (métadonnées, district, en-tête de pavé ou pavé), appartenant à la fenêtre de visualisation.

Chaque descripteur de segment peut être qualifié par un certain nombre d'octets. Un descripteur de segment additif qui est qualifié par le nombre d'octets, *B*, indique que le client possède déjà dans son cache au moins les *B* premiers octets du segment indiqué de données dans son cache; le serveur distant peut ajouter les *B* premiers octets du segment de données à son modèle de cache. Un descripteur de segment soustractif qui est qualifié par le nombre d'octets, *B*, indique que le client possède au plus les *B* premiers octets du segment indiqué de données; le serveur distant doit supprimer de son modèle de cache tous les octets suivant les *B* premiers octets du segment de données.

EXEMPLE 1: Un descripteur qualifié de segment soustracteur tel que "-P23:10" signifie que le serveur distant devrait supprimer de son modèle de cache tous les octets sauf les 10 premiers du segment de données de district 23. Cela n'implique pas que le client possède les 10 premiers octets du segment de données de district 23 dans son cache et le serveur distant ne devrait pas le supposer en ajoutant ces octets à son modèle de cache s'ils n'étaient pas déjà présents.

Les descripteurs de segment de district peuvent en variante être qualifiés par un certain nombre de couches. Un descripteur de segment additif qui est qualifié par le nombre de couches, *L*, indique que le client possède déjà au moins les *L* premières couches (*L* premiers paquets) du district indiqué; le serveur distant peut ajouter à son modèle de cache les octets correspondant à ces couches. Un descripteur de segment à district soustractif qui est qualifié par le nombre de couches, *L*, indique que le client possède au plus les *L* premières couches (*L* paquets) du district indiqué; le serveur distant doit supprimer de son modèle de cache les octets correspondant à toutes les couches éventuellement subséquentes de ce district.

Un descripteur de segment sans qualificateur de nombre d'octets ou de nombre de couches désigne la totalité explicite du segment de données.

EXEMPLE 2: "model=M0,Hm,H7:20,P3" signifie que le client possède dans son cache au moins la totalité du segment de métadonnées 0, la totalité de l'en-tête principal de flux codé, les 20 premiers octets de l'en-tête de pavé 7 et tous les octets du district 3.

EXEMPLE 3: "model=P3:256,P5:L2,-P6:20" signifie que le client possède au moins les 256 premiers octets du district 3 et les deux premières couches (paquets) du district 5, mais (au plus) il ne possède rien au-delà du 20ème octet du district 6 (il ne peut pas avoir les 20 premiers octets non plus).

EXEMPLE 4: "model=M*,-M5,-H*,-P*:L3" signifie que le client possède (ou est disposé à laisser le serveur distant estimer qu'il possède) tous les segments de métadonnées sauf le segment de métadonnées 5, aucun segment de données d'en-tête de pavé qui se rapporte à la fenêtre de visualisation et au plus les 3 premières couches de tout district qui appartient à la fenêtre de visualisation. Noter que les génériques utilisés ici ne sont admissibles que quand l'instruction "model" apparaît dans une requête sans description d'état.

EXEMPLE 5: "model=[30-200],Hm,H*,M*,P0,[0-29],-Hm,-H*,-M*,-P*" signifie que le client possède tous les en-têtes et toutes les métadonnées, plus le segment de données de district 0 des flux codés 30 à 200 inclus, mais qu'il a supprimé, des 30 premiers flux codés, tous les segments de données d'en-tête, de métadonnées et de district.

C.8.1.3 Forme implicite

```
implicit-bin-descriptor = 1*implicit-bin [":" number-of-layers]
implicit-bin = implicit-bin-prefix (data-uid / index-range-spec)
implicit-bin-prefix = %x74      ; t -- tile
                       / %x63      ; c -- component
                       / %x72      ; r -- resolution level
                       / %x70      ; p -- position

index-range-spec = first-index-pos "-" last-index-pos
first-index-pos = UINT
last-index-pos = UINT
data-uid = UINT / "*"

```

Les valeurs de descripteur de segment qui se rapportent implicitement à des segments de données sont des types suivants: t (pavé auquel le district appartient), c (composant d'image auquel le district appartient), r (niveau de résolution de l'élément de pavé auquel le district appartient) ou p (position du district dans sa résolution d'élément de

ISO/CEI 15444-9:2005 (F)

pavé). Des descripteurs de segment implicites sont utilisés afin d'identifier des segments de données de district au moyen des indices. Tous les indices doivent commencer à 0. Un indice de niveau de résolution égal à 0, r0, se rapporte au plus bas niveau de résolution (sous-bande LL) de l'élément de pavé. Les indices de position, p, vont de gauche à droite et de haut en bas lors de la progression de résolution d'un élément de pavé, sous forme de balayage, comme décrit dans la Rec. UIT-T T.800 | ISO/CEI 15444-1.

Dans les requêtes sans description d'état, l'un quelconque ou la totalité des spécificateurs de segment implicite de pavé, de composant, de niveau de résolution ou de position peut être remplacé par l'étendue des indices ou par le caractère générique, "*". Dans un cas comme dans l'autre, le descripteur de segment est étendu de façon à inclure toutes les valeurs de l'étendue des indices appartenant à la fenêtre de visualisation. Aucune de ces options ne doit être utilisée dans le cas des requêtes émises au cours d'une session.

Dans les requêtes sans description d'état, l'un quelconque ou la totalité des indices de pavé, de composant, de niveau de résolution ou de position peut également être remplacé par une même étendue d'indices. La valeur du paramètre "first-index-pos" contenue dans une spécification "index-range-spec" donne le premier indice contenu dans une étendue. La valeur du paramètre "last-index-pos" donne le dernier indice contenu dans l'étendue et doit être supérieure ou égale à la valeur du paramètre "first-index-pos". Les deux indices spécifiés sont inclus. Le paramètre "last-index-pos" ne peut pas être omis. Si une étendue d'indices de pavé ("t") est indiquée, elle se rapporte à une série tabulaire de pavés de forme rectangulaire, dont le coin supérieur gauche a la valeur du paramètre "first-index-pos" et dont le coin inférieur droit a la valeur du paramètre "last-index-pos". De même, si une étendue d'indices de position ("p") est indiquée, cette étendue se rapporte à une série rectangulaire de positions de district dont les coins supérieur gauche et inférieur droit sont indiqués respectivement par les valeurs des paramètres "first-index-pos" et "last-index-pos". Comme les génériques, les étendues ne doivent pas être utilisées dans les requêtes émises au cours d'une session.

Les descripteurs implicites de segment de district peuvent être qualifiés par un certain nombre de couches dont la syntaxe et l'interprétation sont identiques à celles des descripteurs de segment de district explicitement qualifiés en couche, décrits plus haut.

EXEMPLE 1: "model=t0c2r3p4:L5" indique que le client possède les 5 premiers paquets du 5ème district en séquence, du quatrième niveau de résolution, du troisième composant, du pavé 0.

EXEMPLE 2: "model=t10r0,t*r1:L4" signifie que le client possède toutes les couches de l'indice de pavé 10 au niveau de résolution 0 et les 4 premières couches de tous les pavés appartenant à une fenêtre de visualisation au niveau de résolution 1. Noter que le générique n'est approprié que dans le cas des requêtes sans description d'état.

EXEMPLE 3: "model=t0-10:L2" indique que le client possède les 2 premières couches à partir des pavés 0 à 10. Noter que l'étendue n'est appropriée que dans le cas des requêtes sans description d'état.

EXEMPLE 4: "model=t*r0-2:L4" indique que le client possède les 4 premières couches à partir des niveaux de résolution 0 à 2 de tous les pavés appartenant à la fenêtre de visualisation. Noter que le générique et l'étendue ne sont appropriés que dans le cas des requêtes sans description d'état.

C.8.2 Résumé des options de descripteur de cache (pour information)

Tableau C.5 – Résumé des options de descripteur de cache

Type de forme	Générique		Etendue d'indices	Nombre de couches (par exemple " :L3 ")	Nombre d'octets (par exemple " :256 ")
	sans description d'état	en mode session			
Forme explicite	Autorisée	Non autorisée	Non autorisée	Autorisée	Autorisée
Forme implicite	Autorisée	Non autorisée	Autorisée seulement sans description d'état	Autorisée	Non autorisée

C.8.3 Modèle d'élément de pavé utilisant des flux JPT (tpmodel)

```

tpmodel = "tpmodel" "=" 1#tpmodel-item
tpmodel-item = [codestream-qualifier "," ] tpmodel-element
tpmodel-element = ["-"] tp-descriptor
tp-descriptor = tp-range / tp-number
tp-range = tp-number "-" tp-number

```

```

tp-number = tile-number "." part-number
tile-number = UINT
part-number = UINT

```

Ce champ peut servir à indiquer des éléments de pavé spécifiques que le client voudrait ajouter ou soustraire à partir du modèle de cache du serveur distant. Comme le champ "model", ce champ peut être utilisé aussi bien en mode session que dans les requêtes sans description d'état. Dans ce dernier cas, le modèle de cache est vide au début de la requête et ne persiste pas entre les requêtes, tout en offrant encore un mécanisme utile pour l'identification des éléments d'image qui sont déjà dans le cache du client.

Si un descripteur d'élément de pavé est précédé par un caractère "-", il est considéré comme étant soustractif. Sinon il est additif. Un descripteur d'élément de pavé additif indique que le client possède déjà l'élément de pavé ou l'étendue d'éléments de pavé dans son cache; le serveur distant peut alors ajouter ces éléments à son modèle de cache. Un descripteur d'élément de pavé soustractif indique que le client ne possède pas l'élément de pavé ou l'étendue d'éléments de pavé dans son cache; le serveur distant doit alors supprimer ces éléments de son modèle de cache.

La première valeur contenue dans le numéro d'élément de pavé est l'indice de pavé (à partir de 0); la seconde valeur est le numéro d'élément (à partir de 0) dans le pavé. Une étendue "tp-range" est considérée comme contenant indépendamment des pavés à partir du premier numéro de pavé jusqu'au second numéro de pavé et comme contenant des éléments de pavés à partir du premier numéro d'élément de pavé jusqu'au second numéro d'élément de pavé. Ainsi, 4.0-5.1 contient les éléments de pavé 4.0, 4.1, 5.0 et 5.1, mais non les éléments 4.2 ou 5.2.

Les champs de requête "tpmodel" et "model" peuvent apparaître tous les deux dans une requête isolée. Dans ce cas cependant, le serveur distant doit refléter les effets du champ "model" sur son modèle de cache avant traitement du champ "tpmodel".

Les valeurs du qualificateur de flux codé peuvent être réparties dans la liste d'éléments de type "tpmodel" afin de modifier l'ensemble des flux codés auquel les éléments de type "tpmodel" suivants sont applicables, selon exactement les mêmes principes que pour le champ de requête de type "model".

NOTE – A la différence du champ de requête de type "model", les étendues d'éléments de pavé et les étendues de flux codés (dans les qualificateurs de flux codé) sont les unes et les autres autorisées dans le champ de requête "tpmodel", que ce champ apparaisse dans une requête en mode session ou dans une requête sans description d'état.

EXEMPLE 1: "tpmodel=4.0,4.1,5.0-6.2" indique que le client possède déjà dans son cache les deux premiers éléments du pavé 4 et les 3 premiers éléments des pavés 5 et 6.

EXEMPLE 2: "tpmodel=-4.0-6.254" indique que le client n'a aucun élément du pavé 4, 5 ou 6 dans son cache.

EXEMPLE 3: "tpmodel=3.0,[131-133],4.0,[100],-0.0-65534.254" indique que le client possède l'élément de pavé 0 du pavé 3 à partir du flux codé 0 référencé dans la requête, plus l'élément de pavé 0 du pavé 4 à partir de chacun des flux codés 131 à 133 inclus et qu'il va supprimer de son cache tous les éléments de pavé du flux codé 100.

C.8.4 Besoin de requêtes sans description d'état (need)

```

need = "need" "=" 1#need-item
need-item = [codestream-qualifier "," ] bin-descriptor

```

Ce champ ne peut apparaître que dans les requêtes sans description d'état, c'est-à-dire qui ne contiennent pas de champ de requête d'identificateur de canal. Il a la même syntaxe que le champ de requête de modèle, sauf que les descripteurs de segment ne doivent pas être précédés par un symbole "-". Le champ de requête "need" ne doit pas apparaître dans la même requête qu'un champ de requête de type "model" ou "tpmodel".

Le champ de requête "need" indique l'ensemble des segments de données (ou de suffixes de segment de données) qui peuvent intéresser le client. Le serveur distant n'a pas besoin d'envoyer d'informations qui n'ont pas d'intérêt potentiel. Quelle que soit la dimension possible de l'ensemble de segments de données potentiellement intéressants, le serveur distant ne devrait envoyer que les informations qui se rapportent aux champs de requête de fenêtre de visualisation ou au champ de requête de métadonnées.

L'effet du champ "need" sur la requête du serveur distant peut être expliqué au moyen du concept de modèle de cache temporaire, qui est initialisé (vide) immédiatement avant que la requête soit traitée et qui est rejeté une fois la réponse produite. Si un champ "need" apparaît dans la requête, tous les segments de données possibles sont ajoutés au modèle de cache, après quoi tous les éléments référencés par les descripteurs de segment dans le champ "need" sont supprimés du modèle de cache. Le serveur distant traite alors la fenêtre de visualisation demandée en utilisant ce modèle de cache afin de déterminer les éléments qui n'ont pas besoin d'être envoyés au client.

ISO/CEI 15444-9:2005 (F)

Les qualificateurs de flux codé peuvent être répartis dans la liste des descripteurs de segment afin de modifier l'ensemble de flux codés auquel les descripteurs de segment suivants seront applicables, selon exactement les mêmes principes que pour les champs de requête "model" et "tpmodel".

EXEMPLE 1: "need=M1,H0:20,P0" signifie que le client a besoin de tous les segments de métadonnées 1, des données à partir du 20ème octet du segment de données d'en-tête de pavé 0, et de la totalité du segment de données de district 0.

EXEMPLE 2: "need=P1:256,P5:L2" signifie que le client a besoin des données situées au-delà du 256ème octet (ou à partir de l'octet 256) du segment de données de district 1, et des couches situées au-delà de la 2ème couche du segment de données de district 5.

EXEMPLE 3: "need=H*,P*:L3" signifie que le client a besoin de tous les segments de données d'en-tête de pavé appartenant à la fenêtre de visualisation, et des couches situées au-delà de la 3ème couche de tous les segments de données de district appartenant à la fenêtre de visualisation.

EXEMPLE 4: "need=t10r0,t*r1:L4" signifie que le client a besoin de toutes les couches de l'indice de pavé 10 au niveau de résolution 0, et des couches situées au-delà de la 4ème couche de tous les pavés appartenant à la fenêtre de visualisation au niveau de résolution 1.

EXEMPLE 5: "need=*r0-2:L4" signifie que le client a besoin de toutes les couches à partir de la couche 4 de tous les segments de données de district aux niveaux de résolution 0 à 2 (0, 1 et 2) dans tous les pavés et composants appartenant à la requête de fenêtre de visualisation.

EXEMPLE 6: "need=[120-131],r0,[140;143-145],r0-1" signifie que le client a besoin du niveau de résolution 0 des flux codés 120 à 131 inclus, et des niveaux de résolution 0 et 1 des flux codés 140 et 143 à 145 inclus.

C.8.5 Besoin d'élément de pavé dans le cas des requêtes sans description d'état (tpneed)

```
tpneed = "tpneed" "=" 1#tpneed-element
```

```
tpneed-element = [codestream-qualifier "," ] tp-descriptor
```

Ce champ ne peut apparaître que dans les requêtes sans description d'état, c'est-à-dire qui ne contiennent pas de champ de requête d'identificateur de canal. Il a la même syntaxe que le champ de requête "tpmodel", sauf que les descripteurs d'élément de pavé ne doivent pas être précédés par un symbole "-". Le champ de requête "tpneed" ne doit pas apparaître dans la même requête qu'un champ de requête "model" ou "tpmodel".

Le champ de requête "tpneed" indique l'ensemble des éléments de pavé qui peuvent intéresser le client. Le serveur distant n'a pas besoin d'envoyer d'informations qui n'ont pas d'intérêt potentiel. Quelle que puisse être la grandeur de l'ensemble des éléments de pavé d'intérêt, le serveur distant ne devrait envoyer que les informations qui se rapportent aux champs de requête de fenêtre de visualisation ou au champ de requête de métadonnées.

L'effet du champ "tpneed" contenu dans la requête du serveur distant peut être expliqué au moyen du concept de modèle de cache temporaire, qui est initialisé (vide) immédiatement avant que la requête soit traitée et qui est rejeté une fois la réponse produite. Si un champ "tpneed" apparaît dans la requête, tous les éléments de pavé et segments de données possibles sont ajoutés au modèle de cache, après quoi tous les éléments référencés par les descripteurs de segment dans le champ "need" et tous les pavés référencés dans le champ "tpneed" sont supprimés du modèle de cache. Le serveur distant traite alors la fenêtre de visualisation demandée en utilisant ce modèle de cache afin de déterminer les éléments qui n'ont pas besoin d'être envoyés au client.

Les qualificateurs de flux codé peuvent être répartis dans la liste d'éléments de pavé afin de modifier l'ensemble des flux codés auquel les éléments de pavé suivants sont applicables, selon exactement les mêmes principes que pour les champs de requête "model" et "tpmodel".

C.8.6 Ensemble de modèles dans le cas des requêtes émises au cours d'une session (mset)

```
mset = "mset" "=" 1#sampled-range
```

Ce champ remplit deux fonctions. En premier lieu, il informe le serveur distant de l'ensemble des flux codés pour lesquels le client est disposé à mettre en mémoire cache des données acheminées par le serveur distant. En deuxième lieu, il offre un mécanisme permettant au client de s'informer sur les flux codés pour lesquels le serveur distant est disposé à modéliser le cache du client. Spécifiquement, si l'ensemble d'indices de flux codé fourni dans une requête "mset" diffère d'une façon ou d'une autre de l'ensemble de flux codés pour lequel le serveur distant est actuellement disposé à offrir une modélisation de cache, ce serveur doit offrir un en-tête de réponse d'ensemble de modèles, comme indiqué en D.2.18.

La chaîne paramétrique du champ de requête "mset" se compose d'une liste séparée par des virgules d'étendues d'indices de flux codé, éventuellement sous-échantillonnés, suivant les conventions décrites en C.4.6 à propos du champ de requête de flux codé.

En plus des flux codés mentionnés dans la requête "mset", le serveur distant peut également offrir un modèle de cache pour tous les flux codés associés à sa réponse à la requête actuelle. Il s'agit de l'ensemble des flux codés identifiés par la requête du client (voir les champs de requête de flux codé et de contexte de flux codé en C.4.7), à moins que le serveur distant n'indique qu'un ensemble réduit de flux codés, au moyen d'un en-tête de flux codé de réponse (voir D.2.9). Si aucun champ de requête "mset" n'est fourni, le client ne devrait pas partir du principe que le serveur distant va fournir un modèle de cache pour tout flux codé autre que ceux qui sont associés à sa réponse; cependant, il peut modéliser d'autres flux codés. Si un champ de requête "mset" est indiqué, le serveur distant doit rejeter toutes les informations de modèle de cache qu'il possède pour tous les flux codés autres que ceux qui sont mentionnés, soit dans la requête "mset" ou dans l'ensemble de flux codés associé à ses données de réponse. En outre, les effets de toute manipulation du modèle de cache au moyen du champ de requête "model" ou "tpmodel" doivent être limités à ces seuls flux codés.

Le serveur distant peut, à sa discrétion, réduire le nombre de flux codés qui sont contenus dans le champ "mset", auquel cas il doit fournir un en-tête de réponse "mset" identifiant l'ensemble réel des flux codés qui doivent être modélisés; en outre, cet ensemble de flux codés modélisés doit au moins comprendre tous les flux codés qui sont associés aux données de réponse du serveur distant (celles qui ont été demandées par la requête du client, ou qui ont été identifiées par l'en-tête de réponse de flux codé du serveur, le cas échéant). Dans ce cas, ces instructions s'appliquent aux flux codés contenus dans le champ "mset" et identifiés par le serveur distant, lequel ne peut pas identifier un plus grand ensemble de flux codés que mentionné dans la requête "mset" du client, combinés avec les flux codés qui sont associés aux données de réponse du serveur distant.

Noter que le serveur distant peut changer son champ "mset" de requête en requête, de façon que les clients qui doivent garder trace de l'ensemble "mset" du serveur distant et/ou lui imposer des contraintes serrées puissent choisir d'inclure un champ de requête "mset" dans chaque requête.

C.9 Paramètres de requête de téléchargement amont

C.9.1 Téléchargement vers l'amont (upload)

```
upload = "upload" "=" upload-type
upload-type = image-return-type ; C.7.3
```

Ce champ spécifie que le client va télécharger en amont de nouvelles images ou métadonnées vers le serveur distant. La valeur du paramètre "upload-type" peut être l'une quelconque des valeurs valides "image-return-type" qui pourraient être utilisées avec le champ de requête "type". Voir en Annexe E de plus amples informations sur les données de téléchargement amont.

C.10 Champs de requête de capacité et de préférence du client

C.10.1 Capacité du client (cap)

```
cap = "cap" "=" 1#capability-group
capability-group = processing-capability
                  / depth-capability
                  / config-capability
processing-capability = compatibility-capability
                      / vendor-capability
compatibility-capability = "cc." compatibility-code
vendor-capability = "vc." vendor-code [":" vendor-value]
vendor-code = 1*(LOWER / DIGIT / "." / "-")
vendor-value = TOKEN
depth-capability = "depth:" UINT
config-capability = "config:" UINT
```

Ce champ spécifie les capacités du client. Pour les requêtes en mode session (celles qui comprennent un champ de requête d'identificateur de canal), tous les champs de capacité transmis par le client ne doivent affecter que le canal associé à la requête et doivent être considérés comme rémanents. Les capacités n'ont pas besoin d'être retransmises par le client dans le cas de requêtes ultérieures sur le même canal.

Quand un nouveau canal est créé à partir d'un canal existant, ses capacités de client sont héritées. Dans le cas des requêtes sans description d'état et dans celui des requêtes émises dans un canal dont les capacités n'ont jamais été spécifiées ou héritées, les capacités de client peuvent être déterminées ou prévues autrement. Les capacités associées à

un canal peuvent être modifiées par inclusion d'un champ de requête de capacités de client dans une requête quelconque.

Si le champ de requête de capacités de client désigne une ou plusieurs des options de l'élément "processing-capability", le serveur distant doit partir du principe que le client ne possède aucune des autres options de l'élément "processing-capability" qui pourraient avoir été mentionnées. Si aucune option de l'élément "processing-capability" n'est fournie dans les champs de requête de capacités de client, le serveur distant doit continuer à utiliser toute information antérieure qu'il possédait concernant les capacités de traitement. Les options de l'élément "processing-capability" définies par la présente Recommandation | Norme internationale sont décrites dans le Tableau C.6.

Tableau C.6 – Capacités légales du paramètre "processing-capabilities"

Capacité	Signification
compatibility-capability	Le client prend en charge tous les fichiers qui contiennent l'élément "compatibility-code" dans la liste de compatibilités contenue dans la boîte de type de fichier. Par exemple, afin d'indiquer que le client prend en charge tous les fichiers JP2, le client transmettra le code "cc.jp2_" dans un champ de requête de capacité. Une valeur de "compatibility-code" égale à "jp2c" doit servir à indiquer la prise en charge des flux JPEG 2000 non traités.
vendor-capability	Le client prend en charge la capacité de vendeur définie par le code de vendeur (vendor-code), lequel doit être une chaîne spécifiant le nom du domaine de retour du vendeur qui a défini l'élément de service, suivi par le nom de l'élément de service du vendeur. Par exemple, si la chaîne "exemple.com" définit un élément de service appelé "distance", alors la valeur du code de vendeur pour cet élément de service doit être "com.exemple.distance". L'élément "vendor-value" spécifie une valeur facultative, comme défini par l'élément de service facultatif du vendeur.

Si un paramètre "depth-capability" est fourni, cet élément indique la profondeur maximale de l'échantillon en termes de bits (précision) à laquelle le client est en mesure d'exploiter l'imagerie décompressée. Si le client prend en charge différentes profondeurs de bits pour différents composants d'image, ce champ doit spécifier la profondeur de bits du composant pour lequel le client possède la plus grande capacité de profondeur de bits.

NOTE 1 – Si un client prend en charge une profondeur de 12 bits pour la luminance et de 8 bits pour la chrominance, la valeur de l'élément "depth-capability" doit être égale à 12.

NOTE 2 – Les clients possédant la capacité de manipuler seulement *N* bits par échantillon resteront généralement en mesure de manipuler des flux codés dont le marqueur SIZ indique une profondeur de bits beaucoup plus grande que *N*. Cependant, ce fanion peut être utilisé par le serveur distant afin de déterminer une façon appropriée de remettre les données d'image demandées.

Si un paramètre "config-capability" est fourni, il doit être dans l'étendue de 0 à 255, représentant un mot de 8 bits dont chaque élément binaire est interprété comme un fanion de configuration. L'interprétation des fanions de configuration est fournie dans le Tableau C.7.

Tableau C.7 – Valeurs légales du paramètre "config-capability"

Valeur	Signification
1xxx yyyy	Le client est capable de traiter les données d'image en couleur.
0xxx yyyy	Le client n'est pas capable de traiter les données d'image en couleur et souhaite que le serveur distant transmette toutes les régions d'image demandées en échelle de gris.
x1xx yyyy	Le client possède un dispositif de pointage pour l'interaction avec l'utilisateur final.
x0xx yyyy	Le client ne possède pas de dispositif de pointage pour l'interaction avec l'utilisateur final.
xx1x yyyy	Le client possède un clavier pour l'interaction avec l'utilisateur final.
xx0x yyyy	Le client ne possède pas de clavier pour l'interaction avec l'utilisateur final.
xxx1 yyyy	Le client possède des capacités de sortie acoustique.
xxx0 yyyy	Le client ne possède pas de capacités de sortie acoustique.
Autres valeurs	Champ réservé pour utilisation par l'ISO.

Une valeur binaire de "x" dans le Tableau C.7 indique que la valeur spécifiée inclut les cas où ce bit est réglé soit à "1" soit à "0". Les bits indiqués par "y" ne sont pas utilisés par la présente Recommandation | Norme internationale; ils doivent être réglés à 0 par les clients et être ignorés par les serveurs distants.

C.10.2 Préférences du client (pref)

C.10.2.1 Généralités

```
pref = "pref" "=" 1#(related-pref-set ["/r"])
```

```

related-pref-set = view-window-pref          ; C.10.2.2
                  / colour-meth-pref         ; C.10.2.3
                  / max-bandwidth           ; C.10.2.4
                  / bandwidth-slice         ; C.10.2.5
                  / placeholder-pref        ; C.10.2.6
                  / codestream-seq-pref     ; C.10.2.7
                  / other

```

other = TOKEN

Ce champ spécifie les préférences du client concernant le comportement du serveur distant. Dans le cas des requêtes en mode session (qui comprennent un champ de requête d'identificateur de canal), tout champ de préférence transmis par le client ne doit affecter que le canal associé à la requête et doit être considéré comme rémanent. Les préférences n'ont pas besoin d'être retransmises par le client dans le cas de requêtes ultérieures sur le même canal. Chaque préférence doit apparaître au plus 1 fois dans un même champ de requête de préférence.

Quand un nouveau canal est créé à partir d'un canal existant, ses préférences sont héritées. Dans le cas des requêtes sans description d'état et dans celui des requêtes émises dans un canal dont les préférences n'ont jamais été spécifiées ou héritées, les préférences du client peuvent être déterminées ou prévues autrement. Si le client souhaite modifier ses préférences, il doit renvoyer entièrement le paramètre "related-pref-set" qui lui a été affecté.

Sauf indication contraire, chaque paramètre "related-pref-set" spécifie une liste de jetons individuels de préférence, dans l'ordre décroissant des préférences. Si possible, le serveur distant doit respecter les préférences du client identifiées par ce champ de requête. Si un paramètre "related-pref-set" est suivi par le modificateur "/r" (requis), le serveur distant doit soit prendre en charge une des préférences énumérées dans ce paramètre "related-pref-set", ou répondre par une erreur. Dans le dernier cas, le serveur distant doit renvoyer un en-tête de réponse de préférence indisponible qui désigne tout paramètre "related-pref-set" qui avait le modificateur "/r" mais qui n'a pas pu être pris en charge. Voir en D.2.20 plus de détails sur l'en-tête de réponse de préférence indisponible.

Par exemple, considérons la requête suivante de préférences du client:

```
pref=fullwindow/r,color-ricc:2;color-icc
```

Cette requête de préférence nécessite que le serveur distant renvoie toute la fenêtre de visualisation demandée, quelle que puisse être sa grandeur (voir en C.10.2.2 une analyse de la préférence "fullwindow"). Comme le modificateur "/r" a été utilisé, le serveur distant devrait renvoyer une réponse d'erreur à moins qu'il ne soit en mesure de prendre en charge cette préférence. En outre, le client préfère utiliser des profils ICC limités plutôt qu'arbitraires, à condition que le profil ICC limité soit au moins de "qualité exceptionnelle". Voir en C.10.2.3 une analyse des préférences en termes d'espace chromatique.

Un serveur distant doit ignorer toute valeur du paramètre "related-pref-set" qu'il ne peut pas interpréter et qui n'est pas immédiatement suivie par "/r". Si la valeur non interprétée est suivie par "/r", alors le serveur distant doit renvoyer l'en-tête de réponse de préférence indisponible, en indiquant la préférence qu'il n'est pas en mesure d'assurer.

Les valeurs du jeton "other" sont réservées pour utilisation par l'ISO.

C.10.2.2 Préférences relatives à la manipulation de la fenêtre de visualisation

```
view-window-pref = "fullwindow" / "progressive"
```

La présente Recommandation | Norme internationale définit deux options afin de spécifier le comportement du serveur distant si la requête ne peut pas être desservie exactement comme déclaré, conformément à un ordonnancement progressif de la qualité des données de réponse. Ces deux options sont spécifiées dans le Tableau C.8.

Tableau C.8 – Préférences relatives à la manipulation de la fenêtre de visualisation

Option	Signification
"fullwindow"	Le serveur distant doit donner suite aux paramètres de requête de fenêtre de visualisation mais est autorisé à renvoyer les données dans un ordre de qualité non progressif.
"progressive"	Le serveur distant peut modifier les paramètres de requête de fenêtre de visualisation afin de conserver les caractéristiques de progression de la qualité des données de réponse. Si le serveur distant ne modifie pas les paramètres de requête de fenêtre de visualisation, la fenêtre de visualisation modifiée devrait être un sous-ensemble de la fenêtre de visualisation initialement demandée.

Si aucune valeur "fullwindow" ou "progressive" n'est spécifiée dans le champ de requête de préférences du client, le serveur distant doit partir du principe que la préférence du client est "progressive".

ISO/CEI 15444-9:2005 (F)

Noter que l'interprétation d'une remise "progressive" peut être affectée par la présence d'un champ de requête de vitesse d'acheminement, comme expliqué en C.7.4.

C.10.2.3 Préférence relative à la méthode de détermination de l'espace chromatique

```
color-meth-pref = 1$(color-meth [":" meth-limit])  
color-meth = "color-enum" / "color-ricc" / "color-icc" / "color-vend"  
meth-limit = UINT
```

La présente Recommandation | Norme internationale définit quatre options qui spécifient quelles formes de données de spécification de l'espace chromatique devraient être renvoyées par le serveur distant. Un même fichier JPEG 2000 peut contenir de multiples spécifications de l'espace chromatique pour un même flux codé ou pour une même couche de composition. Cela permet à un rédacteur de fichier d'offrir la spécification optimale de l'espace chromatique tout en continuant à fournir des solutions interopérables.

Cependant, tous les lecteurs ne prendront pas en charge toutes les méthodes de détermination de l'espace chromatique. Par ailleurs, les données fournies pour certaines méthodes de détermination de l'espace chromatique peuvent être de taille importante. Dans ces cas-là, le serveur distant devrait n'envoyer que les données de spécification de l'espace chromatique qui sont recherchées par le client.

Si le champ de requête de préférences du client ne contient pas de préférences quant à la méthode de détermination de l'espace chromatique, alors les méthodes de détermination de l'espace chromatique prises en charge sont définies conformément aux informations contenues dans le champ de capacité et aucune préférence n'est définie.

Chaque préférence relative à la méthode de détermination de l'espace chromatique se compose de deux parties: la méthode particulière de détermination de l'espace chromatique et une limite facultative quant à cette préférence. Les valeurs légales de la méthode de détermination de l'espace chromatique sont spécifiées dans le Tableau C.9.

Tableau C.9 – Préférences du client concernant la méthode de détermination de l'espace chromatique

Méthode	Signification
"color-enum"	Le client préfère les spécifications d'espace chromatique qui utilisent la méthode des boîtes énumérées
"color-ricc"	Le client préfère les spécifications d'espace chromatique qui utilisent la méthode ICC limitée
"color-icc"	Le client préfère les spécifications d'espace chromatique qui utilisent une méthode ICC quelconque
"color-vend"	Le client préfère les spécifications d'espace chromatique qui utilisent la méthode du vendeur

La valeur facultative "meth-limit" spécifie une limite de la valeur APPROX pour cette méthode particulière de détermination de l'espace chromatique. Lorsqu'il utilise ces préférences afin de sélectionner une spécification de l'espace chromatique, le serveur distant doit considérer une spécification de méthode de détermination de l'espace chromatique avec une valeur APPROX égale à "meth-limit" ou moins, comme si la valeur APPROX réelle était (exactement) égale à 1. Cela permet aux clients de spécifier le point auquel la fidélité chromatique n'est plus importante dans une certaine méthode de détermination de l'espace chromatique, pour l'application en cours. Par exemple, une application de mise en page qui est seulement orientée vers l'alignement des données d'image avec d'autres éléments sur la page peut ne pas traiter du tout la fidélité chromatique et peut fixer la limite de méthode à 4, c'est-à-dire que la précision de la méthode de détermination de l'espace chromatique n'est pas importante. Une autre application qui affiche des images sur un écran de basse qualité peut fixer la limite de méthode à 3 afin d'indiquer que, tant que la précision chromatique est suffisante, elle sera affichée. Les caractères du champ doivent être interprétés comme un entier décimal non signé. Les valeurs légales sont définies en même temps que le champ APPROX dans le Tableau M.24 de la Rec. UIT-T T.801 | ISO/CEI 15444-2 et par des extensions et amendements à cette norme.

Lors de la sélection de la boîte de spécification d'espace chromatique à transmettre au client, le serveur distant doit utiliser l'algorithme suivant, comme représenté dans la Figure C.3.

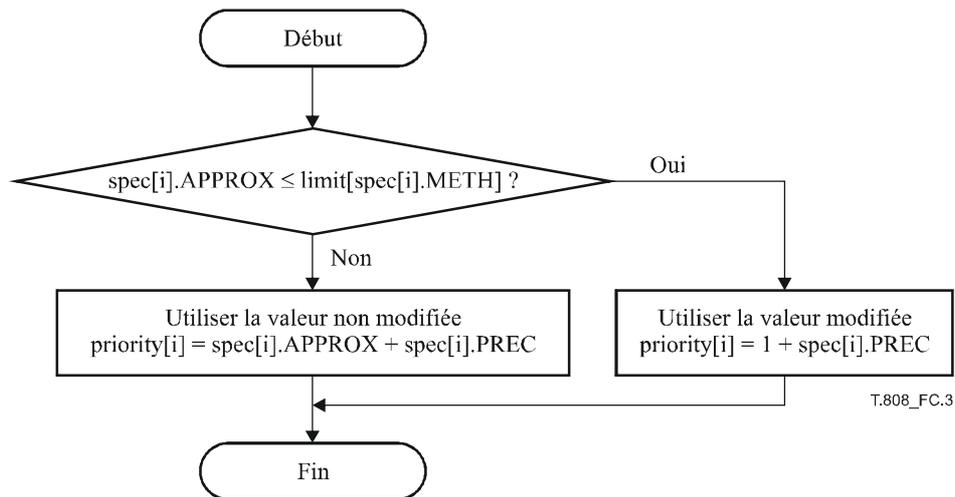


Figure C.3 – Spécification de l'espace chromatique: procédure de sélection de boîte

Pour chaque boîte de spécification de l'espace chromatique qui utilise une méthode prise en charge par le client:

- spec[] est une série tabulaire contenant toutes les boîtes de spécification d'espace chromatique à partir de la cible logique indiquée.
- spec[i].APPROX est la valeur du champ APPROX pour la *i*ème boîte de spécification de l'espace chromatique telle qu'elle apparaît dans la cible logique.
- spec[i].METH est la valeur du champ METH pour la *i*ème boîte de spécification de l'espace chromatique telle qu'elle apparaît dans la cible logique.
- spec[i].PREC est la valeur du champ PREC pour la *i*ème boîte de spécification de l'espace chromatique telle qu'elle apparaît dans la cible logique.
- limit[] est une série tabulaire contenant les valeurs "meth-limit" spécifiées dans le champ de requête, indexées par les valeurs légales du champ METH dans la boîte de spécification de l'espace chromatique.
- priority[] est une série tabulaire de valeurs de priorité calculées pour chaque boîte de spécification de l'espace chromatique dans la cible logique indiquée. priority[i] correspond à spec[i].

Si le serveur distant est informé du fait que le client ne prend pas en charge une certaine boîte de spécification de l'espace chromatique, alors le serveur distant doit ignorer cette boîte afin de sélectionner la boîte préférée de spécification de l'espace chromatique. Une fois que les valeurs priority[] ont été calculées pour chaque boîte de spécification de l'espace chromatique prise en charge, le serveur distant doit sélectionner la boîte ayant la plus basse valeur de priorité. Si de multiples boîtes ont une valeur de priorité égale à la valeur minimale pour cette cible logique, ce serveur distant doit sélectionner la méthode de détermination de l'espace chromatique dans l'ordre de préférence suivant:

- 1) Méthode des boîtes énumérées.
- 2) Méthode du vendeur.
- 3) Méthode ICC limitée.
- 4) Méthode ICC quelconque.

Quelles que soient les préférences du client en matière de boîte de spécification d'espace chromatique, le serveur distant peut renvoyer plus de boîtes de spécification d'espace chromatique que la seule boîte chromatique spécifiée par cet algorithme, selon la subdivision d'un fichier dans le segment de métadonnées.

C.10.2.4 Largeur de bande maximale

```
max-bandwidth = "mbw:" mbw
```

```
mbw = UINT ["K" / "M" / "G" / "T"]
```

Cette préférence signale la vitesse maximale à laquelle le client voudrait recevoir des données par cible logique. Si la valeur "mbw" se termine par "K", cette valeur est exprimée en kilobit/seconde, où 1 kilobit = 1 024 bits. Si la valeur "mbw" se termine par "M", cette valeur est exprimée en mégabit/seconde, où 1 mégabit = 1 024² bits. Si la valeur "mbw" se termine par "G", cette valeur est exprimée en gigabit/seconde, où 1 gigabit = 1 024³ bits. Si la valeur "mbw" se termine par "T", cette valeur est exprimée en téra-bit/seconde, où 1 téra-bit = 1 024⁴ bits. Sinon, la valeur est exprimée en

bit/seconde. Soit la capacité du serveur distant ou le réseau peut encore limiter la largeur de bande maximale disponible pour le service du protocole JPIP.

C.10.2.5 Tranche spectrale (bandwidth-slice)

```
bandwidth-slice = "slice:" slice
slice = NONZERO
```

Cette préférence peut servir à identifier la fraction de la largeur de bande disponible qui devrait être attribuée à ce canal. La valeur de tranche doit être strictement plus grande que 0. La fraction spectrale est obtenue par division de la valeur de tranche de chaque canal par la somme de toutes les valeurs de tranche de canal. Si elle n'est pas spécifiée, la valeur de tranche du canal revient par défaut à 1.

Par exemple, une faible valeur de tranche pourrait être utilisée afin de demander une fenêtre de visualisation "d'arrière-plan", bien qu'une tranche plus élevée puisse être utilisée pour une fenêtre de visualisation "d'avant-plan". Si la session contient des canaux qui sont associés aux différentes cibles logiques, leurs valeurs de tranche affectent la proportion de la largeur de bande disponible qui est assignée à ces différentes cibles (images).

C.10.2.6 Préférence en terme de générique

```
placeholder-pref = "meta:" placeholder-branch
placeholder-branch = "incr" / "equiv" / "orig"
```

Cette préférence peut servir à indiquer le traitement préféré des boîtes génériques. Lorsque des boîtes génériques apparaissent dans les métadonnées d'un flux JPP ou JPT, il peut y avoir jusqu'à trois représentations différentes du contenu d'une boîte: la boîte originale; une boîte à écoulement de flux équivalent; et un flux codé incrémentiel (signalé au moyen de l'indice). Ces possibilités sont expliquées en A.3.6 et A.4. Comme expliqué en A.4, l'hypothèse par défaut recommandée est que le client préférerait recevoir le flux codé incrémentiel, si disponible, à défaut de quoi il préférerait recevoir la boîte d'écoulement de flux équivalent, si disponible. Le client peut signaler une autre préférence en variante au moyen du mécanisme décrit ici. Les valeurs légales des préférences relatives aux génériques sont spécifiées dans le Tableau C.10.

Tableau C.10 – Préférences relatives aux éléments génériques

Méthode	Signification
"orig"	Le client préférerait recevoir la boîte originale, si disponible. A défaut, le client préférerait recevoir une boîte d'écoulement de flux équivalent, si disponible.
"equiv"	Le client préférerait recevoir une boîte d'écoulement de flux équivalent, si disponible. A défaut, le client préférerait recevoir la boîte originale, si disponible.
"incr"	Le client préférerait recevoir les segments de données de flux codé incrémentiel, si disponibles. A défaut, le client préférerait recevoir la boîte d'écoulement de flux équivalent, si disponible. Il s'agit de la politique recommandée par défaut.

Il n'est pas légal d'offrir plusieurs valeurs pour la préférence en terme de générique.

C.10.2.7 Séquencement des flux codés

```
codestream-seq-pref = "codeseq:" codestream-seq-option
codestream-seq-option = "sequential" / "reverse-sequential"
                        / "interleaved"
```

Cette préférence peut servir à indiquer comment le client souhaite que le serveur distant remette de multiples flux codés qui ont été demandés dans une requête isolée. Les valeurs légales des préférences relatives au séquencement des flux codés sont spécifiées dans le Tableau C.11.

Tableau C.11 – Préférences relatives au séquençement des flux codés

Méthode	Signification
"sequential"	Le client préférerait recevoir les multiples flux codés dans l'ordre séquentiel des trames (par exemple, desservir séquentiellement de multiples trames contenues dans un fichier MPEG 2000).
"reverse-sequential"	Le client préférerait recevoir les multiples flux codés (c'est-à-dire de multiples trames dans un fichier MPEG 2000) dans l'ordre inverse des trames.
"interleaved"	Le client préférerait recevoir les multiples flux codés de façon entrelacée (par exemple, de multiples couches de composition entrelacées dans le serveur distant à partir d'un fichier JPX).

Il n'est pas légal d'offrir plusieurs valeurs pour les préférences relatives au séquençement des flux codés.

C.10.3 Sensibilité différentielle (csf)

```

csf = "csf" "=" 1#csf-sample-line
csf-sample-line = csf-density [;" csf-angle] ";" 1$sensitivity
csf-density = "density" ":" UFLOAT
csf-angle = "angle" ":" UFLOAT
sensitivity = UFLOAT

```

Ce champ peut servir à fournir des informations concernant la sensibilité différentielle. Alors que ces informations peuvent représenter les effets de la sensibilité visuelle comme de la fonction de transfert de modulation d'un dispositif d'affichage, elles sont très facilement décrites en termes d'une hypothétique fonction de transfert de modulation. Quand elle est reproduite à la longueur de trame identifiée par le champ de requête de longueur de trame, l'imagerie est considérée comme étant transmise au moyen d'un dispositif dont la fonction de transfert de modulation (MTF) est $m(\omega_1, \omega_2)$, après quoi elle est visualisée par un sujet dont le système visuel humain a une fonction de sensibilité différentielle parfaitement uniforme. La fonction MTF $m(\omega_1, \omega_2)$ est décrite au moyen d'un ensemble d'échantillons. Ceux-ci sont logarithmiquement espacés dans le sens radial, le long d'un ou de plusieurs axes orientés. Le serveur distant peut interpoler ces échantillons au moyen de toute méthode qu'il juge appropriée, afin de récupérer la fonction MTF, qui à son tour peut servir à régler l'ordre dans lequel les séries d'octets des segments de données sont communiquées au client au moyen de messages à flux JPP ou JPT.

Chaque paramètre `csf-sample-line` représente les échantillons de fonction MTF $m(\omega_1, \omega_2)$ avec $\omega_1 = \pi d^n \cos \psi$, $\omega_2 = \pi d^n \sin \psi$, où n est l'indice d'échantillon, à partir de $n=0$ pour le premier échantillon `csf-density` contenu dans la ligne `csf-sample-line`, où ψ est l'orientation de la ligne d'échantillons de la fonction CSF, exprimée en degrés (revenant à 0 s'il n'y a aucune valeur `csf-angle`) et où d est la densité d'échantillonnage, qui ne doit pas être plus grande que 1,0. La valeur ω_1 décrit la fréquence horizontale en radians, où $\omega_1 = \pi$ est la fréquence de Nyquist horizontale. La valeur ω_2 décrit la fréquence verticale en radians, où $\omega_2 = \pi$ est la fréquence de Nyquist verticale.

Les valeurs des échantillons de la fonction MTF n'ont de sens qu'en relation les unes avec les autres; il n'y a aucune interprétation particulière pour leurs valeurs absolues.

Annexe D

Signalisation de la réponse du serveur distant

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

D.1 Syntaxe de réponse

D.1.1 Introduction

Cette annexe décrit tous les éléments possibles dans une réponse JPIP. Chaque paragraphe principal décrit le code de description d'état et sa phrase de cause associée, ses en-têtes de réponse et leurs valeurs possibles, ainsi que les données de réponse. En général, une réponse se composera de multiples en-têtes de réponse.

D.1.2 Structure de réponse

La réponse JPIP se compose des éléments suivants:

- code de description d'état ("status-code");
- phrase de cause ("reason-phrase");
- en-tête de réponse JPIP ("jpip-response-header");
- données de réponse.

Les éléments contenus dans la réponse devraient être conformes au protocole de transport choisi. Par exemple, en protocole HTTP, le code de description d'état et la phrase de cause apparaissent dans la ligne de description d'état, les en-têtes de réponse JPIP apparaissent dans les en-têtes de réponse HTTP et les données de réponse (si elles existent) apparaissent dans le corps de l'entité HTTP.

```
Status-Code = 3DIGIT
```

```
Reason-Phrase = *<TEXT, excluding CR and LF>
```

```
jpip-response-header =
    / JPIP-tid           ; D.2.2
    / JPIP-cnew         ; D.2.3
    / JPIP-qid          ; D.2.4
    / JPIP-fsiz         ; D.2.5
    / JPIP-rsiz         ; D.2.6
    / JPIP-roff         ; D.2.7
    / JPIP-comps        ; D.2.8
    / JPIP-stream       ; D.2.9
    / JPIP-context      ; D.2.10
    / JPIP-roi          ; D.2.11
    / JPIP-layers       ; D.2.12
    / JPIP-srate        ; D.2.13
    / JPIP-metareq      ; D.2.14
    / JPIP-len          ; D.2.15
    / JPIP-quality      ; D.2.16
    / JPIP-type         ; D.2.17
    / JPIP-mset         ; D.2.18
    / JPIP-cap          ; D.2.19
    / JPIP-pref         ; D.2.20
```

La chaîne "reason-phrase" devrait théoriquement comporter une explication textuelle du code de description d'état. Les codes de description d'état suivants peuvent être suffisants pour les applications du protocole JPIP.

D.1.3 Codes de description d'état et phrases de cause

D.1.3.1 Généralités

Le paramètre `Status-Code` est un entier de 3 chiffres représentant le code de résultat d'une tentative d'interprétation et de satisfaction de la requête. On utilise un sous-ensemble des codes de description d'état et des phrases de cause figurant dans le protocole HTTP/1.1. Les clients du protocole JPIP devraient s'attendre aux codes suivants. Les clients du protocole JPIP fonctionnant au-dessus du protocole HTTP peuvent voir également d'autres codes de description d'état.

D.1.3.2 Code 200 (OK)

Le serveur distant devrait utiliser ce code de description d'état s'il accepte la requête de fenêtre de visualisation pour traitement, éventuellement avec certaines modifications à la fenêtre de visualisation demandée, comme indiqué par les en-têtes additionnels inclus dans la réponse.

D.1.3.3 Code 202 (requête acceptée)

Les serveurs distants devraient émettre ce code de description d'état si la requête de fenêtre de visualisation était acceptable, mais au cas où une requête subséquente de fenêtre de visualisation aurait été découverte dans la file d'attente et aurait ensuite remplacé la première requête (cause: "wait=no"). Quand la première requête devient inapplicable avant que le serveur distant soit en mesure de traiter et de commencer la transmission d'une réponse, alors le code 202 de description d'état doit être utilisé. Il s'agit d'un cas courant en pratique, car un utilisateur interactif peut changer sa région recherchée plusieurs fois avant que le serveur distant finisse de répondre à une requête antérieure, ou avant que le serveur distant soit disposé à interrompre le traitement en cours.

D.1.3.4 Code 400 (requête erronée)

Les serveurs distants devraient émettre ce code de description d'état si la requête est incorrectement formatée, ou contient un champ non reconnu dans la chaîne d'interrogation.

D.1.3.5 Code 404 (échec d'identification)

Ce code de description d'état devrait être émis si le serveur distant ne peut pas rapprocher la ressource demandée d'un identificateur de cible émis. Cela peut être le résultat de tentatives d'accès non autorisées ou, plus probablement, de l'expiration d'une temporisation. Si le client manque cette fenêtre temporelle à cause d'une connexion de mauvaise qualité, il peut constater que l'identificateur de cible n'est plus actif.

D.1.3.6 Code 415 (type de média non pris en charge)

Ce code de description d'état peut être utilisé si le type particulier d'image spécifié dans le champ de requête de type de retour d'image ne peut pas être desservi.

D.1.3.7 Code 501 (ressource non implémentée)

Ce code de description d'état peut être utilisé si une portion de la présente Recommandation | Norme internationale, qui est demandée par la requête, ne peut pas être desservie.

D.1.3.8 Code 503 (service indisponible)

Ce code de description d'état devrait être utilisé si un identificateur de canal spécifié dans le champ de requête d'identificateur de canal est invalide.

D.2 En-têtes de réponse JPIP**D.2.1 Introduction aux en-têtes de réponse JPIP**

En répondant à une requête émise par un client, le serveur distant peut modifier certains aspects de la requête. Si le serveur distant modifie la requête, les paramètres modifiés doivent être identifiés au moyen d'en-têtes de réponse. Le nom de chaque en-tête de réponse est déduit du nom du champ de requête dont les paramètres doivent être modifiés, en préfixant le nom du champ de requête avec "JPIP-". Sauf spécification contraire, si les paramètres identifiés dans l'en-tête de réponse avaient été originellement spécifiés dans la requête du client, alors le serveur distant aurait répondu de la même façon, sauf que la réponse ne contiendrait plus ces en-têtes de réponse. En outre, des en-têtes de réponse JPIP peuvent être envoyés par le serveur distant afin d'informer le client des valeurs d'autres champs de requête non spécifiés, à utiliser dans de futures requêtes.

La réponse JPIP-*qid* est une exception en ce sens qu'elle doit être envoyée chaque fois que le client a inclus un identificateur de requête dans la requête: dans ce cas, la valeur de l'identificateur JPIP-*qid* doit toujours être la même que celle de l'identificateur *qid*.

Les paramètres relatifs à l'en-tête de réponse déduit, indiqués par le même élément du formalisme BNF que les paramètres contenus dans le champ initial de requête, ont la même signification et le même formatage que les paramètres relatifs au champ initial de requête.

Les seules exceptions à cette règle se rapportent aux en-têtes de réponse de nouveau canal et de qualité.

D.2.2 Identificateur de cible (JPIP-tid)

```
JPIP-tid = "JPIP-tid" ":" LWSP target-id
```

Le serveur distant doit envoyer cet en-tête de réponse si l'identificateur unique de cible du serveur distant diffère d'une façon ou d'une autre de l'identificateur fourni avec un champ de requête d'identificateur de cible, ou si le client n'a pas spécifié de champ de requête d'identificateur de cible. L'identificateur de cible est une chaîne arbitraire, attribuée par le serveur distant, dont la longueur ne dépasse pas 255 caractères. Si le champ de requête d'identificateur de cible spécifie une valeur de "0", le serveur distant est obligé d'inclure un en-tête de réponse d'identificateur de cible indiquant l'actuel identificateur de cible. Si le serveur distant est dans l'impossibilité d'attribuer des identificateurs uniques à la cible logique demandée et ne peut donc pas garantir son intégrité lors de multiples requêtes ou sessions, alors l'en-tête de réponse d'identificateur de cible doit spécifier une valeur de 0. Si le serveur distant fournit un identificateur de cible qui est différent de celui qui est spécifié dans la requête, il doit négliger tous les champs de requête "model", "tpmodel", "need" et "tpneed" lorsqu'il répond à cette requête.

D.2.3 Nouveau canal (JPIP-cnew)

```
JPIP-cnew = "JPIP-cnew" ":" LWSP "cid" "=" channel-id
            [ "," 1#(transport-param "=" TOKEN) ]

transport-param = TOKEN
```

Le serveur distant doit envoyer cet en-tête de réponse si et seulement s'il attribue un nouveau canal en réponse à un champ de requête de nouveau canal. La chaîne de valeur se compose d'une liste séparée par des virgules de paires nom=valeur, dont la première désigne le jeton identificateur du nouveau canal.

Les jetons suivants du paramètre de transport "transport-param" sont définis par la présente Recommandation | Norme internationale (voir le Tableau D.1).

Tableau D.1 – Valeurs légales du paramètre de transport

Valeur	Signification
"transport"	Ce paramètre doit être assigné à une des valeurs contenues dans la liste de noms de transport acceptables fournie dans le champ de requête de nouveau canal. Si de multiples noms de transport ont été fournis dans le champ de requête, l'en-tête de réponse doit identifier le transport réel qui sera utilisé avec le canal.
"host"	Ce paramètre désigne le nom ou l'adresse IP du serveur local pour le serveur distant JPIP qui doit gérer le nouveau canal. Le paramètre n'a pas besoin d'être renvoyé à moins que le serveur local ne diffère par rapport à celui auquel la requête a été réellement envoyée.
"path"	Ce paramètre désigne le composant de chemin de la localisation URL à utiliser lors de la construction de futures requêtes avec ce canal. Ce paramètre n'a pas besoin d'être renvoyé à moins que le nom du chemin ne diffère par rapport à celui qui a été utilisé dans la requête qui a été réellement envoyée.
"port"	Ce paramètre désigne le numéro du point d'accès (décimal) surveillé par le serveur distant JPIP qui doit gérer le nouveau canal afin de détecter des requêtes. Ce paramètre n'a pas besoin d'être renvoyé si le serveur local et le numéro du point d'accès sont identiques à ceux auxquels la requête originale a été envoyée. Le paramètre n'a pas non plus besoin d'être renvoyé si le serveur local diffère par rapport à celui auquel la requête a été envoyée et si la valeur par défaut du numéro de point d'accès associé au transport correspondant doit être utilisée.
"auxport"	Ce paramètre est utilisé avec des transports nécessitant un second canal physique. Si le protocole de transport "http-tcp" est utilisé, le point auxiliaire d'accès sert à connecter le canal TCP auxiliaire. Pour de plus amples détails, voir l'Annexe G. Ce paramètre n'a pas besoin d'être renvoyé si la requête originale impliquait un canal qui employait également un canal auxiliaire, possédant le même numéro auxiliaire de point d'accès. Sinon, le paramètre n'a besoin d'être renvoyé que si le numéro auxiliaire de point d'accès diffère de la valeur par défaut associée au transport choisi.

D.2.4 Identificateur de requête (JPIP-qid)

```
JPIP-qid = "JPIP-qid" ":" LWSP UINT
```

Le serveur distant doit envoyer cet en-tête de réponse si la requête du client inclut un identificateur de requête "qid". La valeur du paramètre "JPIP-qid" doit être identique à celle de l'identificateur "qid". Le serveur distant ne doit pas comprendre d'en-tête de réponse d'identificateur de requête si la requête du client correspondante ne comportait pas d'identificateur de requête.

NOTE – L'identificateur de requête du serveur distant, JPIP-qid, doit toujours être identique à l'identificateur de requête du client. L'identificateur de requête se distingue donc du fait que cet en-tête de réponse est envoyé quand le client a utilisé l'identificateur de requête, et non quand le serveur distant modifie la valeur.

D.2.5 Longueur de trame (JPIP-fsiz)

```
JPIP-fsiz = "JPIP-fsiz" ":" LWSP fx "," fy
```

Le serveur distant devrait envoyer cet en-tête de réponse si la longueur de trame pour laquelle des données de réponse seront desservies diffère de celle qui est demandée au moyen du champ de requête de longueur de trame.

D.2.6 Taille de région (JPIP-rsiz)

```
JPIP-rsiz = "JPIP-rsiz" ":" LWSP sx "," sy
```

Le serveur distant devrait envoyer cet en-tête de réponse si la taille de la région pour laquelle des données de réponse seront desservies diffère de celle qui est demandée.

D.2.7 Décalage (JPIP-roff)

```
JPIP-roff = "JPIP-roff" ":" LWSP ox "," oy
```

Le serveur distant devrait envoyer cet en-tête de réponse si le décalage de la région pour laquelle des données de réponse seront desservies diffère de celui qui est demandé.

D.2.8 Composants (JPIP-comps)

```
JPIP-comps = "JPIP-comps" ":" LWSP 1#UINT-RANGE
```

Le serveur distant devrait envoyer cet en-tête de réponse si les composants pour lesquels il va desservir des données diffèrent de ceux qui ont été demandés au moyen du champ de requête de composants. Il n'est pas obligé d'envoyer cet en-tête de réponse si les composants d'image demandés n'existent pas dans l'un quelconque des flux codés demandés.

D.2.9 Flux codé (JPIP-stream)

```
JPIP-stream = "JPIP-stream" ":" LWSP 1#(prefixed-range / sampled-range)
prefixed-range = "<" ctxt-id ":" ctxt-elt ">" sampled-range
ctxt-id = UINT
ctxt-elt = UINT
```

Le serveur distant devrait envoyer cet en-tête de réponse afin d'informer le client du (des) flux codé(s) pour lequel (lesquels) il va desservir des données, à moins qu'il ne soit en train de desservir des données, en réponse à tous les flux codés demandés directement au moyen de tout champ de requête de flux codé et en réponse à tous les flux codés demandés indirectement au moyen de tout champ de requête de contexte de flux codé. Le serveur distant devrait utiliser la syntaxe du paramètre "prefixed-range" afin d'identifier les flux codés pour lesquels des données doivent être desservies en réponse à un champ converti de requête de contexte de flux codé. Dans ce cas, la valeur "ctxt-id" doit identifier l'étendue spécifique "context-range" à partir du champ de requête de contexte de flux codé dont la conversion va produire les flux codés applicables. En outre, la valeur "ctxt-elt" doit identifier l'élément particulier dans l'étendue de contexte identifiée par "ctxt-id", dont la conversion va produire les flux codés applicables.

Une valeur de 0 pour l'identificateur "ctxt-id" signifie que la première étendue de contexte contenue dans le champ de requête de contexte de flux codé est celle qui a produit l'étendue de flux codés qui suit le préfixe. De même, une valeur de 1 pour l'identificateur "ctxt-id" signifie que la deuxième étendue de contexte contenue dans le champ de requête de contexte de flux codé est celle qui a produit l'étendue suivante de flux codés, et ainsi de suite.

Une valeur de 0 pour le paramètre "ctxt-elt" signifie que le premier contexte contenu dans l'étendue de contexte applicable est celui qui a produit l'étendue de flux codés qui suit le préfixe.

Exemple:

Requête émise par un client:

```
stream=0&context=jpxl<2-7:2>[s0i0],jpxl<3-5>[s1i3]
```

Réponse du serveur distant:

```
JPIP-context: jpxl<2-7:2>[s0i0]=0,1;jpxl<9-10>[s1i3]=0
```

```
JPIP-stream: 0,<0:1>1,<1:0>0,<1:1>0
```

C'est-à-dire que le serveur distant va répondre avec des données issues de:

- 1) l'application directe de la fenêtre de visualisation au flux codé 0 (comme demandé au moyen de "stream=0");
- 2) la conversion de la fenêtre de visualisation en couche de composition JPX 4, conformément à l'instruction de composition 0 contenue dans l'ensemble d'instructions de composition 0, dans la mesure où il s'applique au flux codé 1;

- 3) la conversion de la fenêtre de visualisation en couche de composition JPX 9, conformément à l'instruction de composition 3 contenue dans l'ensemble d'instructions de composition 1, dans la mesure où il s'applique au flux codé 0; et
- 4) la conversion de la fenêtre de visualisation en couche de composition JPX 10, conformément à l'instruction de composition 3 contenue dans l'ensemble d'instructions de composition 1, dans la mesure où il s'applique au flux codé 0.

D.2.10 Contexte du flux codé (JPIP-context)

```
JPIP-context = "JPIP-context" ":" LWSP 1$(context-range "=" 1#sampled-range)
```

Le serveur distant devrait envoyer cet en-tête de réponse s'il est en mesure de traiter l'une quelconque des valeurs du paramètre "context-range" fournies au moyen d'un champ de requête de contexte de flux codé. L'en-tête décrit chaque étendue context-range qui doit être traitée, de même que les indices de tous les flux codés qui sont associés à cette étendue de contexte. Le serveur distant peut omettre certaines valeurs du paramètre "context-range" qui ont été originellement fournies dans le champ de requête de contexte de flux codé, si elles ne sont pas en cours de traitement. Le serveur distant peut également modifier les valeurs du paramètre "context-range" originellement fournies dans le champ de requête de contexte de flux codé. Deux types de modification sont autorisés:

- a) le serveur distant peut limiter l'ensemble des éléments d'image (par exemple les couches de composition) qui ont été originellement demandés;
- b) le serveur distant peut abandonner des modificateurs de transformation géométrique qu'il n'est pas en mesure de prendre en charge (par exemple un modificateur "track" ou "movie" contenu dans une chaîne mj2t-context).

D.2.11 ROI (JPIP-roi)

```
JPIP-roi = "JPIP-roi" ":" LWSP
"roi" "=" region-name ";"
"fsiz" "=" UINT "," UINT ";"
"rsiz" "=" UINT "," UINT ";"
"roff" "=" UINT "," UINT ";"
```

```
region-name = 1*(DIGIT / ALPHA / "_")
```

En réponse à une requête du client pour une région ROI, un serveur distant doit spécifier, au moyen de l'en-tête de réponse de région ROI, l'étendue de la région ROI réellement desservie. Si le serveur distant est dans l'impossibilité de satisfaire la requête de région ROI, il doit répondre avec l'en-tête de réponse de région ROI simplement mis à: "JPIP-roi: roi=no-roi". En plus de la région ROI, le serveur distant spécifie également, au moyen des en-têtes de réponse concernant la longueur de trame, la trame de la région et le décalage, la région de l'image qu'il est en train de desservir au titre d'un repli.

Si le serveur distant est en mesure de desservir la région ROI, mais que, pour certaines causes, il ait besoin de redimensionner la portion de l'image renvoyée, il doit envoyer l'en-tête de réponse de région ROI décrivant cette région ROI, ainsi que les en-têtes de réponse relatifs à la longueur de trame, à la taille de la région et au décalage, qui décrivent la partie de la région ROI qui est renvoyée.

D.2.12 Couches (JPIP-layers)

```
JPIP-layers = "JPIP-layers" ":" LWSP UINT
```

Le serveur distant devrait envoyer cet en-tête de réponse si le nombre de couches qu'il va desservir est plus petit que la valeur spécifiée par le champ de requête de couches. Comme la fenêtre de visualisation est typiquement desservie en mode de progression qualitative, le serveur distant n'est pas obligé (et même peut ne pas être en mesure) de déterminer le nombre de couches qui sont couvertes par les données de réponse qu'il remet. Cependant, si le nombre demandé de couches dépasse le nombre de couches disponibles à partir de tous les flux codés dans la fenêtre de visualisation, le serveur distant devrait au moins identifier le nombre maximal de couches disponibles. Tout serveur distant qui accepte un champ de requête d'alignement (voir C.7.1) doit offrir une réponse relative aux couches JPIP si le nombre de couches qu'il va desservir est plus petit que la valeur spécifiée par le champ de requête de couches.

D.2.13 Fréquence d'échantillonnage (JPIP-srate)

```
JPIP-srate = "JPIP-srate" ":" LWSP UFLOAT
```

Le serveur distant devrait envoyer cet en-tête de réponse si la fréquence moyenne d'échantillonnage des flux codés qu'il va envoyer au client est censée différer de celle qui est demandée au moyen d'un champ de requête de fréquence

d'échantillonnage et si la fréquence d'échantillonnage est connue. Si les flux codés de source ne contiennent pas d'informations temporelles, cet en-tête de réponse ne devrait pas être envoyé.

D.2.14 Requête de métadonnées (JPIP-metareq)

```
JPIP-metareq = "JPIP-metareq" ":" LWSP
               1#( "[" 1$(req-box-prop) "]" [root-bin] [max-depth] )
               [metadata-only]
```

```
req-box-prop = box-type [limit] [metareq-qualifier] [priority]
```

Le serveur distant devrait envoyer cet en-tête de réponse s'il doit modifier la valeur du paramètre "max-depth", "limit", "metareq-qualifier" ou "priority" fournie dans un champ de requête de métadonnées.

D.2.15 Longueur maximale de réponse (JPIP-len)

```
JPIP-len = "JPIP-len" ":" LWSP UINT
```

Le serveur distant devrait envoyer cet en-tête de réponse si la limite d'octets spécifiée dans un champ de requête de longueur maximale de réponse a été trop petite pour permettre une réponse non vide à moins que la limite d'octets n'ait été égale à zéro. S'il est renvoyé, l'en-tête "JPIP-len" doit être une valeur qui informe le client au sujet d'une longueur maximale de réponse appropriée, "len", pour des requêtes ultérieures. Si "len=0", le serveur distant devrait satisfaire cette requête avec des en-têtes de réponse sans données de réponse.

D.2.16 Qualité (JPIP-quality)

```
JPIP-quality = "JPIP-quality" ":" LWSP (1*2DIGIT / "100" / "-1")
```

Le serveur distant peut envoyer cet en-tête de réponse afin d'informer le client de la valeur de qualité qui sera associée aux données d'image renvoyées une fois que cette requête aura été complétée. Si la requête est interrompue par une autre requête (n'ayant pas la valeur "wait=yes"), cette valeur de qualité ne peut pas être précise. La valeur de qualité se rapporte seulement à la fenêtre de visualisation demandée et a la même interprétation que le champ de requête de qualité. Si le serveur distant a ignoré la requête du client, une valeur "-1" doit être renvoyée.

D.2.17 Type de retour d'image (JPIP-type)

```
JPIP-type = "JPIP-type" ":" LWSP image-return-type
```

Le serveur distant devrait inclure cet en-tête de réponse à moins qu'un autre mécanisme ne désigne le sous-type MIME des données d'image renvoyées. Exemples d'autres mécanismes:

- un en-tête HTTP "Content-Type:";
- des réponses à des requêtes qui sont associées à une session dont le type d'image en retour a déjà été signalé.

D.2.18 Ensemble de modèles (JPIP-mset)

```
JPIP-mset = "JPIP-mset" ":" LWSP 1#sampled-range
```

Le serveur distant devrait inclure cet en-tête de réponse si la requête du client contient un champ de requête d'ensemble de modèles et si l'ensemble des flux codés identifiés par ce champ de requête d'ensemble de modèles du client diffère d'une façon ou d'une autre de l'ensemble des flux codés pour lesquels le serveur distant est actuellement disposé à conserver des informations de modèle de cache. L'ensemble des flux codés pour lesquels le serveur distant tient à jour des informations de modèle de cache devrait inclure tous les flux codés qui sont associés aux données de réponse du serveur distant (soit ceux qui sont identifiés dans la requête du client, ou ceux qui sont identifiés par l'en-tête de réponse de flux codé du serveur, le cas échéant). En dehors de ces flux codés, l'ensemble "mset" du serveur distant peut ne pas être plus grand que celui qui a été identifié par le champ de requête d'ensemble de modèles du client.

D.2.19 Capacité recherchée (JPIP-cap)

```
JPIP-cap = "JPIP-cap" ":" LWSP 1#capability-code
```

Cet en-tête de réponse spécifie que le client doit prendre en charge un certain élément de service afin d'interpréter la cible logique de façon conforme. Les capacités légales sont celles qui ont été définies pour le champ de requête de capacité dans le Tableau C.6.

D.2.20 Préférence indisponible (JPIP-pref)

```
JPIP-pref = "JPIP-pref" ":" LWSP 1#related-pref-set
```

Cet en-tête de réponse devrait être fourni si et seulement si un champ de requête de préférences du client contenait un paramètre "related-pref-set" avec le modificateur "/r" (requisites) que le serveur distant n'était pas disposé à prendre en charge. Dans ce cas, une valeur d'erreur devrait également être renvoyée pour le code de description d'état de la réponse. La chaîne de valeur se compose d'un ou de plusieurs des ensembles "related-pref-set" qui n'ont pas pu être pris en charge, répétés exactement comme ils apparaissaient dans la requête de préférences du client.

Bien que cela soit souhaitable, il n'est pas nécessaire que cet en-tête de réponse énumère tous les paramètres "related-pref-set" requis qui ne peuvent pas être pris en charge. Il est donc admissible qu'un serveur distant ne pénètre dans le champ de requête de préférences du client que lorsqu'il rencontre un paramètre "related-pref-set" qui spécifie "/r" et ne peut pas être pris en charge. Voir en C.10.2.1 de plus amples informations sur le moment où cet en-tête de réponse doit être utilisé.

Tableau D.2 – Codes de cause définis

Code de cause	Cause	Explication
1	Image terminée	Le serveur distant a transféré vers le client toutes les informations d'image disponibles (et non pas seulement les informations appartenant à la fenêtre de visualisation demandée). Ce code de cause a une certaine signification pour les requêtes en mode session. Pour une requête en mode session, ce code de cause implique que le client a reçu toutes les données qui pourraient avoir été envoyées en réponse à toute requête en mode session associée à cette cible logique. A l'exception possible des requêtes qui comprennent des champs de requête de gestion de cache, toute requête subséquente en mode session sera satisfaite sans données de réponse et avec R=1 EOR.
2	Fenêtre terminée	Le serveur distant a transféré toutes les informations disponibles qui appartenaient à la fenêtre de visualisation demandée. Ce code de cause a une certaine signification pour les requêtes en mode session. Pour une requête en mode session, ce code de cause implique que le client a reçu toutes les données qui pourraient avoir été envoyées en réponse à cette requête et que les données de réponse n'ont pas été limitées par un champ de limite de données (en longueur ou en qualité) dans la requête, ni par le traitement d'une requête subséquente. A l'exception possible des requêtes qui comprennent des champs de requête de gestion de cache, toute répétition subséquente de la requête sera satisfaite sans données de réponse et avec R=2 EOR.
3	Changement de fenêtre	Le serveur distant va terminer sa réponse afin de desservir une nouvelle requête qui ne spécifie pas "wait=yes".
4	Limite d'octets atteinte	Le serveur distant va terminer sa réponse parce que la limite d'octets spécifiée dans un champ de requête de longueur maximale de réponse a été atteinte.
5	Limite de qualité atteinte	Le serveur distant va terminer sa réponse parce que la limite de qualité spécifiée dans un champ de requête de qualité a été atteinte.
6	Limite de session atteinte	Le serveur distant va terminer sa réponse parce qu'une certaine limite – comme une temporisation – a été atteinte concernant les ressources de la session. Aucune nouvelle requête ne devrait être émise au moyen d'un identificateur de canal associé à cette session.
7	Limite de réponse atteinte	Le serveur distant va terminer sa réponse parce qu'une certaine limite, par exemple une temporisation, a été atteinte. Si la requête est émise dans une session, de nouvelles requêtes pourront encore être émises au moyen d'un identificateur de canal associé à cette session.
0xFF	Cause non spécifiée	Le serveur distant va terminer sa réponse pour une raison qui n'est pas spécifiée.
Autres valeurs		Champ réservé pour utilisation par l'ISO.

D.3 Données de réponse

Pour tout type de retour d'image autre qu'en flux JPP ou JPT, y compris le flux codé non traité, les données de réponse devraient se composer de l'entité demandée en totalité. Pour les types de retour d'image en flux JPP ou JPT, les données de réponse se composent d'une séquence de messages tels que définis dans l'Annexe A, terminés par un même message EOR (fin de réponse), lequel n'est pas défini dans l'Annexe A et ne fait pas formellement partie des types de média à flux JPP ou JPT.

Un message EOR se compose d'un en-tête et d'un corps. L'en-tête de message EOR se compose de l'identificateur sur un seul octet, 0x00, suivi par un unique octet de code de cause, R, puis d'un unique décompte d'octets de segment VBAS, indiquant le nombre d'octets contenus dans le corps du message EOR. La présente Recommandation | Norme internationale n'offre aucune interprétation normative concernant le contenu du corps du message EOR.

Noter que le corps du message EOR ne contribue pas à la restriction relative au décompte d'octets qui est associée au champ de requête de longueur maximale de réponse comme défini dans l'Annexe C.

Noter que le message EOR signifie que le serveur distant a acheminé tout le contenu pertinent des segments correspondants de données dans le cas d'une requête émise par un client. Il ne s'agit pas nécessairement du contenu entier de ces segments de données. La réponse est terminée quand une limite spécifiée par le client a été atteinte. Si aucune limite n'a été spécifiée, alors le message EOR signifiera que tout le contenu des segments correspondants de données a été desservi.

Les codes de cause sont actuellement définis (voir le Tableau D.2).

Annexe E

Téléchargement amont d'images vers le serveur distant

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

E.1 Introduction

Il est prévu que les images soient placées dans un serveur distant de diverses façons qui sont en dehors du domaine d'application de la présente Recommandation | Norme internationale. L'objet de cette annexe consiste à décrire un mécanisme permettant de télécharger des portions d'une image en amont vers un serveur distant.

E.2 Requête de téléchargement amont

E.2.1 Structure des requêtes

Une requête de téléchargement amont se compose d'un ou de plusieurs champs de requête définis dans l'Annexe C et d'un corps de requête.

E.2.2 Champs de requête de téléchargement amont

Les champs de requête visant un téléchargement vers l'amont doivent contenir un champ de requête de téléchargement amont. Les champs de cible, de sous-cible et d'identificateur de cible (voir C.2.2, C.2.3 et C.2.4) peuvent également être utilisés. Pour le téléchargement vers l'amont d'un type de média à image complète, les champs de requête de longueur de trame, de décalage et de taille de région (voir C.4.2, C.4.3 et C.4.4) servent à indiquer la position de la portion téléchargée vers l'amont dans l'image entière. Pour les téléchargements amont de flux JPT et de flux JPP, le numéro du segment de données (et donc le numéro du pavé ou du district) de même que l'en-tête principal, indiquent l'emplacement des données codées de sorte que les champs de requête de fenêtre de visualisation ne sont pas nécessaires.

E.2.3 Corps d'une requête de téléchargement amont

E.2.3.1 Généralités

Le corps d'une requête de téléchargement amont se compose d'un des types d'image pris en charge: flux JPP, flux JPT, ou type de média à image complète. Le corps contient les données dont le client va demander le traitement par le serveur distant. La présente Recommandation | Norme internationale ne prend pas en charge le téléchargement amont des données d'image non traitées.

E.2.3.2 Flux JPT

Le corps de la requête contient tous les segments de données dont le client souhaite le remplacement par le serveur distant (segments de données d'en-tête, segment de métadonnées et segments de données de pavé). Si le client ne télécharge pas vers l'amont un segment de données d'en-tête principal, les segments de données de pavé doivent être codés de façon compatible avec l'en-tête principal actuel.

E.2.3.3 Flux JPP

Le corps de la requête contient tous les segments de données dont le client souhaite le remplacement par le serveur distant (segments de données d'en-tête, segments de données d'en-tête de pavé, segment de métadonnées et segments de données de district). Si le client ne télécharge pas vers l'amont un segment de données d'en-tête principal ou un segment de données d'en-tête de pavé, les districts doivent être codés de façon compatible avec l'en-tête principal et l'en-tête de pavé actuels.

E.2.3.4 Téléchargement amont d'image complète

Le corps de la requête contient un type de média à image complète représentant les échantillons que le client souhaite modifier.

Dans le cas d'un téléchargement amont d'image complète, la requête peut comporter des champs de requête de longueur de trame, de taille de région et de décalage. Le champ de requête de longueur de trame doit indiquer la taille de la grille de référence de l'image. Dans le cas d'un téléchargement amont d'image complète, la compression n'a pas besoin d'être terminée de façon compatible avec la cible logique dans le serveur distant. Si la taille de l'image téléchargée en amont dépasse l'étendue indiquée dans le champ de requête de taille de région, le serveur distant devrait limiter les modifications dans la mesure spécifiée par le champ de requête de taille de région.

E.3 Réponse du serveur distant

E.3.1 Généralités

Le serveur distant doit satisfaire à une requête de téléchargement amont avec un code de description d'état et une phrase de cause conformément à l'Annexe D. Les codes de retour et les phrases de cause utiles pour le téléchargement d'image vers l'amont sont présentés dans les paragraphes suivants.

E.3.2 Code 201 (créé)

Le serveur distant devrait utiliser ce code de description d'état si, dès réception d'une requête de téléchargement amont, une nouvelle ressource a été définie dans le serveur distant. Celui-ci doit avoir complété la création avant de renvoyer cette requête. S'il doit y avoir un retard, le serveur distant devrait renvoyer le code 202 (accepté) au lieu de 201 (créé).

Le serveur distant devrait inclure un en-tête dans la réponse avec un nouveau champ d'identificateur de cible pour la ressource mise à jour.

Aucun corps n'a besoin d'être renvoyé.

E.3.3 Code 202 (accepté)

Le serveur distant devrait utiliser ce code de description d'état si un téléchargement vers l'amont crée une nouvelle ressource mais que le serveur distant ne soit pas encore disposé à la desservir. Le serveur distant peut également utiliser ce code de description d'état pour la mise à jour d'une ressource actuelle.

E.3.4 Code 400 (requête erronée)

Les serveurs distants devraient émettre ce code de description d'état si la requête est incorrectement formatée, ou si l'interrogation contient des champs de requête qui sont incompatibles avec le téléchargement amont ou contient un champ non reconnu dans la chaîne d'interrogation.

E.3.5 Code 404 (échec d'identification)

Ce code de description d'état devrait être émis si le serveur distant ne peut pas rapprocher la ressource demandée d'un identificateur de cible émis.

E.3.6 Code 415 (type de média non pris en charge)

Ce code de description d'état peut être émis afin d'indiquer que, bien que les téléchargements amont soient pris en charge, les téléchargements amont d'un type particulier (par exemple image complète, flux JPT, ou flux JPP), inclus dans la requête, ne sont pas pris en charge.

E.3.7 Code 501 (ressource non implémentée)

Cette description d'état pourra être utilisée si le serveur distant ne prend pas en charge le téléchargement vers l'amont ou ne prend pas en charge une certaine option avec téléchargement amont.

E.4 Fusionnement de données dans le serveur distant

E.4.1 Mise à jour de l'image

Après réception des données téléchargées vers l'amont, le serveur distant peut créer une nouvelle version de la cible logique et l'offrir aux clients accédant à une localisation URL ancienne ou nouvelle. Cependant, le serveur distant ne doit pas utiliser l'ancien champ de requête d'identificateur de cible afin de donner accès à d'éventuelles données fusionnées ou mises à jour.

Si le client inclut un champ de requête d'identificateur de cible dans la requête de téléchargement amont et que l'identificateur de cible ne corresponde pas à l'identificateur de cible actuel du serveur distant pour la ressource, ce serveur distant ne devrait pas mettre à jour l'image. Cette discordance peut indiquer que le client a édité une version précédente de l'image qui a déjà été modifiée. Des serveurs distants peuvent refuser d'accepter les téléchargements amont qui ne contiennent pas de champ de requête d'identificateur de cible. C'est une façon d'empêcher de multiples éditions simultanées d'une cible par clients différents. Les serveurs distants offrant des capacités d'édition peuvent s'occuper de tels problèmes par un moyen ou un autre, comme le verrouillage de cible.

Un client du protocole JPIP peut télécharger vers l'amont une partie d'une nouvelle image en spécifiant un identificateur de cible de 0, ou au moyen d'une nouvelle localisation URL, ou d'une cible que le serveur distant ne possède pas. Le serveur distant devrait émettre un identificateur de cible pour le téléchargement amont. Un client peut continuer à télécharger vers l'amont des portions additionnelles de la nouvelle image en utilisant l'identificateur de cible renvoyé par le serveur distant avec le téléchargement amont précédent.

E.4.2 Flux JPT

Un serveur distant acceptant des données de segment de pavé doit d'abord supprimer toutes les anciennes données de segment de pavé pour les pavés en cours de téléchargement amont puis doit inclure les nouvelles données de segment de pavé dans le flux codé. Une mise à jour ne peut pas être effectuée s'il en résulte une modification du nombre, de la dimension ou de l'emplacement de pavés – car la structure de l'image ne peut pas être modifiée par un téléchargement amont. En particulier, un serveur distant ne devrait pas accepter de téléchargement amont de segments de données de pavé pour un flux codé contenant un segment de marqueur PPM dans l'en-tête principal, à moins que le client n'offre un nouvel en-tête principal dans ce téléchargement amont. Tous les segments marqueurs PLM ou TLM doivent être supprimés ou mis à jour. Un segment de données d'en-tête principal en flux JPT doit être téléchargé en amont pour de nouvelles images.

La façon dont les éléments de pavé d'un flux codé sont formés à partir d'un segment de données de pavé n'est pas spécifiée. Le client n'a pas nécessairement besoin d'offrir tous les éléments d'un pavé, ni de voir le dernier élément de pavé complété. Le serveur distant doit mettre à jour l'en-tête principal et toutes portions du format de fichier affectées (par exemple la longueur de la boîte de flux codé).

Lors du fusionnement de données, le nombre ou la taille des pavés ne doit pas être modifié et les données qui ne sont pas remplacées par le processus de téléchargement vers l'amont doivent avoir la signification qu'elles avaient initialement, avant le téléchargement amont.

E.4.3 Flux JPP

Un serveur distant acceptant des messages à segments de données de district doit d'abord supprimer les anciens segments de données de district correspondant aux districts en cours de téléchargement amont, puis inclure les nouvelles données de segment de données de district. Une modification ne peut pas être apportée à un en-tête qui se traduit par une modification du nombre de districts, ou de la signification de l'identificateur de district, ou de l'emplacement ou de la taille de chaque district dans sa résolution d'élément de pavé. Les segments de données d'en-tête de pavé et les segments de données d'en-tête principal d'un flux JPP doivent être téléchargés en amont afin d'obtenir de nouvelles images.

La façon dont les paquets de district sont formés à partir d'un segment de données de district n'est pas spécifiée. Le client n'a pas nécessairement besoin d'offrir tous les paquets d'un district, ni même de compléter le dernier paquet fourni.

Lors du fusionnement de données, le nombre ou la taille des districts ne doit pas être modifié et les données qui ne sont pas remplacées par le processus de téléchargement vers l'amont doivent avoir la signification qu'elles avaient initialement, avant le téléchargement amont.

E.4.4 Segments de métadonnées à flux JPP ou JPT

Un segment de métadonnées peut être téléchargé vers l'amont, en remplaçant le contenu d'un segment de métadonnées existant. Comme le serveur distant contrôle la subdivision attribuant des métadonnées aux segments de métadonnées, le client doit suivre la structure du segment de métadonnées de ce serveur distant. Le client ne doit pas changer les génériques contenus dans un segment de métadonnées, sauf afin d'en supprimer un complètement. Lors du téléchargement amont d'un segment entier de métadonnées, les clients peuvent ajouter de nouvelles métadonnées en les ajoutant à la fin de l'ancien segment de métadonnées, ou en insérant de nouvelles métadonnées entre les boîtes de l'ancien segment de métadonnées. Le serveur distant doit gérer les génériques et la structure du segment de métadonnées. Cela implique la mise à jour de tous les génériques pointant sur toutes les boîtes de métadonnées précédentes qui ont été modifiées ou affectées par la modification. Le serveur distant doit supprimer toutes les boîtes de métadonnées qui ont été pointées par un générique que le client a supprimé. Le serveur distant peut restructurer les métadonnées après qu'un téléchargement vers l'amont a été accepté, mais avant que la nouvelle ressource ait été créée. Si des sections inutilisées sont laissées dans le fichier après téléchargement amont, des boîtes libres doivent servir à remplir ces sections.

E.4.5 Téléchargement amont d'image complète

Dans le cas d'un téléchargement acceptable d'une image complète vers l'amont, le serveur distant devrait décompresser (au besoin) la sous-image téléchargée en amont, décompresser certaines portions de l'image complète dans le serveur distant, remplacer ces pixels dans le domaine spatial (décompressé) et recompresser tous les pavés ou districts affectés par l'opération de mise à jour.

NOTE – Cette technique nécessite plus de calculs dans le serveur distant; cependant, elle supprime la possibilité que le client utilise des données d'image comprimée de façon incompatible (par exemple le nombre erroné de niveaux de transformation en ondelette).

Annexe F

Utilisation du protocole JPIP au-dessus du protocole HTTP

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

F.1 Introduction

Cette annexe définit la méthode permettant d'utiliser le protocole JPIP avec le protocole HTTP aussi bien pour les requêtes que pour les réponses. Les paramètres de requête JPIP conformes à l'Annexe C sont encapsulés dans des structures de requête http légalles. Les réponses du serveur distant (y compris les codes de description d'état, les en-têtes, les messages et les codes de réponse), conformes à l'Annexe D, sont encapsulées dans des réponses HTTP légalles. Toutes les requêtes et réponses doivent être codées comme spécifié par la norme HTTP.

Noter que le texte et les exemples contenus dans cette annexe décrivent l'utilisation du protocole JPIP au-dessus du protocole HTTP. Il est attendu que la même association puisse être utilisée pour le protocole HTTP.

F.2 Requêtes

F.2.1 Introduction relative aux requêtes

L'Annexe C définit les champs de requête. Quand elle est transportée au moyen du protocole HTTP, la requête JPIP peut apparaître comme une chaîne d'interrogation pour une requête HTTP "GET" ou comme le corps d'une requête HTTP "POST". Etant donné que certains systèmes HTTP limitent la longueur de la chaîne d'interrogation fournie dans une requête "GET", la requête "POST" est préférée pour les longues requêtes JPIP.

NOTE 1 – La requête HTTP est définie dans le § 5 du document RFC 2616 comme suit:

```
Request = Request-Line           ; HTTP Section 5.1
        0*(( general-header      ; HTTP Section 4.5
           / request-header      ; HTTP Section 5.3
           / entity-header ) CRLF) ; HTTP Section 7.1
        CRLF
        [ message-body ]        ; HTTP Section 4.3
```

NOTE 2 – Les paramètres HTTP Request-Line et Request-URI sont définis comme suit:

```
Request-Line = Method SP Request-URI SP HTTP-Version CRLF
Request-URI  = "*" / absoluteURI / abs_path / authority
```

NOTE 3 – Le document RFC 2396 définit ceci:

```
absoluteURI  = scheme ":" ( hier_part / opaque_part )
hier_part    = ( net_path / abs_path ) [ "?" query ]
abs_path     = "/" path_segments
```

F.2.2 Requêtes GET

Une requête JPIP peut être fournie à un serveur distant comme requête HTTP. Pour une requête "GET", la requête HTTP est limitée comme suit:

- l'élément "Method" doit être "GET";
- l'élément "query" doit être zéro ou plusieurs champs "jpip-request-field" séparés par '&'.

Exemple de requête JPIP encapsulée dans une requête HTTP "GET":

```
GET /images/kids.jpg?rsiz=640,480&roff=320,240&fsiz=1280,1024 HTTP/1.1
Host: get.jpeg.org
CRLF
```

ISO/CEI 15444-9:2005 (F)

Un exemple équivalent utilisant un identificateur "absoluteURI" au lieu d'un chemin "abs_path" est le suivant:

```
GET http://get.jpeg.org/images/kids.jp2?rsiz=640,480&roff=320,240
&fsiz=1280,1024 HTTP/1.1
```

CRLF

NOTE – La présente Recommandation | Norme internationale n'impose aucune restriction concernant le composant de système de l'identificateur "absoluteURI".

F.2.3 Requêtes POST

Une requête JPIP peut être fournie à un serveur distant après avoir été encapsulée dans une requête HTTP "POST", dans laquelle la requête HTTP est limitée comme suit:

- L'élément "Method" doit être "POST".
- L'élément "entity-body" doit être zéro ou plusieurs champs "jpip-request-field" séparés par '&'.
- La ligne d'en-tête "Content-type:" devrait être incluse sous forme d'en-tête "entity-header" et devrait contenir la valeur "application/x-www-form-urlencoded".

Exemple de requête JPIP encapsulée dans une requête HTTP "POST":

```
POST /cgi-bin/j2k_server.cgi HTTP/1.1
Host: post.jpeg.org
Content-type: application/x-www-form-urlencoded
Content-length: 62
CRLF
target=/images/kids.jp2&rsiz=640,480&roff=320,240&fsiz=1280,1024
```

F.2.4 Requêtes de téléchargement amont

Une requête de téléchargement amont est une requête HTTP légale qui est limitée comme suit:

- L'élément "Method" doit être "POST".
- La localisation URL doit contenir le champ d'interrogation relatif au téléchargement amont.
- Le type de contenu doit être le type d'image du corps: image/jpt-stream, image/jpp-stream, ou un type de média à image complète.

Exemple de requête JPIP de téléchargement amont:

```
POST /images/kids.jp2?rsiz=640,480&roff=320,240&fsiz=1280,1024 HTTP/1.1
Host: post.jpeg.org
Content-type: image/jpt-stream
CRLF
```

F.3 Etablissement de session

Une session HTTP en mode session est établie au moyen du champ de requête de nouveau canal avec une valeur de "http", c'est-à-dire "cnew=http" dans le cadre d'une requête. Celle-ci est typiquement acheminée par protocole HTTP. La requête peut contenir une requête de fenêtre de visualisation qui devient la première requête dans le nouveau canal. La réponse à cette requête est renvoyée sur la connexion dans laquelle la requête a été faite.

Un client peut ouvrir une connexion HTTP et émettre une requête qui contient l'en-tête HTTP "Connection: keep-alive". Cela est utile pour des sessions efficaces, mais ce n'est ni nécessaire ni suffisant pour avoir une session. Une même connexion HTTP peut être utilisée pour du trafic vers différentes cibles dans différents canaux, ou même pour du trafic non JPIP, par exemple les requêtes visant des fichiers HTML. Une requête JPIP qui fait partie d'une session peut arriver sur des connexions HTTP autres que la connexion HTTP utilisée pour demander et émettre le nouveau canal, bien que cela soit déconseillé.

F.4 Réponses

F.4.1 Introduction

Chaque composant d'une réponse conforme à l'Annexe D peut être encapsulé sous forme de portion d'une réponse HTTP légale.

NOTE – La réponse HTTP est définie dans le § 6 du document RFC 2616 comme suit:

```

Response = Status-Line           ; HTTP Section 6.1
          0* ( ( general-header   ; HTTP Section 4.5
              / response-header  ; HTTP Section 6.2
              / entity-header )  ; HTTP Section 7.1
              CRLF
          [ message-body ]       ; HTTP Section 7.2

```

Les réponses JPIP transportées au-dessus du protocole HTTP doivent être des réponses HTTP légales, avec d'autres limitations sur certaines parties de la réponse HTTP comme décrit dans les paragraphes suivants.

F.4.2 Code de description d'état et phrase de cause

Tous les codes de description d'état énumérés en D.1.3 peuvent être utilisés directement comme codes de description d'état HTTP. Par ailleurs, un serveur distant fournissant le protocole JPIP au-dessus du protocole HTTP peut utiliser tout code de description d'état HTTP jugé utile, par exemple 402.

Toutes les valeurs de phrase de cause fournies en D.1.3 peuvent être utilisées directement comme phrase de cause HTTP. La phrase de cause doit être appropriée au code de description d'état. Un serveur distant fournissant le protocole JPIP au-dessus du protocole HTTP peut utiliser toute phrase de cause HTTP jugée utile, par exemple "Paiement requis".

F.4.3 Informations d'en-tête

F.4.3.1 En-têtes JPIP

Les lignes d'en-tête extraites de D.2 doivent être incluses comme élément "entity-header" dans la réponse HTTP sans modification.

F.4.3.2 Utilisation de l'en-tête HTTP "Accept"

Un serveur distant fournissant le protocole JPIP au-dessus du protocole HTTP peut utiliser une ligne d'en-tête HTTP "Accept:" découverte dans une requête afin de déterminer le type de réponse JPIP. Si la requête contient un paramètre d'interrogation "type=", le type de retour doit être un des types énumérés dans le paramètre de type. Si la requête contient à la fois un paramètre d'interrogation "type=" et une ligne d'en-tête "Accept:", le serveur distant peut utiliser les priorités spécifiées dans la ligne "Accept:" afin de faire un choix parmi les types spécifiés dans le paramètre d'interrogation "type=". Si aucun paramètre d'interrogation "type=" n'est présent dans la requête, le serveur distant peut sélectionner un type de retour pris en charge par le serveur JPIP sous-jacent à partir de la liste de types contenue dans la ligne "Accept:".

F.4.3.3 Utilisation d'en-tête de commande de cache

Noter que les caches contenus dans les serveurs intermédiaires HTTP diffèrent des caches et modèles de cache du protocole JPIP.

Toute requête JPIP contenant un champ de requête de nouveau canal fait partie d'une session et de telles réponses ne peuvent généralement pas être mises en mémoire cache par des serveurs intermédiaires HTTP. De même, toute réponse qui contient un en-tête de réponse de nouveau canal fait également partie d'une session. Dans les deux cas, la réponse du serveur distant devrait inclure une ligne d'en-tête HTTP "Cache-Control:" avec la valeur "no-cache".

F.4.3.4 Utilisation d'un en-tête de type de contenu

Un serveur distant fournissant le protocole JPIP au-dessus du protocole HTTP devrait inclure une ligne d'en-tête "Content-type:" indiquant le type de données contenues dans le corps. Le plus souvent, il s'agit du flux "image/jpp-stream" ou "image/jpt-stream".

F.4.3.5 Utilisation de l'en-tête de réacheminement

L'en-tête de réacheminement HTTP peut être utile afin d'informer un client que les ressources se sont déplacées ou devraient être atteintes à partir d'un autre serveur local.

Noter que la réponse JPIP définit également une façon d'effectuer un réacheminement. La réponse JPIP devrait être préférée dans une session.

F.4.4 Corps

Les messages conformes à l'Annexe D doivent être inclus en tant que corps de la réponse HTTP. Noter qu'une réponse HTTP doit avoir un mécanisme permettant de déterminer la longueur de la réponse. Si le serveur distant ne prévoit pas d'interrompre une réponse, il peut offrir ces informations avec une ligne d'en-tête HTTP "Content-Length". La méthode préférée d'indication de la longueur consiste à utiliser la ligne d'en-tête HTTP "Transfer-Encoding: chunked" puis à offrir le corps en fragments de taille déterminée par le serveur distant et spécifiée avant chaque fragment. L'indication de la fin d'une réponse par fermeture de la connexion HTTP est déconseillée.

F.5 Caractéristiques additionnelles du protocole HTTP

F.5.1 Utilisation de la méthode "HEAD" du protocole HTTP

Clients et serveurs distants du protocole JPIP ne sont pas tenus d'utiliser ou de prendre en charge la méthode HTTP "HEAD". Un serveur distant choisissant d'implémenter la méthode "HEAD" doit le faire comme spécifié dans le § 9.4 du document RFC 2616. En particulier, "la méthode HEAD est identique à la requête GET sauf que le serveur distant ne doit pas renvoyer de corps de message dans la réponse".

Les clients peuvent juger utile d'émettre des requêtes HTTP "HEAD" afin de déterminer si le serveur distant va modifier l'un quelconque des paramètres de requête comme spécifié dans l'Annexe D. Les clients ne devraient pas émettre de requête HTTP "HEAD" avec des champs d'interrogation de modèle de cache car cela peut conduire le serveur distant à mettre à jour son modèle de cache.

Noter qu'un client qui souhaite mettre à jour le modèle de cache du serveur distant sans recevoir de réponse peut utiliser le champ de requête de longueur maximale de réponse.

Les serveurs peuvent refuser toute(s) requête(s) "HEAD". A la différence des requêtes HTTP "HEAD" typiques dont la satisfaction nécessite relativement peu d'effort de la partie d'un serveur distant, certaines réalisations de serveur JPIP pourraient avoir à obtenir des données à partir de plusieurs emplacements dans une cible logique, à calculer la nature de la réponse puis à rejeter le corps de la réponse afin de satisfaire une requête "HEAD".

F.5.2 Utilisation de la méthode des options HTTP

Clients et serveurs distants du protocole JPIP ne sont pas tenus d'utiliser ou de prendre en charge la méthode HTTP "OPTIONS".

F.5.3 Usage de la balise d'entité (ETag)

Noter que le protocole HTTP définit le mécanisme de balise d'entité (ETag) qui est semblable au champ de requête JPIP d'identificateur de cible en ce sens qu'il sert à repérer des modifications dans une ressource. Si à la fois une balise d'entité et un identificateur de cible sont associés à une ressource, il est recommandé que la balise ETag définie par protocole HTTP soit modifiée chaque fois que l'identificateur de cible est modifié.

F.5.4 Utilisation du codage de transfert fragmenté

Etant donné que les réponses contenant des données comprimées peuvent être très grandes et ainsi prendre longtemps pour leur transmission, il est important de pouvoir effectuer un arrêt au milieu de la transmission. A moins que la ligne "Transfer-Encoding: chunked" ne soit spécifiée, les requêtes HTTP doivent spécifier la longueur totale du corps dans un en-tête "Content-Length:" ou doivent indiquer la fin des données par fermeture de la connexion. Aucune de ces indications n'est souhaitable dans un protocole interactif, car il peut être nécessaire d'arrêter la réponse actuelle et d'envoyer plus de données sur la même connexion pour une nouvelle réponse.

Noter que le § 19.4.6 du document RFC 2616 offre un algorithme pour supprimer le codage de transfert fragmenté.

Noter également que le codage de transfert fragmenté peut être utile dans le protocole JPIP quand il est acheminé sur des protocoles autres que HTTP.

F.6 Protocole HTTP et champ de requête de longueur (pour information)

Avec un canal de retour HTTP, le serveur distant ne reçoit pas de réaction continue à partir du client et peut facilement propulser une grande quantité de données dans le circuit: celles-ci doivent être entièrement reçues avant que les données d'une nouvelle fenêtre puissent être traitées. Afin de conserver la réactivité, les clients devraient utiliser le champ de requête de longueur maximale de réponse afin de réguler le flux de trafic et donc conserver la réactivité. Les clients auront généralement besoin d'implémenter leurs propres algorithmes de commande de débit afin d'adapter la longueur de requête à des conditions de réseau changeantes.

Annexe G

Utilisation du protocole JPIP avec requêtes HTTP et retours TCP

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

G.1 Introduction

Le protocole JPIP proprement dit est neutre par rapport aux mécanismes de transport sous-jacents en termes de requêtes émises par un client et de réponses des serveurs distants, sauf en ce qui concerne les requêtes de canal représentées par le champ de requête de nouveau canal ("cnew") (voir C.3.3) et par l'en-tête de réponse de nouveau canal ("JPIP-cnew") (voir D.2.3), où des détails spécifiques du transport doivent être communiqués. La présente Recommandation | Norme internationale définit deux transports spécifiques, qui sont identifiés par les chaînes "http" et "http-tcp" dans la chaîne de valeur associée aux requêtes de nouveau canal. Cette annexe offre des détails sur le second mode de transport, qui doit être désigné dans ce texte par l'abréviation "HTTP-TCP". Le premier mode de transport est désigné dans ce texte par l'abréviation "HTTP" et est décrit dans l'Annexe F.

Le transport HTTP-TCP utilise exactement les mêmes mécanismes que le transport HTTP afin d'envoyer au serveur distant des requêtes de client et afin de recevoir les en-têtes de réponse et les codes de description d'état du serveur distant. Cependant, les données de réponse du serveur distant (et non les en-têtes de réponse) sont acheminées sur une connexion TCP auxiliaire. Les informations transportées sur cette connexion TCP auxiliaire sont identiques à celles qui auraient été transportées en tant que corps d'entité d'une pure réponse HTTP, sauf qu'elles sont verrouillées en fragments, dont chacun a un numéro séquentiel.

Le client acquitte explicitement l'arrivée de chaque fragment par le renvoi de son numéro séquentiel vers le serveur distant sur le chemin de retour de la connexion TCP auxiliaire. Un des principaux avantages du transport HTTP-TCP est que le serveur distant reçoit une notification incrémentielle de l'arrivée de ses fragments de données de réponse au moyen de ce mécanisme d'acquiescement par le client. Cela permet au serveur distant de gérer le flux de données de façon à conserver la réactivité et l'efficacité du réseau.

Toutes les requêtes envoyées par transport HTTP doivent être codées comme spécifié par la norme HTTP.

G.2 Requêtes émises par un client

Les requêtes sont acheminées sur le canal primaire exactement comme les requêtes HTTP. Elles ont exactement la même forme que les requêtes émises sur un canal qui utilise le transport HTTP décrit dans l'Annexe F. En particulier, les requêtes HTTP "GET" et "POST" peuvent toutes les deux être utilisées.

G.3 Etablissement de session

G.3.1 Ouverture de canal

Un nouveau canal peut être établi vers un serveur distant JPIP par émission d'une requête qui contient le champ de requête de nouveau canal (voir C.3.3). Par exemple, une telle requête pourrait être émise au moyen du protocole HTTP, bien qu'elle puisse également être envoyée à un serveur distant spécifique du protocole JPIP utilisant tout mécanisme de transport approprié. Si la réponse du serveur distant (par l'en-tête de réponse de nouveau canal en D.2.3) indique qu'un nouveau canal a été créé afin de fonctionner avec le transport HTTP-TCP, le client doit établir la connexion TCP auxiliaire au moyen du numéro auxiliaire de point d'accès renvoyé au moyen de l'en-tête de réponse de nouveau canal. Par ailleurs, la requête qui comprenait le champ de requête de nouveau canal est ensuite traitée comme si elle avait été émise dans le canal nouvellement créé de transport HTTP-TCP, c'est-à-dire que les données de réponse produites par cette requête doivent être renvoyées au moyen de la connexion TCP auxiliaire, dès que celle-ci a été établie.

Afin d'établir la connexion TCP auxiliaire, le client envoie une requête de connexion TCP au serveur distant identifié au moyen de l'en-tête de réponse de nouveau canal, au point d'accès identifié par l'en-tête de réponse de nouveau canal. Le client envoie alors immédiatement une unique ligne de texte ASCII, composée de la nouvelle chaîne d'identification de canal, suivie par deux paires consécutives de commandes CR-LF. Il s'agit de la seule communication en mode texte acheminée sur la connexion TCP auxiliaire.

Le client attend alors de recevoir les données de réponse du serveur distant sur la connexion TCP auxiliaire. Ces données de réponse ne peuvent pas être vides, car chaque requête émise dans un canal de transport HTTP-TCP doit avoir un flux de données de réponse qui comporte au moins le message EOR (voir D.3). Voir G.4 pour plus de détails à ce sujet.

G.3.2 Verrouillage par le serveur des données de réponse

Toutes les données de réponse envoyées par le serveur distant au moyen de la connexion TCP auxiliaire doivent être verrouillées en fragments. Chaque fragment se compose d'un en-tête de fragment de 8 octets, suivi par le corps de fragment qui contient les données de réponse du serveur distant, comme représenté dans la Figure G.1. Le premier mot de 2 octets de l'en-tête de fragment contient un entier gros-boutiste non signé représentant la longueur totale du fragment, y compris le mot de longueur proprement dit. Le contenu des 6 octets restants de l'en-tête de fragment n'est pas défini par la présente Recommandation | Norme internationale. Ces octets peuvent être utilisés pour une signalisation additionnelle propre au serveur distant. Le client renverra la totalité des 8 octets de l'en-tête de fragment dans ses messages d'acquiescement de fragment.

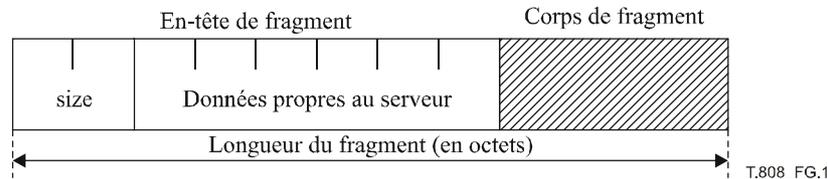


Figure G.1 – Structure des données de réponse au-dessus de la connexion http-tcp

G.3.3 Acquiescement par le client des fragments de réponse du serveur distant

Dès réception d'un fragment de données de réponse du serveur sur la connexion TCP auxiliaire, le client doit renvoyer les 8 octets de l'en-tête de fragment au serveur distant sous forme de flux de données non verrouillées, au moyen du chemin de retour de la connexion TCP. Chaque fragment reçu doit être acquiescé en séquence.

G.4 Réponse des serveurs distants

En réponse à chaque requête émise par un client, le serveur distant renvoie à ce dernier un alinéa de réponse HTTP sur le canal primaire. Cet alinéa de réponse contient le code de description d'état, la phrase de cause et tous les en-têtes de réponse JPIP applicables et tous les en-têtes de réponse HTTP appropriés. Cependant, aucune donnée de réponse n'est renvoyée au moyen du canal primaire. C'est la raison pour laquelle il ne doit y avoir aucun corps d'entité HTTP dans une réponse HTTP-TCP. Ni l'en-tête de réponse HTTP "Content-length:" ni l'en-tête de réponse HTTP "Transfer-Encoding:" ne doit être utilisé.

Les données de réponse proprement dites sont acheminées sur le canal TCP auxiliaire, verrouillées en fragments comme décrit en G.3.2. Comme le transport HTTP-TCP ne peut être utilisé qu'avec des sessions et donc qu'avec des types de retour d'image à flux JPP ou JPT, les données de réponse se composent invariablement d'une séquence de messages à flux JPP ou JPT.

Les données de réponse résultant de chaque requête doivent se composer d'un nombre entier de fragments, c'est-à-dire qu'aucun fragment ne peut contenir de données de réponse produites en réponse à deux requêtes différentes.

La réponse à toutes les requêtes doit se terminer par un message EOR (voir D.3), même si les données de réponse auraient autrement été vides. Le message EOR est considéré comme faisant partie des données de réponse et est verrouillé en fragments de même que les messages réels en flux JPP ou JPT.

C'est-à-dire que chaque requête émise sur un canal JPIP transporté par protocole HTTP-TCP se traduit par la production d'au moins un fragment de réponse non vide à partir du serveur distant; cela signifie également que le dernier fragment produit en réponse à chaque requête se termine par le message EOR.

Noter qu'il n'y a aucune exigence réelle pour que les fragments de réponse transportés par protocole HTTP-TCP soient alignés sur les frontières de message.

G.5 Protocole TCP et champ de requête de longueur (pour information)

Il peut n'y avoir que peu de raison – ou aucune raison – d'utiliser le champ de requête de longueur maximale de réponse avec un canal de retour TCP lorsque le serveur distant est en mesure de réguler précisément le flux de données de réponse au client de façon à conserver la réactivité.

Annexe H

Utilisation du protocole JPIP avec des transports de remplacement

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

H.1 Introduction

La présente Recommandation | Norme internationale ne définit aucun protocole spécifique de transport autre que le transport "http" décrit dans l'Annexe F et le transport "http-tcp" décrit dans l'Annexe G. L'objet de cette annexe consiste à offrir des directives sur le déploiement du protocole JPIP sur des transports non fiables et offre une approche générique qui peut être appliquée à une grande variété de transports.

Lors de la mise au point de l'approche générale, il est utile de subdiviser les aspects de la communication en deux connexions de transport logiques, appelées la "connexion de requête" et la "connexion de données". Chaque connexion logique est censée offrir aussi bien un chemin de communication direct qu'un chemin de communication inverse. Les rôles joués par ces chemins sont les suivants:

- le chemin direct de connexion de requête sert à remettre au serveur distant des requêtes JPIP issues du client;
- le chemin inverse de connexion de requête sert au serveur distant à accuser réception de requêtes et à renvoyer des en-têtes de réponse au client;
- le chemin direct de connexion de données sert à remettre au client des messages en flux JPIP à partir du serveur distant;
- le chemin inverse de connexion de données sert au client à accuser réception de messages en flux JPIP à partir du serveur distant.

Le lecteur commentera que ces rôles sont cohérents avec ceux qui sont joués par les chemins direct et inverse de communication des deux canaux TCP utilisés par le transport "http-tcp" décrit dans l'Annexe G. Le matériel exposé dans cette annexe peut en fait être interprété comme une extension du transport "http-tcp" à des modes de transport non fiables. Noter cependant que, bien que cette annexe soit décrite en termes de deux connexions logiques différentes, rien ne s'oppose à ce que la communication ne puisse pas être transportée sur une même connexion de transport.

Finalement, l'on part du principe que chaque connexion logique offre un des deux types suivants de services:

- a) un service fiable en mode flux, comme celui qui est offert par le protocole TCP;
- b) un service non fiable en mode paquet, comme celui qui est offert par le protocole UDP. Dans ce cas, les paquets peuvent arriver dans le désordre ou pas du tout et le dialogue d'acquiescement doit être implémenté explicitement de façon à déterminer qu'un paquet est arrivé correctement.

Deux scénarios sont pris en considération dans la présente annexe. Dans le premier cas, le chemin de la connexion de requête est censé offrir un service fiable en mode flux, mais le chemin de la connexion de données n'est pas fiable. Dans le second cas, les deux chemins – de requête et de connexion – ne sont pas fiables. Il est utile de traiter ces deux scénarios de façon ordonnée.

H.2 Requêtes fiables avec données non fiables

Dans le présent paragraphe, la connexion de requête est fiable, c'est-à-dire que les requêtes arrivent au serveur distant en ordre et sans perte tandis que les réponses des serveurs distants sont reçues par le client en ordre et de nouveau sans perte. Dans ce cas, les champs de requête et les en-têtes de réponse peuvent être communiqués exactement comme dans le protocole "http-tcp" et, en fait, le protocole HTTP est recommandé pour le transport des requêtes et des en-têtes de réponse. Un protocole de transport de ce genre pourrait, par exemple, être nommé "http-udp", mais de telles spécificités sont hors du domaine d'application de la présente annexe.

Les messages en flux JPIP, y compris le message EOR (voir D.3), doivent être subdivisés en paquets et être acheminés sur la connexion de données non fiable (par exemple par protocole UDP). Le client doit accuser réception de chacun de ces paquets en renvoyant l'en-tête de paquet au serveur distant, ce qui permet à celui-ci d'estimer les conditions du réseau et de déterminer si une retransmission du paquet est justifiée. Si la fenêtre de visualisation du client a changé, le serveur distant peut décider de ne pas retransmettre un paquet non acquitté.

Les directives générales suivantes devraient être observées lors de la construction de protocoles de transport de ce type:

- a) Chaque requête devrait inclure un champ de requête d'identificateur de requête (voir C.3.5).
- b) Pour chaque requête, il doit y avoir un message EOR correspondant, même si aucun message en flux JPIP n'est envoyé en réponse à la requête. Cette exigence s'applique également dans le cas du transport "http-tcp".

- c) Chaque paquet de connexion de données construit par le serveur distant doit se composer d'un nombre entier de messages en flux JPIP et/ou de messages EOR. En outre, le premier message à flux JPIP dans chaque paquet doit contenir un en-tête complet, ne dépendant pas de la répétition de l'identificateur de flux codé ou de composants de code de classe contenus dans un message précédent.
- d) Tous les messages en flux JPIP (non nécessairement des messages EOR) découverts dans un paquet de connexion de données doivent appartenir à la réponse issue d'une requête isolée et l'identificateur de requête correspondant doit être codé dans l'en-tête de paquet.
- e) Des messages EOR peuvent être découverts soit à la fin d'un paquet comportant la même valeur d'identificateur de requête que la requête dont la réponse doit être achevée, ou dans un bloc d'un ou de plusieurs messages EOR consécutifs, découverts au début du premier paquet suivant le dernier paquet comportant cet identificateur de requête. Cette politique permet de regrouper des messages EOR correspondant à une réponse vide ou à plusieurs réponses vides consécutives (par exemple à cause de requêtes préemptées) dans le premier paquet de la réponse non vide subséquente.
- f) En plus de la valeur d'identificateur de requête, chaque en-tête de paquet devrait inclure un numéro séquentiel de paquet. Le compteur de paquets en séquence est réglé à 0 pour le premier paquet associé à toute valeur particulière d'identificateur de requête. Les paquets suivants qui ont la même valeur d'identificateur de requête possèdent des numéros séquentiels consécutifs. Cette politique permet à un client d'identifier tout message EOR qui pourrait ne pas avoir été reçu à cause d'une perte de paquet. Il est important qu'un client soit en mesure d'associer des requêtes avec des données de réponse de façon à synchroniser, avec l'état de son propre cache, les effets d'instructions de manipulation du modèle de cache dans le serveur distant.
- g) Les clients doivent accuser réception de chaque paquet par l'envoi de messages d'acquittement au serveur distant sur le chemin de connexion de données ouvert pour la réponse. Chaque message d'acquittement devrait contenir une copie de l'en-tête de paquet reçu, mais pourrait théoriquement contenir des informations additionnelles. Le client peut, à sa discrétion, regrouper des messages d'acquittement concernant plusieurs paquets lors de la construction de paquets d'acquittement. Cependant, une agrégation excessive peut affecter la fiabilité d'estimation, pouvant être effectuée par les serveurs distants, des statistiques de réseau.
- h) Le serveur distant n'est pas obligé de retransmettre un quelconque paquet non acquitté et les clients ne devraient pas s'attendre à la retransmission de paquets manquants. Un serveur distant intelligent pourrait, par exemple, choisir de retransmettre les paquets non acquittés selon leur applicabilité à la fenêtre de visualisation actuelle.

H.3 Requêtes non fiables avec données non fiables

Le présent paragraphe est orienté vers les transports où aussi bien la requête que la connexion de données ne sont pas fiables. Les directives relatives à la connexion de données sont exactement comme décrit en H.2 pour le cas où les données sont acheminées de façon non fiable. Avec une connexion de requête non fiable, cependant, il est possible qu'une ou plusieurs requêtes se perdent ou arrivent dans le désordre au serveur distant. Le protocole JPIP est bien adapté au traitement de cette situation, car les serveurs distants ont la possibilité de préempter de précédentes requêtes quand une nouvelle requête arrive.

Les directives générales suivantes devraient être observées lors du traitement de requêtes non fiables, en plus de celles qui sont énumérées en H.2 pour les connexions de données non fiables.

- a) Chaque paquet de requête devrait inclure un en-tête désignant la valeur de l'identificateur de requête.
- b) Chaque paquet de requête devrait également comprendre un numéro séquentiel, acheminant des informations suffisantes pour déterminer que tous les paquets associés à une requête ont été reçus.
- c) Souvent, les serveurs distants peuvent simplement ignorer les paquets de requête manquants quand une nouvelle requête arrive. A cette fin, le serveur distant n'a qu'à envoyer des messages EOR sur la connexion de données, en indiquant que la requête manquante a été préemptée immédiatement. Il n'est pas nécessaire qu'un message d'acquittement soit envoyé en réponse à des paquets de requête. Il n'est pas nécessaire que des en-têtes de réponse soient envoyés en réponse à des requêtes qui vont être immédiatement préemptées parce que tout ou partie des paquets de requête ont été perdus.
- d) Pour chaque requête qui arrive en totalité au serveur distant, celui-ci devrait envoyer un ou plusieurs paquets de réponse qui indiquent l'identificateur de requête et qui comprennent d'éventuels en-têtes de réponse. Cela est vrai même si la requête arrive après l'envoi de la réponse à d'éventuelles requêtes ultérieures (par exemple parce que certains paquets de la requête ont été indûment retardés). Cela offre au client un mécanisme permettant de déterminer qu'une requête importante a été reçue par le serveur distant.

- e) Certains types de requêtes doivent être traités par le serveur distant afin d'éviter une perte de synchronisation avec le client. Les plus importantes sont les requêtes qui comprennent des champs de manipulation soustractive du modèle de cache. Afin de permettre au serveur distant de détecter de telles requêtes sans avoir à mettre en série tout le flux de requête, les en-têtes de paquet de requête devraient inclure les deux champs suivants:
- 1) un fanion indiquant que le paquet appartient à une requête qui doit être traitée avant les requêtes ultérieures;
 - 2) l'identificateur de requête associé à la plus récente requête pour laquelle le fanion mentionné ci-dessus en e) 1) a été activé.

Si le serveur distant ne reçoit pas un ou plusieurs paquets d'une requête avec le fanion e) 1) activé (c'est-à-dire que des requêtes avec condition e) 2) arrivent et que la requête avec fanion e) 1) fait défaut), ce serveur doit rester au repos jusqu'à ce que le client retransmette ces paquets.

H.4 Syntaxe de requête et de réponse

La syntaxe de requête et de réponse décrite dans les Annexes C et D devrait être suivie lors de la conception de nouveaux transports pour le protocole JPIP. Cependant, il est permis de mettre au point des représentations binaires équivalentes de divers champs de requête et en-têtes de réponse.

H.5 Etablissement de session

Le champ de requête de nouveau canal (voir D.2.3) et l'en-tête de réponse correspondant peuvent servir à créer des canaux associés aux protocoles de transport autres que "http" et "http-tcp" décrits normativement dans la présente Recommandation | Norme internationale. A cette fin, de nouveaux noms de protocole de transport peuvent être enregistrés auprès du bureau d'enregistrement défini dans l'Annexe J. La procédure de création de canaux pour de nouveaux transports devrait suivre les conventions générales déjà décrites pour le transport "http-tcp". En particulier, les en-têtes de réponse pour la requête qui crée le nouveau canal devraient être renvoyés par le transport qui a été utilisé pour créer le canal, alors que les données de réponse devraient être acheminées au moyen du transport du nouveau canal.

Annexe I

Indexation de fichiers JPEG 2000 pour le protocole JPIP

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

I.1 Introduction (pour information)

La Rec. UIT-T T.800 | ISO/CEI 15444-1:2004 et d'autres normes définissent une famille de formats de fichier JPEG 2000 qui utilise un syntaxe commune, dont l'élément fondamental est le conteneur appelé *boîte*. Cette annexe définit de nouvelles boîtes de format de fichier contenant des informations d'indexation, dont l'inclusion dans les fichiers de la famille JPEG 2000 peut faciliter le déploiement de ces fichiers dans un système JPIP, en permettant aux lecteurs de fichier de localiser dans ces fichiers les éléments qui sont requis afin de construire des images de façon progressive.

En particulier, ces boîtes peuvent être utiles:

- à une implémentation du protocole JPIP du côté du serveur distant;
- à un client téléaccédant à une image au moyen d'un protocole très simple, lui donnant accès à des étendues spécifiées d'octets du fichier.

Cette annexe définit des boîtes d'indexation correspondant aux informations du niveau des fichiers comme de celui des flux codés. Ces boîtes peuvent être rangées dans les catégories suivantes:

- La superboîte d'indice de flux codé (*cidx*) désigne les informations au niveau du flux codé correspondant aux classes de segment de données d'en-tête principal, d'en-tête de pavé, de pavé et de district en flux JPP ou JPT. Elle contient une boîte de localisateur de flux codé (*cptr*) pointant sur le flux codé indexé, une boîte de résumé (*manf*) résumant le reste du contenu et des boîtes de table d'indices qui sont: la boîte de table d'indices d'en-tête (*mhix*), la superboîte de table d'indices d'éléments de pavé (*tpix*), la superboîte de table d'indices d'en-tête de pavé (*thix*), la superboîte de table d'indices de paquet de district (*ppix*) et la superboîte de table d'indices d'en-tête de paquet (*phix*). Les boîtes de tables d'indices correspondent aux différents types de données de flux codé représentés par les classes de segment de données dans les flux JPP et JPT définis dans l'Annexe A. Les boîtes de table d'indices qui sont des superboîtes contiennent les boîtes d'indices de série tabulaire de fragments (*faix*) ou la table d'indices d'en-tête énumérant les éléments actuels du flux codé. Chaque superboîte de table d'indices d'en-tête, de paquet de district et d'en-tête de paquet contient également une boîte de résumé.
- La superboîte d'indice de fichier (*fidx*) désigne les informations au niveau du fichier correspondant à la classe de segment de métadonnées du flux JPP ou JPT. A moins qu'elle n'indexe le niveau sommital du fichier, auquel cas elle est appelée *boîte radicale* d'indexation de fichier, cette superboîte contient une boîte de localisateur de fichier (*fptr*) pointant sur la superboîte indexée. Elle peut contenir des boîtes intermédiaires (*prxy*) représentant le contenu de l'indice de fichier ou de la superboîte indexée.
- La boîte de localisateur d'indice (*iptr*) pointe sur un indice radical de fichier, permettant de découvrir son emplacement.

La Figure I.1 ci-dessous décrit un exemple de fichier JPEG 2000 contenant des boîtes d'indexation PIP:

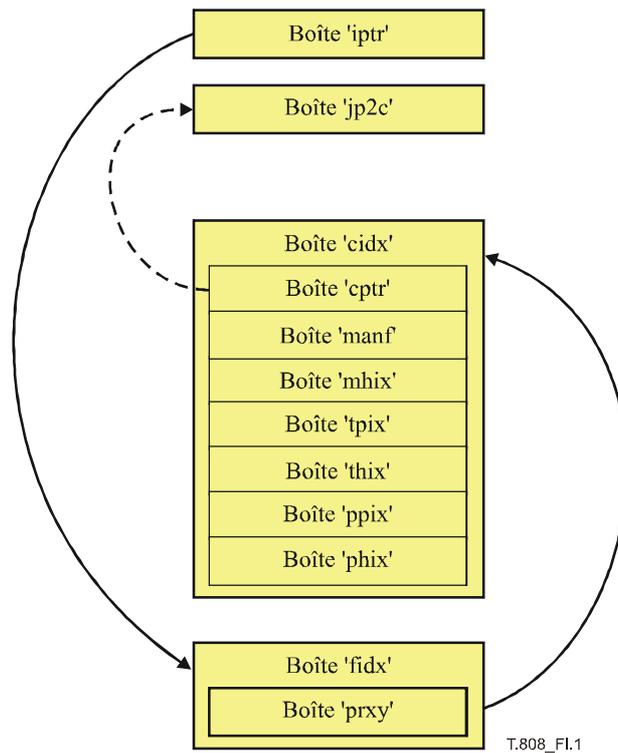


Figure I.1 – Exemple partiel de fichier JPEG 2000 contenant des boîtes d'indexation JPIP

I.2 Identification de l'utilisation de boîtes d'indexation JPIP dans la liste de compatibilités des formats de fichier JPEG 2000

Les fichiers qui contiennent une ou plusieurs des boîtes d'indexation définies dans la présente Recommandation | Norme internationale peuvent contenir un champ CL¹ dans la boîte de type de fichier (comme défini dans l'Annexe I de la Rec. UIT-T T.800 | ISO/CEI 15444-1) avec la valeur 'jpip' (0x6a70 6970).

I.3 Boîtes d'indexation définies

I.3.1 Généralités

Le Tableau I.1 énumère toutes les boîtes définies dans le cadre de la présente Recommandation | Norme internationale. Pour le placement et les restrictions concernant chaque boîte, voir le paragraphe approprié définissant cette boîte.

Le Tableau I.1 est informatif. Les définitions normatives de chaque boîte sont contenues dans les sections individuelles désignées dans le tableau.

Tableau I.1 – Boîtes d'indexation définies (pour information)

Nom de boîte	Type	Superboîte	Commentaires
Boîte d'indexation de flux codé (I.3.2)	'cidx' (0x6369 6478)	Oui	Cette boîte contient des informations d'indexation concernant un flux à codage JPEG 2000.
Boîte de localisateur de flux codé (I.3.2.2)	'cptr' (0x6370 7472)	Non	Cette boîte pointe sur un flux à codage JPEG 2000.
Boîte de table d'indices d'en-tête (I.3.2.4.3)	'mhix' (0x6D68 6978)	Non	Cette boîte spécifie un indice des segments marqueurs dans l'en-tête principal d'un flux codé ou dans les en-têtes des éléments d'un pavé.
Boîte de table d'indices d'élément de pavé (I.3.2.4.4)	'tpix' (0x7470 6978)	Oui	Cette boîte spécifie l'emplacement et la longueur de chaque pavé dans le flux codé.
Boîte de table d'indices d'en-tête de pavé (I.3.2.4.5)	'thix' (0x7468 6978)	Oui	Cette boîte spécifie l'emplacement et la longueur de chaque partie du flux codé qui est nécessaire pour construire des en-têtes pour chaque pavé en vue du décodage correct de données de paquets de district.

Tableau I.1 – Boîtes d'indexation définies (pour information)

Nom de boîte	Type	Superboîte	Commentaires
Boîte de table d'indices de paquet de district (I.3.2.4.6)	'ppix' (0x7070 6978)	Oui	Cette boîte spécifie l'emplacement et la longueur de paquets dans le flux codé.
Boîte de table d'indices d'en-tête de paquet (I.3.2.4.7)	'phix' (0x7068 6978)	Oui	Cette boîte spécifie l'emplacement et la longueur des entêtes de paquet dans le flux codé.
Boîte de résumé (I.3.2.3)	'manf' (0x6D61 6E66)	Non	Cette boîte résume les boîtes qui la suivent immédiatement et qui lui sont adjacentes, à l'intérieur de sa boîte englobante ou du fichier situé au même niveau que la boîte de résumé.
Boîte d'indexation d'une série tabulaire de fragments (I.3.2.4.2)	'faix' (0x6661 6978)	Non	Cette boîte spécifie l'emplacement et la longueur des éléments d'un flux codé.
Boîte d'indexation de fichiers (I.3.3)	'fidx' (0x6669 6478)	Oui	Cette boîte peut servir à trouver d'autres indices et des données arbitraires dans le fichier.
Boîte de localisateur de fichier (I.3.3.2)	'fptr' (0x6670 7472)	Non	Cette boîte pointe sur une boîte indexée.
Boîte intermédiaire (I.3.3.3)	'prxy' (0x7072 7879)	Non	Cette boîte représente, dans une boîte d'indexation de fichier, une boîte située ailleurs dans le fichier.
Boîte de localisateur d'indice (I.3.4)	'iptr' (0x6970 7472)	Non	Cette boîte pointe sur la boîte radicale d'indexation d'un fichier.

I.3.2 Boîte d'indexation de flux codé (superboîte)

I.3.2.1 Généralités

La boîte d'indexation de flux codé contient des informations d'indexation sur un flux à codage JPEG 2000. Le type d'une boîte d'indexation de flux codé doit être 'cidx' (0x6369 6478). Le contenu d'une boîte d'indexation de flux codé doit être le suivant (Figure I.2):

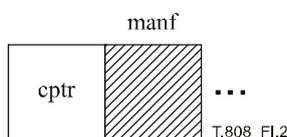


Figure I.2 – Organisation du contenu d'une boîte d'indexation de flux codé

- cptr:** Boîte de localisateur de flux codé. Cette boîte pointe sur le flux codé qui est indexé par la boîte d'indexation de flux codé. Sa structure est spécifiée en I.3.2.2.
- manf:** Boîte de résumé. Cette boîte résume les tables d'indices qui la suivent à l'intérieur de la boîte d'indexation de flux codé. Sa structure est spécifiée en I.3.2.3.

I.3.2.2 Boîte de localisateur de flux codé

La boîte de localisateur de flux codé pointe sur un flux à codage JPEG 2000. Le type d'une boîte de localisateur de flux codé doit être 'cptr' (0x6370 7472). Le contenu d'une boîte de localisateur de flux codé doit être le suivant (Figure I.3):

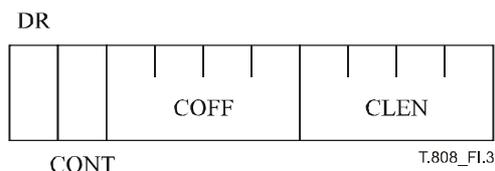


Figure I.3 – Organisation du contenu d'une boîte de localisateur de flux codé

- DR:** Référence des données. Ce champ spécifie l'emplacement du flux codé, ou de la boîte de table de fragments qui le représente. Si sa valeur est 0, le flux codé ou sa boîte de table de fragments existe dans le fichier actuel. Sinon, la grandeur désigne une entrée dans la boîte de référence des données contenue dans le fichier actuel. Dans ce cas, l'entrée de référence de données identifiée par DR

indique la ressource qui contient le flux codé ou la boîte de table de fragments. Ce champ est mémorisé par les 2 octets d'un entier gros-boutiste non signé.

- CONT:** Type de conteneur. Ce champ est mémorisé par les 2 octets d'un entier gros-boutiste non signé. Les valeurs définies dans la présente Recommandation | Norme internationale sont décrites dans le Tableau I.2.
- COFF:** Décalage du flux codé. Ce champ spécifie l'emplacement du flux codé ou de la boîte de liste de fragments, selon le cas, se rapportant au début du fichier ou à la ressource identifiée par DR. Ce champ est mémorisé par les 8 octets d'un entier gros-boutiste non signé.
- CLEN:** Longueur du flux codé. Ce champ spécifie la longueur du flux codé ou de la boîte de liste de fragments, selon le cas. Ce champ est mémorisé par les 8 octets d'un entier gros-boutiste non signé.

Tableau I.2 – Valeurs de type de conteneur

CONT	Signification
0	Le flux codé entier apparaît comme une étendue continue d'octets contenus dans son fichier ou sa ressource. Dans ce cas, les valeurs de décalage et de longueur indiquées ici se rapportent au flux codé proprement dit. Noter que le flux codé peut aussi bien être contenu dans une boîte de flux codé contigu, mais les valeurs de décalage et de longueur se rapportent au flux codé proprement dit, commençant à partir du marqueur SOC et se terminant immédiatement après le marqueur EOC.
1	Le flux codé est fragmenté et les valeurs d'emplacement et de longueur se rapportent à la boîte de liste de fragments (y compris son en-tête de boîte) décrivant les emplacements et longueurs de chacun des fragments qui représentent le flux codé. Noter que tous les emplacements et longueurs subséquents sont exprimés par rapport au début du flux codé, tel qu'il apparaîtrait après reconstitution de tous les fragments identifiés dans la boîte de liste de fragments.
Toutes autres valeurs	Champ réservé pour utilisation par l'ISO.

I.3.2.3 Boîte de résumé

La boîte de résumé résume les boîtes qui la suivent immédiatement et qui lui sont adjacentes, dans sa boîte englobante ou dans son fichier situé au même niveau que la boîte de résumé.

NOTE – La boîte de résumé peut servir à faciliter l'accès aléatoire à ces boîtes subséquentes, telles que les boîtes d'indexation qui la suivent à l'intérieur d'une boîte d'indexation de flux codé.

Le type d'une boîte de résumé doit être 'manf' (0x6D61 6E66). Le contenu de la boîte de résumé doit être le suivant (Figure I.4).

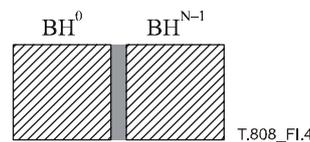


Figure I.4 – Organisation du contenu d'une boîte de résumé

- BHⁱ:** En-tête de boîte. Ce champ contient l'en-tête complet de la nième boîte qui suit immédiatement la boîte de résumé. La longueur de ce champ est de 16 octets si la valeur du champ "LBox" contenu dans cet en-tête de boîte est 1, ou de 8 octets sinon.

Le nombre de boîtes, N , dont les en-têtes sont contenus dans la boîte de résumé, est déterminé par la longueur de la boîte de résumé. Quand il est utilisé à l'intérieur d'une boîte de table d'indices de paquet de district ou à l'intérieur d'une boîte d'indices d'en-tête de paquet, N est le nombre de composants du flux codé.

A l'intérieur d'une boîte d'indexation de flux codé, d'une boîte de table d'indices d'en-tête de pavé, d'une boîte de table d'indices de paquet de district ou d'une boîte de table d'indices d'en-tête de paquet, une boîte de résumé doit inclure toutes les boîtes qui lui font suite, jusqu'à la fin de la boîte englobante.

I.3.2.4 Table d'indices

I.3.2.4.1 Généralités

La boîte d'indexation de flux codé peut contenir une table d'indices pour chacune des sortes suivantes de données de flux codé: en-tête principal, éléments de pavé, en-têtes de pavé, paquets (de district) et en-têtes de paquet. Chaque table d'indices est un type de boîte différent. Il doit y avoir au plus une seule de chaque sorte de table dans une boîte d'indexation de flux codé.

Les boîtes de table d'indices d'élément de pavé, de table d'indices de paquet de district et de table d'indices d'en-tête de paquet sont des superboîtes contenant des boîtes de table d'indices de série tabulaire de fragments. La boîte de table d'indices d'en-tête de pavé est une superboîte contenant des boîtes de table d'indices d'en-tête. Nous définirons d'abord, ci-dessous, la boîte d'indexation d'une série tabulaire de fragments puis les boîtes de tables d'indices.

I.3.2.4.2 Boîte d'indexation d'une série tabulaire de fragments

La boîte d'indexation d'une série tabulaire de fragments énumère les emplacements et les longueurs des éléments d'un flux codé. Elle est utilisée dans les superboîtes de table d'indices d'élément de pavé, de table d'indices de paquet de district et de table d'indices d'en-tête de paquet.

Le type d'une boîte d'indexation d'une série tabulaire de fragments doit être 'faix' (0x6661 6978). Le contenu de la boîte d'indexation d'une série tabulaire de fragments doit être le suivant (Figure I.5):

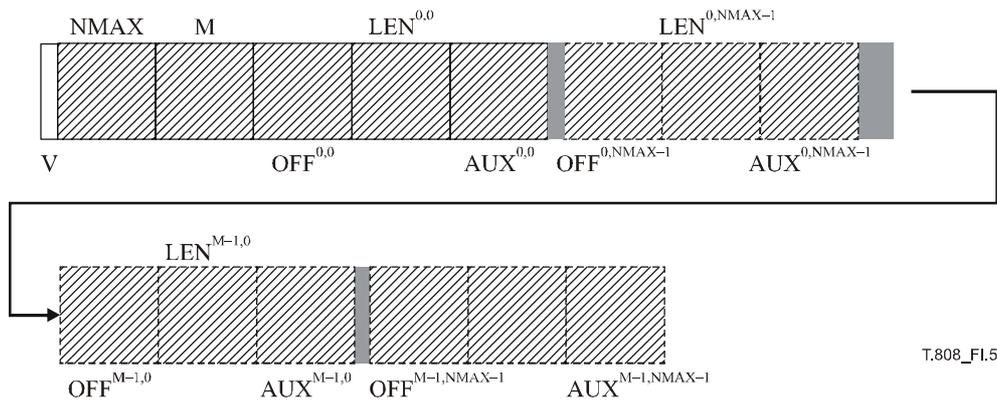


Figure I.5 – Organisation du contenu d'une boîte d'indexation d'une série tabulaire de fragments

- V:** Version. Ce champ est codé par 1 octet d'un entier non signé. Les valeurs définies dans la présente Recommandation | Norme internationale sont décrites dans le Tableau I.3.
- NMAX:** Nombre maximal d'éléments valides contenus dans toute rangée de la série tabulaire. Quand il est utilisé à l'intérieur d'une table d'indices de flux codé, NMAX est le nombre maximal d'éléments qui sera spécifié pour un pavé quelconque.
- M:** Nombre de rangées de la série tabulaire. Quand il est utilisé à l'intérieur d'une table d'indices de flux codé, M est le nombre de pavés.
- OFF^{ij}:** Décalage. Ce champ spécifie le décalage en octets (par rapport au début du flux codé) du *j*^{ème} élément dans la rangée *i* de la série tabulaire.
- LEN^{ij}:** Longueur. Ce champ spécifie la longueur en octets du *j*^{ème} élément dans la rangée *i* de la série tabulaire.
- AUX^{ij}:** Auxiliaire. Ce champ spécifie des informations auxiliaires sur le *j*^{ème} élément contenu dans la rangée *i* de la série tabulaire. La valeur de ce champ doit être zéro sauf autorisation contraire par la superboîte contenant cette boîte. Toutes les valeurs différentes de zéro dans ce champ sont réservées.

Alors que toutes les rangées de la série tabulaire spécifiée dans la boîte d'indexation de série tabulaire de fragments doivent être mémorisées avec NMAX éléments, l'objet décrit par cette rangée peut avoir un plus petit nombre d'éléments à spécifier. Dans ce cas, pour toute rangée *i* contenant *J* éléments valides où *J* est inférieur à NMAX, les valeurs de OFF^{i,j} à OFF^{i,NMAX-1} et de LEN^{i,j} à LEN^{i,NMAX-1} doivent être réglées à zéro.

Tableau I.3 – Valeurs de version

CONT	Signification
0	NMAX, M et tous les champs OFF ^{ij} et LEN ^{ij} sont codés par les 4 octets d'un entier gros-boutiste non signé et les champs AUX ^{ij} ne sont pas présents.
1	NMAX, M et tous les champs OFF ^{ij} et LEN ^{ij} sont codés par les 8 octets d'un entier gros-boutiste non signé et les champs AUX ^{ij} ne sont pas présents.
2	Tous les champs autres que V sont codés par les 4 octets d'un entier gros-boutiste non signé.
3	NMAX, M et tous les champs OFF ^{ij} et LEN ^{ij} sont codés par les 8 octets d'un entier gros-boutiste non signé et tous les champs AUX ^{ij} sont codés par les 4 octets d'un entier gros-boutiste non signé.
Toutes autres valeurs	Champ réservé pour utilisation par l'ISO.

I.3.2.4.3 Boîte de table d'indices d'en-tête

La boîte de table d'indices d'en-tête désigne l'en-tête principal d'un flux codé ou les en-têtes d'élément d'un pavé, en indiquant la longueur totale de l'en-tête principal ou la longueur totale du premier élément de pavé ainsi que les emplacements et longueurs des segments marqueurs contenus dans l'en-tête. Tous les segments marqueurs doivent être inclus, sauf que le segment de marqueur SOT peut être omis pour les en-têtes d'élément de pavé qui se composent seulement des marqueurs SOT et SOD. Les segments marqueurs n'ont pas besoin d'être énumérés dans l'ordre où ils apparaissent dans le flux codé. La boîte de table d'indices d'en-tête peut apparaître seulement à l'intérieur d'une boîte d'indexation de flux codé. Au niveau sommital, elle désigne un flux codé et doit apparaître au plus 1 fois. A l'intérieur d'une boîte de table d'indices d'en-tête de pavé, elle désigne des en-têtes d'élément de pavé.

NOTE – L'objectif est d'offrir un moyen efficace de sauter dans l'en-tête des informations de pointage qui ne sont pas requises pour explorer efficacement le fichier mais qui peuvent encombrer inutilement l'en-tête. L'énumération de multiples segments marqueurs ayant consécutivement les mêmes codes marqueurs dans la boîte de table d'indices d'en-tête permettra aux lecteurs de sauter les groupes de segments marqueurs qui ne les concernent pas.

Le type de boîte de table d'indices d'en-tête doit être 'mhix' (0x6D68 6978). Le contenu de la boîte de table d'indices d'en-tête doit être le suivant (Figure I.6):

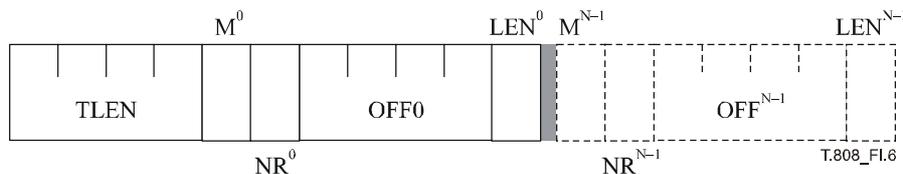


Figure I.6 – Organisation du contenu d'une boîte de table d'indices d'en-tête

- TLEN:** Longueur. Quand la boîte de table d'indices d'en-tête désigne un en-tête principal, ce champ spécifie la longueur totale de l'en-tête principal. Quand la boîte de table d'indices d'en-tête désigne des en-têtes d'élément de pavé, ce champ spécifie la longueur totale du premier en-tête d'élément de pavé. La valeur de ce champ est codée par les 8 octets d'un entier gros-boutiste non signé.
- Mⁱ:** Code marqueur. Ce champ spécifie le code marqueur qui commence le i^{ème} segment de marqueur énuméré dans cette boîte. La valeur de ce champ est codée par les 2 octets d'un entier gros-boutiste non signé.
- NRⁱ:** Nombre restant. Ce champ indique que (au moins) NRⁱ segments marqueurs ayant le même code marqueur Mⁱ sont énumérés immédiatement et consécutivement après le i^{ème} segment de marqueur dans cette liste. La valeur de ce champ est codée par les 2 octets d'un entier gros-boutiste non signé.
- OFFⁱ:** Décalage. Ce champ spécifie le décalage en octets, par rapport au début du flux codé, des paramètres de segment de marqueur (y compris le paramètre de longueur mais non le marqueur proprement dit) pour le i^{ème} segment de marqueur contenu dans cette liste. La valeur de ce champ est codée par les 8 octets d'un entier gros-boutiste non signé.
- LENⁱ:** Longueur. Ce champ spécifie la longueur en octets des paramètres de segment de marqueur (y compris les deux octets du paramètre de longueur mais non les deux octets du marqueur proprement dit) pour le i^{ème} segment de marqueur dans cette liste. La valeur de ce champ est codée par les 2 octets d'un entier gros-boutiste non signé et est identique à la valeur du paramètre de longueur contenu dans le segment de marqueur proprement dit.

Le nombre de segments marqueurs, N , énumérés dans la boîte de table d'indices d'en-tête, est déterminé par la longueur de la boîte de table d'indices d'en-tête.

I.3.2.4.4 Boîte de table d'indices d'élément de pavé (superboîte)

La boîte de table d'indices d'élément de pavé désigne les emplacements et longueurs de chaque pavé dans le flux codé, où chaque pavé commence avec son marqueur SOT et finit avec le dernier paquet du pavé.

Le type d'une boîte de table d'indices d'élément de pavé doit être 'tpix' (0x7470 6978). Le contenu de la boîte de table d'indices d'élément de pavé doit être le suivant (Figure I.7):

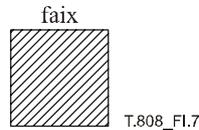


Figure I.7 – Organisation du contenu d'une boîte de table d'indices d'élément de pavé

faix: Boîte d'indexation d'une série tabulaire de fragments. Cette boîte énumère les emplacements et longueurs de tous les pavés dans le flux codé. Sa structure est spécifiée en I.3.2.4.2. Le *mième* rangée de cette table correspond au *mième* pavé dans le flux codé. Les entrées de cette rangée contiennent les emplacements et longueurs de tous les éléments de pavé contenus dans le pavé correspondant, dans l'ordre des flux codés. Si la boîte d'indexation d'une série tabulaire de fragments a un numéro de version égal à 2 ou 3, les champs auxiliaires spécifient pour chaque élément de pavé le plus petit nombre *n* tel que, dans tous les composants pour lesquels $(N_L - n)$ n'est pas négatif, le niveau de résolution $(N_L - n)$ et tous les niveaux de résolution inférieurs ont été complétés quand cet élément de pavé est combiné avec tous les éléments de pavé précédents du même pavé, où N_L est le nombre de niveaux de décomposition, qui peut varier composant par composant. Si aucun niveau de résolution de composant n'a été complété, la valeur du champ auxiliaire est égale à l'unité plus la valeur maximale de N_L dans tous les composants. La valeur zéro est atteinte quand toutes les résolutions de tous les composants ont été complétées. Etant donné que les résolutions n'apparaissent pas nécessairement en ordre dans un pavé, certains niveaux de résolution supérieurs à la valeur signalée par le champ auxiliaire peuvent avoir été complétés.

I.3.2.4.5 Boîte de table d'indices d'en-tête de pavé (superboîte)

La boîte de table d'indices d'en-tête de pavé désigne les en-têtes de chaque pavé, pour le décodage correct des données de paquets de district.

Le type d'une boîte de table d'indices d'en-tête de pavé doit être 'thix' (0x7468 6978). Le contenu de la boîte de table d'indices d'en-tête de pavé doit être le suivant (Figure I.8):

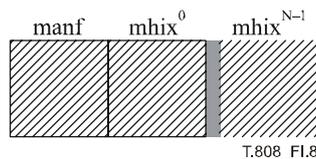


Figure I.8 – Organisation du contenu d'une boîte de table d'indices d'en-tête de pavé

Le nombre de boîte de table d'indices d'en-têtes, *N*, est le nombre de pavés.

- manf:** Boîte de résumé. Cette boîte résume les boîtes spécifiées par $mhix^i$ à l'intérieur de cette boîte de table d'indices d'en-tête de pavé. Sa structure est spécifiée en I.3.2.3.
- $mhix^i$:** Boîte de table d'indices d'en-tête. Cette boîte désigne les en-têtes de pavé pour le *ième* pavé. Sa structure est spécifiée en I.3.2.4.3.

I.3.2.4.6 Boîte de table d'indices de paquet de district (superboîte)

La boîte de table d'indices de paquet de district désigne les paquets dans le flux codé. Le type d'une boîte de table d'indices de paquet de district doit être 'ppix' (0x7070 6978). Le contenu de la boîte de table d'indices de paquet de district doit être le suivant (Figure I.9):

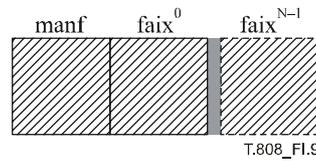


Figure I.9 – Organisation du contenu d'une boîte de table d'indices de paquet de district

Le nombre de boîtes de table d'indices de série tabulaire de fragments, N , ne doit pas être supérieur au nombre de composants du flux codé.

- manf:** Boîte de résumé. Cette boîte résume les boîtes spécifiées par faix^i à l'intérieur de cette boîte de table d'indices de paquet de district. Sa structure est spécifiée en I.3.2.3.
- faixⁱ:** La i ème boîte d'indexation d'une série tabulaire de fragments correspond au i ème composant d'image dans le flux codé. La m ième rangée de ce tableau correspond au m ième pavé dans le flux codé. Les entrées de cette rangée contiennent les emplacements et longueurs de tous les paquets contenus dans l'élément de pavé correspondant. Les paquets apparaissent consécutivement, dans l'ordre croissant des couches, dans leurs districts respectifs qui apparaissent dans l'ordre associé au numéro séquentiel s , défini en A.3.2.1. Cependant, l'ordre fixe des paquets n'est pas nécessairement le même que celui qui est spécifié dans tous les segments marqueurs COD/POC contenus dans le flux codé. La structure de la boîte d'indexation d'une série tabulaire de fragments est spécifiée en I.3.2.4.2.

Si les en-têtes de paquet sont imbriqués dans les segments marqueurs PPM ou PPT, les entrées correspondantes dans la série tabulaire de fragments ne se rapportent qu'à l'emplacement et à la longueur du corps de paquet, telle qu'elle apparaît à l'intérieur de son corps d'élément de pavé. Les entrées qui se rapportent à des paquets inexistant (soit parce que l'élément de pavé correspondant contient moins de paquets qu'un autre élément de pavé dans la même série tabulaire, ou parce que le flux codé a été tronqué avant l'instant auquel ce paquet aurait existé) devraient avoir leur champ d'emplacement réglé à zéro. Les entrées qui se rapportent à des paquets dont le corps est vide et dont l'en-tête se compose exactement de 1 octet, 0x80, peuvent être identifiées au moyen d'une valeur de longueur égale à zéro. De tels paquets apparaissent fréquemment dans les flux à codage JPEG 2000; les applications peuvent éviter la surcharge d'aller chercher explicitement de tels paquets dont le contenu est prévisible. Si le segment de marqueur COD correspondant spécifie que des marqueurs EPH doivent apparaître après chaque en-tête de paquet dans un pavé, la valeur spéciale de longueur égale à zéro doit être interprétée dans ce pavé comme signifiant que le paquet se compose des 0x80 octets, suivis par le marqueur EPH.

I.3.2.4.7 Boîte de table d'indices d'en-tête de paquet (superboîte)

La boîte de table d'indices d'en-tête de paquet désigne les en-têtes de paquet dans le flux codé. Le type de boîte de table d'indices d'en-tête de paquet doit être 'phix' (0x7068 6978). Le contenu de la boîte de table d'indices d'en-tête de paquet doit être le suivant (Figure I.10):

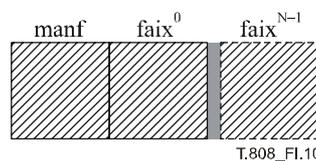


Figure I.10 – Organisation du contenu d'une boîte de table d'indices d'en-tête de paquet

Le nombre de boîtes de table d'indices de série tabulaire de fragments, N , ne doit pas être supérieur au nombre de composants du flux codé.

- manf:** Boîte de résumé. Cette boîte résume les boîtes spécifiées par faix^i à l'intérieur de cette boîte de table d'indices d'en-tête de paquet. Sa structure est spécifiée en I.3.2.3.
- faixⁱ:** La i ème boîte d'indexation d'une série tabulaire de fragments correspond au i ème composant d'image dans le flux codé. La m ième rangée dans cette table correspond au m ième pavé dans le flux codé. Les entrées de cette rangée contiennent les emplacements et longueurs de tous les en-têtes de

paquet contenus dans l'élément de pavé correspondant. Les en-têtes de paquet apparaissent consécutivement, dans l'ordre croissant des couches, dans leurs districts respectifs et les districts apparaissent dans l'ordre associé au numéro séquentiel s , défini en A.3.2.1. Cependant, l'ordre fixe des en-têtes de paquet n'est pas nécessairement le même que celui qui est spécifié dans tous les segments marqueurs COD/POC contenus dans le flux codé. La structure de la boîte d'indexation d'une série tabulaire de fragments est spécifiée en I.3.2.4.2.

Les entrées qui se rapportent aux en-têtes de paquet inexistantes (soit parce que l'élément de pavé correspondant contient moins de paquets qu'un autre élément de pavé dans la même série tabulaire, ou parce que le flux codé a été tronqué avant l'instant auquel cet en-tête de paquet aurait existé) devraient avoir leur champ d'emplacement réglé à zéro. Les entrées qui se rapportent aux paquets dont le corps est vide et dont l'en-tête se compose exactement de 1 octet, 0x80, peuvent être identifiées au moyen d'une valeur de longueur égale à zéro. De tels paquets apparaissent fréquemment dans les flux à codage JPEG 2000; les applications peuvent éviter la surcharge d'aller chercher explicitement de tels paquets, dont le contenu est prévisible. Si le segment de marqueur COD correspondant spécifie que des marqueurs EPH doivent apparaître après chaque en-tête de paquet dans un certain pavé, la valeur spéciale de longueur 0 doit être interprétée dans ce pavé comme indiquant que le paquet se compose de l'octet 0x80, suivi par le marqueur EPH.

I.3.3 Boîte d'indexation de fichiers (superboîte)

I.3.3.1 Généralités

La boîte d'indexation de fichier peut servir à trouver d'autres indices (en particulier, l'indice de flux codé correspondant à un certain flux codé) et à trouver des données arbitraires dans ce fichier.

Une boîte radicale d'indexation de fichier désigne le niveau sommital du fichier. Toute autre boîte d'indexation de fichier désigne une superboîte dans ce fichier. Il doit y avoir au plus une boîte d'indexation de fichier avec un certain domaine d'application (niveau sommital ou une certaine superboîte) dans un fichier donné.

Le type d'une boîte d'indexation de fichier doit être 'fidx' (0x6669 6478). Le contenu de la boîte d'indexation de fichier doit être le suivant (Figure I.11):

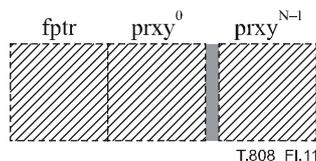


Figure I.11 – Organisation du contenu d'une boîte d'indexation de fichier

- fptr:** Boîte de localisateur de fichier. Une boîte radicale d'indexation de fichier ne doit pas comprendre cette boîte. Toute autre boîte d'indexation de fichier doit inclure cette boîte, qui doit pointer sur la superboîte indexée par la boîte d'indexation de fichier. La structure de la boîte de localisateur de fichier est définie en I.3.3.2.
- prxyⁱ:** Boîte intermédiaire. Cette boîte représente une boîte contenue dans la portion du fichier indexé par la boîte d'indexation de fichier. Une boîte radicale d'indexation de fichier ne doit inclure d'intermédiaires que pour les boîtes situées au niveau sommital du fichier. Toute autre boîte d'indexation de fichier ne doit inclure des intermédiaires que pour les boîtes situées au niveau sommital de la superboîte indexée par la boîte d'indexation de fichier. Les intermédiaires doivent apparaître dans le même ordre que les boîtes, mais toutes les boîtes n'ont pas besoin d'être représentées par des intermédiaires. La structure de la boîte intermédiaire est définie en I.3.3.3.

NOTE – Etant donné que la présence, l'absence, ou la séquence des boîtes dans le fichier est parfois importante, il peut être utile aux applications qu'aucune boîte située dans le domaine d'application de l'indice ne soit omise de celui-ci avant l'apparition de telles boîtes d'intermédiaire.

I.3.3.2 Boîte de localisateur de fichier

La boîte de localisateur de fichier pointe sur une boîte. Le type d'une boîte de localisateur de fichier doit être 'fptr' (0x6670 7472). Le contenu d'une boîte de localisateur de fichier doit être le suivant (Figure I.12):

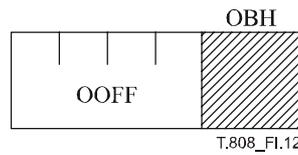


Figure I.12 – Organisation du contenu d'une boîte de localisateur de fichier

- OOFf:** Décalage original. Ce champ spécifie le décalage en octets (par rapport au début du fichier) de la boîte pointée par cette boîte de localisateur de fichier. La valeur de ce champ est codée par les 8 octets d'un entier gros-boutiste non signé.
- OBH:** En-tête de boîte original. Ce champ contient l'en-tête complet de la boîte pointée par cette boîte de localisateur de fichier. La longueur de ce champ est de 16 octets si la valeur du champ "LBox" contenu dans cet en-tête de boîte est 1, ou de 8 octets sinon.

I.3.3.3 Boîte intermédiaire

Dans une boîte d'indexation de fichier, la boîte intermédiaire représente une boîte située ailleurs dans le fichier en indiquant son emplacement et sa longueur, ainsi que l'emplacement et la longueur de tout indice relatif à cette boîte, avec un préfixe décrivant le contenu de la boîte.

Le type d'une boîte intermédiaire doit être 'prxy' (0x7072 7879). Le contenu de la boîte intermédiaire doit être le suivant (Figure I.13):

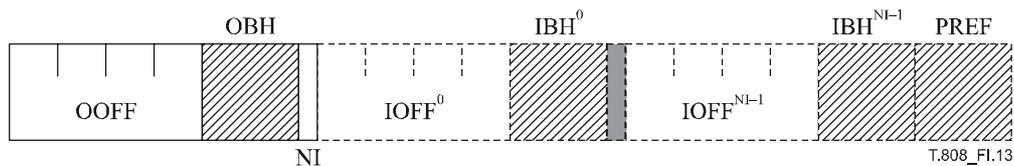


Figure I.13 – Organisation du contenu d'une boîte intermédiaire

- OOFf:** Décalage original. Ce champ spécifie le décalage en octets (par rapport au début du fichier) de la boîte représentée par cette boîte intermédiaire. La valeur de ce champ est codée par les 8 octets d'un entier gros-boutiste non signé.
- OBH:** En-tête de boîte original. Ce champ contient l'en-tête complet de la boîte représentée par cette boîte intermédiaire. La longueur de ce champ est de 16 octets si la valeur du champ "LBox" contenu dans cet en-tête de boîte est 1, ou de 8 octets sinon.
- NI:** Nombre d'indices. Ce champ indique le nombre de pointeurs sur indice inclus dans cette boîte intermédiaire. Chaque ensemble de champs subséquents $IOFF^i$ et IBH^i pointe soit sur un indice de fichier ou sur une boîte d'indexation de flux codé qui désigne la boîte représentée par cette boîte intermédiaire. Toutes les autres valeurs sont réservées. La valeur de ce champ est codée par 1 octet d'un entier non signé.
- $IOFF^i$:** Décalage d'indice. Ce champ contient le décalage en octets (par rapport au début du fichier) de la i ème boîte d'indexation. La valeur de ce champ est codée par les 8 octets d'un entier gros-boutiste non signé.
- IBH^i :** En-tête de boîte d'indexation. Ce champ contient l'en-tête complet de la i ème boîte d'indexation. La longueur de ce champ est de 16 octets si la valeur du champ "LBox" contenu dans cet en-tête de boîte est 1, ou de 8 octets sinon.
- PREF:** Préfixe. Ce champ contient un préfixe arbitraire des données contenues dans la boîte représentée par cette boîte intermédiaire. Il peut avoir toute longueur à partir de zéro jusqu'à la longueur du contenu de la boîte originale.

I.3.4 Boîte de localisateur d'indice

La boîte de localisateur d'indice pointe sur la boîte radicale d'indexation d'un fichier. Elle ne doit apparaître que si ce fichier contient une boîte radicale d'indexation de fichier. Le type d'une boîte de localisateur d'indice doit être 'iptr' (0x6970 7472). Le contenu d'une boîte de localisateur d'indice doit être le suivant (Figure I.14):

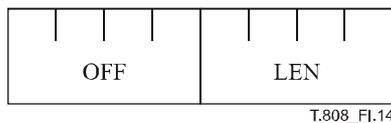


Figure I.14 – Organisation du contenu d'une boîte de localisateur d'indice

- OFF:** Décalage. Ce champ spécifie l'emplacement de la boîte radicale d'indexation de fichier par rapport au début du fichier. Ce champ est mémorisé par les 8 octets d'un entier gros-boutiste non signé.
- LEN:** Longueur. Ce champ spécifie la taille de la boîte radicale d'indexation de fichier. Ce champ est mémorisé par les 8 octets d'un entier gros-boutiste non signé.

I.4 Association des indices de flux codé avec des flux codés

Dans un fichier JP2, JPX ou JPM, la boîte d'indexation de flux codé doit apparaître au niveau sommital du fichier et la nième boîte d'indexation de flux codé doit correspondre au ième flux codé, également au niveau sommital du fichier. La boîte de localisateur de flux codé contenue dans la boîte d'indexation de flux codé indique également le flux codé qui est désigné par la boîte d'indexation de flux codé.

I.5 Restrictions d'emplacement (pour information)

Quelques restrictions d'emplacement ont été imposées aux boîtes définies dans la présente annexe. Elles peuvent être placées à la fin du fichier au besoin; cela est susceptible d'être pratique quand un fichier non indexé sera ultérieurement indexé. Cependant, il peut être utile de placer la boîte de localisateur d'indice à proximité du début du fichier, de préférence immédiatement après des boîtes tenues d'être dans un groupe adjacent au début du fichier (comme après la boîte de type de fichier dans un fichier JP2 ou après les boîtes d'exigences du lecteur dans un fichier JPX), où cette boîte de localisateur peut facilement être découverte par les lecteurs de ce fichier. Afin de minimiser le mouvement des boîtes de fichier, lors de l'adjonction de cette boîte et facultativement lors de l'adjonction d'un code 'jpip' à la liste de compatibilités contenue dans la boîte de type de fichier, une boîte libre (définie dans la Rec. UIT-T T.801 | ISO/CEI 15444-2, Annexe M.11.20) pourrait être utilisée comme générique afin de remplacer la boîte de localisateur dans un fichier restant à indexer.

Annexe J

Enregistrement des extensions à la présente Recommandation | Norme internationale

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

J.1 Introduction relative à l'enregistrement

L'enregistrement est le processus consistant à ajouter des extensions à la présente Recommandation | Norme internationale lorsqu'elle aura été publiée. Dans la présente Recommandation | Norme internationale, de nombreuses capacités peuvent être étendues au moyen d'un enregistrement. Le présent paragraphe désigne les éléments qui peuvent être étendus par enregistrement, le processus par lequel des capacités peuvent être enregistrées et le processus par lequel l'Organisme d'enregistrement publiera ces extensions. Seuls les éléments qui sont spécifiés dans le présent paragraphe peuvent être étendus par enregistrement.

J.2 Eléments d'enregistrement

Le processus d'enregistrement est composé des éléments suivants.

- **Organisme d'enregistrement:** entité organisationnelle chargée de réviser, de tenir à jour, de distribuer et de jouer le rôle de point de contact pour toutes les activités associées à l'enregistrement. L'Organisme d'enregistrement doit être déterminé.
- **Requérant:** organisation ou personne qui demande que l'élément soit enregistré.
- **Commission d'examen:** entité organisationnelle qui approuve l'enregistrement d'un élément proposé. Elle est composée d'un Comité ad hoc désigné par le Président de la Commission d'examen. La Commission d'examen doit être le sous-groupe de travail ISO/CEI JTC 1/SC 29/WG 1.
- **Président de la Commission d'examen:** personne chargée de vérifier que chaque élément candidat est pris en considération. Il communique avec le requérant au moyen de l'Organisme d'enregistrement. Le Président de la Commission d'examen doit être celui du sous-groupe de travail ISO/CEI JTC 1/SC 29/WG 1.
- **Essai:** justificatif que la Commission d'examen devrait utiliser afin de déterminer si un élément de soumission devrait être enregistré.
- **Élément de soumission:** il s'agit de la proposition d'enregistrement. Chaque proposition doit inclure le nom de l'élément à étendre, la balise/l'identité proposée pour l'extension et un justificatif/objet pour l'extension.

J.3 Critères d'évaluation d'enregistrement

La Commission d'examen doit évaluer toutes les soumissions fondées sur les critères suivants:

- Est-ce que l'extension correspond à un besoin déjà satisfait par la norme ou par d'autres extensions?
- Est-ce que l'extension a été suffisamment définie?
- Est-ce que l'extension correspond à un besoin général (par exemple des applications de flux vidéo en général) ou un besoin spécifique du vendeur (par exemple une certaine implémentation de vidéo par écoulement de flux)?

J.4 Eléments qui peuvent être étendus par enregistrement

J.4.1 Boîtes étendues à l'intérieur d'une boîte générique

Les nouveaux types de boîtes, qui seront utilisés dans le champ "ExtendedBoxList" contenu dans la boîte générique (A.3.6.3), doivent être enregistrables. Une proposition visant à enregistrer un nouveau type de boîte doit contenir une définition complète de cette boîte (type et contenu de la boîte), des instructions sur le moment où un serveur distant peut inscrire cette boîte à l'intérieur d'une boîte générique et des instructions sur ce qu'un client peut faire quand il rencontre une boîte générique représentant cette boîte.

J.4.2 Contexte du flux codé

De nouvelles valeurs du paramètre "context-range", afin de demander des flux codés spécifiques au moyen du champ de requête de contexte du flux codé (C.4.7), doivent être enregistrables. Une proposition visant à enregistrer un nouveau paramètre "context-range" doit contenir une définition complète de la valeur, des instructions sur la façon dont le serveur distant doit appliquer cette valeur sur les flux codés qui sont disponibles dans la cible logique, et des

instructions sur la façon dont le serveur distant doit répondre dans l'en-tête de réponse concernant le contexte de flux codé.

J.4.3 Transport par canal

De nouveaux transports par canal (Annexe H) doivent être enregistrables. Une proposition visant à enregistrer un nouveau transport par canal doit contenir une définition complète du transport, y compris l'identificateur à utiliser pour ce transport par canal.

J.4.4 Préférences

Les nouvelles préférences du client doivent être enregistrables. Cela implique de nouveaux ensembles de préférences (nouvelles valeurs du paramètre "related-pref-set" comme spécifié en C.10.2.1), ou de nouvelles options pour un ensemble de préférences existant ou enregistré. Une proposition visant à enregistrer une nouvelle option de préférence ou un nouvel ensemble de préférences doit contenir une définition complète de la syntaxe, la signification de nouvelles options et des instructions sur la façon dont le serveur distant doit répondre quand il agit conformément à cette préférence.

J.5 Processus d'enregistrement

Le processus d'enregistrement est le suivant:

- a) un requérant crée un élément candidat pour enregistrement;
- b) l'élément candidat est soumis à l'Organisme d'enregistrement;
- c) l'Organisme d'enregistrement transmet l'élément candidat au Président de la Commission d'examen;
- d) le Président de la Commission d'examen distribue l'élément candidat à la Commission d'examen et organise des réunions, des appels téléphoniques, etc., selon ce qui convient à l'examen de l'élément;
- e) la Commission d'examen doit évaluer toutes les soumissions. Si le texte de la soumission ne satisfait pas aux exigences, il doit être renvoyé au requérant pour clarification. La préférence sera donnée aux solutions qui sont plus générales. Les solutions proposées qui sont hautement spécifiques du vendeur peuvent être renvoyées au requérant afin qu'il les rende plus générales et plus applicables à l'industrie dans le sens large;
- f) en cas d'approbation, le Président transmet celle-ci à l'Organisme d'enregistrement qui avise l'ISO et le requérant puis met à disposition l'élément enregistré ou publié;
- g) en cas de désapprobation, le Président prépare un document de réponse en indiquant la raison pour laquelle l'élément a été écarté; puis il transmet ce document à l'Organisme d'enregistrement qui en avise le requérant.

J.6 Intervalles de temps pour le processus d'enregistrement

La Commission d'examen doit répondre à toutes les requêtes visant des enregistrements dans les sept mois à partir de la date de soumission. Dans cet intervalle, la Commission d'examen convoquera une réunion officielle de l'ISO/CEI JTC 1/SC 29/WG 1 afin d'évaluer la proposition, de prendre une décision et de rédiger la réponse.

Annexe K

Exemples d'application

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

K.1 Introduction

Cette annexe présente certains exemples informatifs décrivant des réalisations du protocole JPIP.

K.2 Utilisation du protocole JPIP avec des flux codés dans d'autres formats de fichier

Le protocole JPIP peut servir à accéder à des flux à codage JPEG 2000 mémorisés dans des formats de fichier autres que ceux de la famille JPEG 2000. Par exemple, les fichiers DICOM aussi bien que PDF ont la capacité de contenir des flux à codage JPEG 2000. Dans un environnement client/serveur, certaines procédures non spécifiées dans la présente Recommandation | Norme internationale peuvent servir à localiser le flux à codage JPEG 2000. Les requêtes et réponses JPIP peuvent être utilisées sur l'objet une fois que le flux codé est localisé. Le champ de requête de sous-cible est justement destiné à une telle situation. En variante, un serveur distant pourrait donner accès aux flux codés au moyen d'une localisation URL différente.

K.3 Techniques d'implémentation par éléments de pavé

K.3.1 Détermination par le serveur distant des éléments de pavé applicables dans le cas d'une requête de fenêtre de visualisation

Pour la communication au moyen d'éléments de pavé, le mappage d'un fenêtre de visualisation sur un ensemble de pavés est simple. La région recherchée de l'image est convertie en "unités de grille de référence". Les portions XTsiz et YTsiz du segment de marqueur SIZ sont utilisées afin de déterminer quels pavés sont en intersection avec la fenêtre de visualisation.

NOTE – Bien que, sur la grille de référence, tous les pavés aient les mêmes dimensions, ils n'ont pas nécessairement – après décomposition en sous-bandes – les mêmes dimensions sur la grille de référence sous-échantillonnée. Un pavé en intersection avec la fenêtre de visualisation – même un pavé englobé complètement dans cette fenêtre – peut n'apporter aucun échantillon à la fenêtre de visualisation au plus bas niveau de résolution; cependant, les implémentations n'ont pas besoin de tirer parti de ce cas de figure, en omettant d'emblée le pavé de la réponse.

Le niveau de résolution et la qualité servent à déterminer les pavés requis. La boîte de table d'indices d'élément de pavé, si disponible, peut servir à obtenir des informations sur l'emplacement d'éléments de pavé dans le flux codé et (si des champs auxiliaires sont inclus) sur l'achèvement du niveau de résolution à l'intérieur des éléments de pavé. Les segments marqueurs SOT indiquent également les indices de pavé et d'élément de pavé ainsi que le nombre d'octets contenus dans chaque élément de pavé. A partir du flux codé, les octets appropriés, correspondant aux éléments de pavé qui doivent être envoyés, sont transmis au client. Si la fenêtre de visualisation change et si les éléments de pavé applicables à cette fenêtre changent également, alors seuls les éléments de pavé applicables qui n'ont pas déjà été envoyés doivent l'être afin de mettre à jour l'image affichée.

K.3.2 Décodage d'une image à partir de messages en flux JPT renvoyés

Le protocole JPIP spécifie des mécanismes permettant de communiquer des données d'image comprimée et des métadonnées entre un client et un serveur distant. Les mécanismes permettant au client d'afficher les données renvoyées ne sont pas spécifiés et, en fait, varieront largement d'une application à une autre. Le présent paragraphe donne des informations sur la façon d'obtenir des échantillons de composant à partir de données renvoyées.

Une application cliente, qui a reçu toutes les données d'en-tête principal (indiquées par le segment de données d'en-tête complété apparaissant dans un message de réponse pour le segment de données d'en-tête 0), peut concaténer ce segment de données avec des éléments de pavé complétés à partir de segments de données de pavé afin de constituer un flux légal à codage JPEG 2000. Ce flux codé peut être fourni à un décodeur JPEG 2000 conforme et le résultat peut ainsi être affiché. Aux fins de l'efficacité, un client peut évidemment choisir d'offrir les paramètres de fenêtre de visualisation à un décodeur intelligent en association avec le flux codé, de façon que seules les portions requises pour la fenêtre de visualisation actuelle soient affichées.

K.3.3 Signalisation auxiliaire pour éléments de pavé

Les Tableaux K.1 et K.2 décrivent l'utilisation de champs auxiliaires dans des messages étendus de segment de données de pavé et dans la boîte de table d'indices d'élément de pavé.

NOTE – Dans cet exemple, la définition de r diffère par rapport à celle qui a été utilisée ailleurs dans la présente Recommandation | Norme internationale, mais elle est cohérente avec l'Annexe B de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004.

Le Tableau K.1 décrit un simple cas dans lequel tous les éléments de pavé d'un pavé à résolution progressive ont le même nombre de niveaux de décomposition et dans lequel les frontières de message (dans le cas du segment de données) ou de pavé (dans le cas de la boîte d'indexation) apparaissent seulement entre chaque niveau successif de résolution.

Tableau K.1 – Exemple de l'utilisation de champs auxiliaires dans un cas simple

Numéro séquentiel de message dans le segment de données, ou numéro d'élément de pavé dans le pavé	Niveau de résolution r	$n = N_L - r$	Valeur auxiliaire
0	0	2	2
1	1	1	1
2	2	0	0

Le Tableau K.2 décrit un cas plus compliqué dans lequel le nombre de niveaux de décomposition varie selon chaque élément de pavé. Un commentaire est fait dans la dernière colonne du tableau dès la première apparition de chaque nouvelle valeur auxiliaire. Ce cas correspond à un pavé extrait d'une image à trois composants dans un ordre de progression RC... Par exemple, l'ordre de progression LRCP dans une seule couche, ou l'ordre de progression RPCL avec un seul district dans le pavé. Les frontières de message (dans le cas du segment de données) ou d'élément de pavé (dans le cas de la boîte d'indexation) apparaissent entre chaque composant de chaque niveau de résolution ainsi qu'entre niveaux de résolution. Les composants 0 et 1 ont deux niveaux de décomposition ($N_L = 2$) et le composant 2 a un seul niveau de décomposition ($N_L = 1$).

Tableau K.2 – Exemple de l'utilisation de champs auxiliaires dans un cas plus compliqué

Numéro séquentiel de message dans le segment de données, ou numéro d'élément de pavé dans le pavé	Indice de composant c	Niveau de résolution r	$n = N_L - r$	Valeur auxiliaire	Commentaire
0	0	0	2	3	Aucun niveau n'est complet
1	1	0	2	2	$n = 2$ maintenant complet
2	2	0	1	2	
3	0	1	1	2	
4	1	1	1	1	$n = 1$ maintenant également complet
5	2	1	0	1	
6	0	2	0	1	
7	1	2	0	0	Tous niveaux maintenant complets

K.4 Techniques d'implémentation par districts

K.4.1 Détermination par le serveur distant de districts applicables dans le cas d'une requête de fenêtre de visualisation

Quand la communication implique le type de média à flux JPP, le serveur distant convertit la région d'image demandée par le client en un ensemble de districts qui se rapportent à la requête. La première partie de ce processus implique la conversion des paramètres fx , fy , sx , sy , ox et oy fournis par les champs de requête de longueur de trame, de taille de région et de décalage de région, en paramètres de flux codé de longueur de trame, de taille de région et de décalage fx' , fy' , sx' , sy' , ox' et oy' , pour chaque flux codé applicable. Cette conversion s'effectue de la même façon pour les deux services en mode district et en mode pavé. Elle est fondée sur les équations C-1 et C-2, éventuellement modifiées conformément aux équations C-3 et C-4. Le présent paragraphe décrit comment un serveur distant devrait déterminer les districts qui se rapportent à la région définie par les paramètres fx' , fy' , sx' , sy' , ox' et oy' , dans un certain flux codé.

Soit r l'entier non négatif dans l'équation C-1 qui a été utilisé par le serveur distant afin de trouver les paramètres fx' et fy' , sur la base de la requête du client. Comme mentionné à propos de cette équation, la variable r est très facilement interprétée comme étant le nombre de niveaux de transformation DWT de résolution la plus élevée qui ont été rejetés, bien que r soit autorisé à dépasser le nombre réel de niveaux de transformation DWT qui sont disponibles pour un

quelconque élément de pavé. Il est pratique d'appliquer d'abord la région décrite par sx' , sy' , ox' et oy' sur la grille de haute résolution du flux codé. Cela fournit une région dont le coin gauche supérieur est indiqué par $(E_1^{\text{reg}}, E_2^{\text{reg}})$ et dont le coin inférieur droit est indiqué par $(F_1^{\text{reg}} - 1, F_2^{\text{reg}} - 1)$, où:

$$E_1^{\text{reg}} = X\text{Osiz} + 2' \cdot ox', \quad E_2^{\text{reg}} = Y\text{Osiz} + 2' \cdot oy', \quad F_1^{\text{reg}} = E_1^{\text{reg}} + 2' \cdot sx', \quad \text{et} \quad F_2^{\text{reg}} = E_2^{\text{reg}} + 2' \cdot sy'$$

Le serveur distant n'a besoin de considérer que les pavés qui sont en intersection avec cette région sur la grille de haute résolution du flux codé. Pour chacun de ces pavés, le serveur distant n'a besoin de considérer que les composants d'image qui sont demandés par le client, comme décrit à propos des champs de requête de composant et de contexte de flux codé. Pour chaque composant de pavé considéré, indiqué par t et c , soit $D_{t,c}$ être le nombre de niveaux de transformation DWT qui ont servi à comprimer ce composant de pavé. Si $D_{t,c} \geq r$, le serveur distant devrait rejeter tous les districts appartenant aux plus hauts niveaux de résolution du composant de pavé r . Sinon, il devrait rejeter tous les districts appartenant aux plus hauts niveaux de résolution du composant de pavé $D_{t,c}$, en laissant seulement les districts qui représentent la plus basse sous-bande LL de ce composant de pavé.

Pour chaque district qui reste après le rejet des pavés, des composants et des niveaux de résolution mentionnés ci-dessus, le serveur distant devrait vérifier si les blocs codés qui appartiennent à ce district contribuent à la reconstruction de la région définie par $E_1^{\text{reg}}, E_2^{\text{reg}}$ et $F_1^{\text{reg}}, F_2^{\text{reg}}$ sur la grille de haute résolution du flux codé. Un bloc codé contribue à cette région si l'un quelconque de ses échantillons affecte la reconstruction d'un échantillon de composant d'image à résolution entière dont les coordonnées (x,y) satisfont aux relations suivantes:

$$E_1^{\text{reg}} \leq X\text{Rsiz}^c \cdot x < F_1^{\text{reg}} \quad \text{et} \quad E_2^{\text{reg}} \leq Y\text{Rsiz}^c \cdot y < F_2^{\text{reg}}$$

où $X\text{Rsiz}^c$ et $Y\text{Rsiz}^c$ représentent les facteurs de sous-échantillonnage horizontal et vertical pour le composant correspondant, c , dans le segment de marqueur SIZ du flux codé.

Il est important de tenir compte du fait que la reconstruction d'un composant d'image à résolution entière implique une synthèse d'ondelette, ce qui est un processus à expansion intrinsèque. Ainsi, la région à laquelle un quelconque district contribue chevauche généralement les régions auxquelles ses districts adjacents contribuent. Le serveur distant devrait être disposé à tenir compte de ces effets d'expansion de la transformation en ondelette lors de la détermination des districts qui se rapportent à une requête de client.

Le § 10.6.4 de la référence [11]: "JPEG2000: Bases fondamentales, normes et procédés de compression d'image" décrit un mode de calcul des échantillons d'une quelconque sous-bande qui contribuent à une région donnée sur la grille de haute résolution du flux codé. A partir des régions de sous-bande, il est très simple de déduire les blocs codés contributeurs et donc les districts.

K.4.2 Décodage d'une image à partir de messages renvoyés en flux JPP

Le protocole JPIP spécifie des mécanismes permettant de communiquer des données et métadonnées d'image comprimées entre un client et un serveur distant. Les mécanismes permettant au client d'afficher les données renvoyées ne sont pas spécifiés et peuvent même varier largement d'une application à une autre.

K.5 Transcriptions en protocole JPIP

K.5.1 Introduction

Dans les exemples de transcription suivants, le texte qui suit les symboles "<<<" au début d'une ligne est envoyé du client au serveur distant; le texte qui suit les symboles ">>>" au début d'une ligne est envoyé du serveur distant au client; et le texte qui suit les symboles "--" est un commentaire qui n'est pas réellement transmis. Les commentaires peuvent indiquer que certaines des données transmises ne sont pas représentées.

K.5.2 Utilisation du protocole HTTP

La transcription suivante montre cinq requêtes envoyées du client au serveur distant et la réponse du serveur distant.

La première requête demande le fichier JP2, appelé "phoenix.jp2". Le premier flux codé dans le fichier est demandé, une longueur maximale est insérée dans la réponse, un identificateur de cible est demandé, les données sont appelées à être renvoyées en tant que flux JPP et l'établissement d'une session au-dessus du protocole HTTP est demandé. Aucune fenêtre et donc aucune donnée d'image n'est demandée.

ISO/CEI 15444-9:2005 (F)

Le serveur distant répond en fournissant un identificateur de cible pour l'image et un identificateur pour le canal qui vient d'être ouvert. La ligne d'en-tête commençant par "JPIP-cnew" indique un nouveau chemin qui peut servir à accéder au fichier d'image. La valeur du chemin "jpip" peut être un chemin conduisant à un programme d'interface CGI dans le serveur distant, conçu de façon à traiter toutes les commandes interactives du protocole JPIP. Certaines données extraites du fichier sont renvoyées dans le corps: elles deviendront les boîtes de format de fichier et peut-être l'en-tête principal du premier flux codé.

La 2ème requête du client utilise le nouveau chemin "jpip.cgi" et l'identificateur de canal permettant d'identifier l'image recherchée (aucun nom d'image ou identificateur de cible n'est nécessaire). Cette requête spécifie également une certaine fenêtre recherchée.

La réponse à la 2ème requête indique que la fenêtre de visualisation a été modifiée et qu'une plus petite fenêtre centrée dans la fenêtre de visualisation demandée doit être renvoyée. Le serveur distant commence à renvoyer les données pour cette fenêtre.

Avant réception de la réponse complète à la 2ème requête, le client émet une 3ème requête. Le client a ajusté sa fenêtre de visualisation à la taille spécifiée par le serveur distant.

Celui-ci continue à répondre à la 2ème requête pendant un moment, puis commence une réponse à la 3ème requête. Pendant cette réponse, le client émet une 4ème requête avec une région légèrement différente. Le serveur distant continue à répondre à la 3ème requête pendant un moment puis commence à répondre à la 4ème requête.

Le client attend jusqu'à ce que la 4ème réponse soit terminée, puis émet une requête visant à mettre fin à la session et à la connexion HTTP. Il n'y a aucune donnée de réponse représentée dans ce cas, lorsque la connexion est fermée.

```
<< GET /phoenix.jp2?stream=0&len=2000&tid=0&type=jpp-stream&cnew=http
HTTP/1.1
<< Host: dst-m
<<
>> HTTP/1.1 200 OK
>> JPIP-tid: 281B6E135135BBC0BC588452AC9B73C5
>> JPIP-cnew: cid=JPH_033C38BE48115AC9,path=jpip.cgi,transport=http
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>>
>> 102
-- 258 octets de données binaires
>> 0
>>
<< GET /jpip.cgi?fsiz=834,834&roff=0,0&rsiz=834,790&comps=0-
2&stream=0&len=2000&cid=JPH_033C38BE48115AC9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> HTTP/1.1 200 OK, avec modifications
>> JPIP-roff: 120,114
>> JPIP-rsiz: 593,561
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>>
>> 393
-- 915 octets de données binaires
<< GET /jpip.cgi?fsiz=834,834&roff=120,114&rsiz=593,561&comps=0-
2&stream=0&len=2000&cid=JPH_033C38BE48115AC9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> 3f9
-- 1017 octets de données binaires
>> 0
>>
>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
```

```

>> Content-Type: image/jpp-stream
>>
>> 359
-- 857 octets de données binaires
<< GET /jpip.cgi?fsiz=834,834&roff=309,297&rsiz=121,86&comps=0-
2&stream=0&len=3906&cid=JPH_033C38BE48115AC9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> 234
-- 564 octets de données binaires
>> 3d0
-- 976 octets de données binaires
>> 24f
-- 591 octets de données binaires
>> 0
>>
>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>>
>> 3b2
-- 946 octets de données binaires
>> 400
-- 1024 octets de données binaires
>> 263
-- 611 octets de données binaires
>> 356
-- 854 octets de données binaires
>> 209
-- 521 octets de données binaires
>> 0

<< GET /jpip.cgi?cclose=JPH_033C38BE48115AC9&len=0 HTTP/1.1
<< Host: dst-m
<< Connection: close
<< Cache-Control: no-cache
<<

```

Ce qui suit est un exemple de requête GET en mode session du protocole HTTP, avec une requête de modèle.

```

<< GET /jpip.cgi?fsiz=1024,768&cid=JPH_5&model=Hm,H*,M*,P* HTTP/1.1
<< Host: jpip.jpeg.org
<< Cache-Control: no-cache

>> HTTP/1.1 200 OK
>> Cache-control: no-cache
>> Transfer-Encoding: chunked
>> 3
-- 3 octets de données binaires
>> 0

```

Ce qui suit est un exemple d'une requête GET sans description d'état dans le protocole HTTP, avec une requête de modèle.

```

<< GET /images/kids.jp2?fsiz=1024,768&model=M0,Hm,H0:20,P0 HTTP/1.1
<< Host: jpip.jpeg.org
<< Cache-Control: no-cache

>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>> 400

```

ISO/CEI 15444-9:2005 (F)

```
-- 1024 octets de données binaires
>> 3f8
-- 1016 octets de données binaires
>> 0
```

K.5.3 Utilisation du protocole HTTP avec retour en protocole TCP

```
<< GET /phoenix.jp2?stream=0&len=2000&tid=0&type=jpp-stream&cnew=http-
tcp,http HTTP/1.1
<< Host: dst-m
<<
>> HTTP/1.1 200 OK
>> JPIP-tid: 281B6E135135BBC0BC588452AC9B73C5
>> JPIP-cnew: cid=JPHT033C38BE481154F9,path=jpip,transport=http-
tcp,auxport=80
>> Cache-Control: no-cache
>>
<< JPHT033C38BE481154F9 - [NOTE - Il s'agit du message de connexion
par canal TCP]
<<

<< GET /jpip.cgi?fsiz=834,834&roff=0,0&rsiz=834,790&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> HTTP/1.1 200 OK, avec modifications
>> JPIP-roff: 120,114
>> JPIP-rsiz: 593,561
>> Cache-Control: no-cache
>>

<< GET /jpip.cgi?fsiz=834,834&roff=229,254&rsiz=155,113&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>>

<< GET /jpip.cgi?fsiz=1667,1667&roff=457,507&rsiz=310,226&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>>

<< GET /jpip.cgi?fsiz=3334,3334&roff=914,1014&rsiz=620,452&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>>

<< GET /jpip.cgi?cclose=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
```

K.6 Utilisation du protocole JPIP avec le langage HTML

Un système JPIP peut être utilisé de diverses façons avec le langage HTML et avec des pages XHTML. Si un serveur distant JPIP possède la capacité de transcoder des portions d'une image en format JPEG ou en d'autres types de média d'image complète, alors le langage HTML peut servir à accéder à des portions d'image JPEG 2000 sans aucune modification aux navigateurs actuels.

Soit une page HTML contenant le fragment HTML suivant:

```

```

Tout navigateur HTML qui souhaite afficher cette page HTML avec une image va envoyer une requête afin d'obtenir cette image. Cette requête va commencer ainsi:

```
GET /name.jp2?fsiz=128,128&rsiz=128,128&type=image/jpeg
Host: jpip.jpeg.org
```

et va inclure de nombreuses autres lignes d'en-tête HTTP, désignant normalement le navigateur et les types de fichiers que le navigateur accepte. Cette requête HTTP est une requête JPIP légale et un serveur distant JPIP qui reçoit cette requête doit soit renvoyer un message d'erreur soit déterminer la portion correspondante du fichier JP2 à consulter puis la convertir en fichier JPEG. Le message renvoyé pourrait avoir l'allure suivante:

```
HTTP/1.1 200 OK
Content-type: image/jpeg
Content-length: 20387
CRLF
JPEG-Compressed-Image-Data
```

Ce qui précède est une réponse JPIP légale et est également une réponse HTTP légale que tous les navigateurs d'images savent comment afficher. Noter qu'il est préféré mais non requis que le serveur distant utilise le codage de transfert fragmenté de façon que cette requête puisse être interrompue. L'exemple précédent n'est pas un exemple de codage de transfert fragmenté.

Il est également possible d'écrire des pages HTML qui utiliseront le format JPEG quand seul ce format sera disponible; qui utiliseront le format JPEG 2000 quand il sera disponible; et qui utiliseront le format JPT ou JPP quand il sera disponible dans le navigateur du client. Soit le fragment HTML suivant:

```

```

Dans ce cas, aucun type explicite n'est demandé. Un serveur distant JPIP utilisant le protocole HTTP devrait examiner la ligne "Accept:" de la requête HTTP émise par le client. Selon la présence d'un fichier d'image/jp2 ou image/jpt-stream ou image/jpp-stream ou image/jpeg, le serveur distant pourra déterminer un format compatible à renvoyer.

Annexe L

Production du protocole JPIP en formalisme ABNF

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

L.1 Formalisme ABNF d'une requête JPIP

```

;=====
; C.1.1 Structure de requête
;=====

jpip-request-field = target-field
                    / channel-field
                    / view-window-field
                    / metadata-field
                    / data-limit-field
                    / server-control-field
                    / cache-management-field
                    / upload-field
                    / client-cap-pref-field

target-field       = target           ; C.2.2
                    / subtarget       ; C.2.3
                    / tid              ; C.2.4

channel-field      = cid              ; C.3.2
                    / cnew             ; C.3.3
                    / cclose          ; C.3.4
                    / qid              ; C.3.5

view-window-field = fsiz              ; C.4.2
                    / roff            ; C.4.3
                    / rsiz            ; C.4.4
                    / comps           ; C.4.5
                    / stream          ; C.4.6
                    / context         ; C.4.7
                    / srate           ; C.4.8
                    / roi              ; C.4.9
                    / layers          ; C.4.10

metadata-field     = metareq          ; C.5.2

data-limit-field   = len              ; C.6.1
                    / quality         ; C.6.2

server-control-field = align          ; C.7.1
                    / wait            ; C.7.2
                    / type            ; C.7.3
                    / drate           ; C.7.4

cache-management-field = model        ; C.8.1
                    / tpmodel         ; C.8.3
                    / need            ; C.8.4
                    / tpneed          ; C.8.5
                    / mset            ; C.8.6

upload-field       = upload           ; C.9.1
client-cap-pref-field = cap           ; C.10.1
                    / pref            ; C.10.2
                    / csf             ; C.10.3;

;=====
; C.2.2 Cible(target)
;=====

target = "target" "=" PATH

;=====

```

```

; C.2.3 Sous-cible (subtarget)
;=====
subtarget = "subtarget" "=" byte-range
byte-range = UINT-RANGE
;=====
; C.2.4 Identificateur de cible (tid)
;=====
tid = "tid" "=" target-id
target-id = TOKEN
;=====
; C.3.1 Identificateur de canal (cid)
;=====
cid = "cid" "=" channel-id
channel-id = TOKEN
;=====
; C.3.2 Nouveau canal (cnew)
;=====
cnew = "cnew" "=" 1#transport-name
transport-name = TOKEN
;=====
; C.3.3 Fermeture de canal (cclose)
;=====
cclose = "cclose" "=" ("*" / 1#channel-id)
;=====
; C.3.4 Identificateur de requête (qid)
;=====
qid = "qid" "=" UINT
;=====
; C.4.2 Longueur de trame (fsiz)
;=====
fsiz = "fsiz" "=" fx "," fy ["," round-direction]
fx = UINT
fy = UINT
round-direction = "round-up" / "round-down" / "closest"
;=====
; C.4.3 Décalage (roff)
;=====
roff = "roff" "=" ox "," oy
ox = UINT
oy = UINT
;=====
; C.4.4 Taille de la région (rsiz)
;=====
rsiz = "rsiz" "=" sx "," sy
sx = UINT
sy = UINT

```

```

;=====
; C.4.5 Composants (comps)
;=====

comps = "comps" "=" 1#UINT-RANGE

;=====
; C.4.6 Flux codé (stream)
;=====

stream = "stream" "=" 1#sampled-range
sampled-range = UINT-RANGE [":" sampling-factor]
sampling-factor = UINT

;=====
; C.4.7 Contexte du flux codé (context)
;=====

context = "context" "=" 1#context-range
context-range = jpxl-context-range / mj2t-context / reserved-context
jpxl-context-range = "jpxl" "<" jpx-layers ">" [ "[" jpxl-geometry "]" ]
jpx-layers = sampled-range
jpxl-geometry = "s" jpx-iset "i" jpx-inum
jpx-iset = UINT
jpx-inum = UINT
mj2t-context = "mj2t" "<" mj2-track ">" [ "[" mj2t-geometry "]" ]
mj2-track = NONZERO ["+" "now" ]
mj2t-geometry = "track" / "movie"
reserved-context = 1*( TOKEN / "<" / ">" / "[" / "]" / "-" / ":" / "+" )

;=====
; C.4.8 Fréquence d'échantillonnage (srate)
;=====

srate = "srate" "=" streams-per-second
streams-per-second = UFLOAT

;=====
; C.4.9 ROI (roi)
;=====

roi = "roi" "=" region-name
region-name = 1*(DIGIT / ALPHA / "_")
              / "dynamic"

;=====
; C.4.10 Couches (layers)
;=====

layers = "layers" "=" UINT

;=====
; C.5.2 Requête de métadonnées (metareq)
;=====

metareq = "metareq" "=" 1#( "[" 1$(req-box-prop) "]" [root-bin] [max-depth])
          [metadata-only]

req-box-prop = box-type [limit] [metareq-qualifier] [priority]
limit = ":" (UINT / "r")

```

```

metareq-qualifier = "/" 1*("w" / "s" / "g" / "a")
priority = "!"
root-bin = "R" UINT
max-depth = "D" UINT
metadata-only = "!!"

;=====
; C.6.1 Longueur maximale de réponse (len)
;=====
len = "len" "=" UINT

;=====
; C.6.2 Qualité (quality)
;=====
quality = "quality" "=" (1*2DIGIT / "100") ; 0 to 100

;=====
; C.7.1 Alignement (align)
;=====
align = "align" "=" ("yes" / "no")

;=====
; C.7.2 Attente (wait)
;=====
wait = "wait" "=" ("yes" / "no")

;=====
; C.7.3 Type de retour d'image (type)
;=====
type = "type" "=" 1#image-return-type
image-return-type = media-type / reserved-image-return-type
media-type = TOKEN "/" TOKEN *( ";" parameter )
reserved-image-return-type = TOKEN *( ";" parameter )
parameter = attribute "=" value
attribute = TOKEN
value = TOKEN

;=====
; C.7.4 Vitesse d'acheminement (drate)
;=====
drate = "drate" "=" rate-factor
rate-factor = UFLOAT

;=====
; C.8.1.1 Modèle (model)
;=====
model = "model" "=" 1#model-item
model-item = [codestream-qualifier ","] model-element
model-element = ["-"] bin-descriptor
bin-descriptor = explicit-bin-descriptor ; C.8.1.2
                 / implicit-bin-descriptor ; C.8.1.3
codestream-qualifier = "[" 1$(codestream-range) "]"

```

```

codestream-range = first-codestream-id ["-" [last-codestream-id]]
first-codestream-id = UINT
last-codestream-id = UINT
;=====
; C.8.1.2 Forme explicite
;=====
explicit-bin-descriptor = explicit-bin
                           [":" (number-of-bytes / number-of-layers )]
explicit-bin = codestream-main-header-bin
              / meta-bin
              / tile-bin
              / tile-header-bin
              / precinct-bin
number-of-bytes = UINT
number-of-layers = %x4c UINT ; "L"
codestream-main-header-bin = %x48 %x6d ; "Hm"
meta-bin = %x4d bin-uid ; "M"
tile-bin = %x54 bin-uid ; "T"
tile-header-bin = %x48 bin-uid ; "H"
precinct-bin = %x50 bin-uid ; "P"
bin-uid = UINT / "*"
;=====
; C.8.1.3 Forme implicite
;=====
implicit-bin-descriptor = 1*implicit-bin [":" number-of-layers]
implicit-bin = implicit-bin-prefix (data-uid / index-range-spec)
implicit-bin-prefix = %x74 ; t -- tile
                    / %x63 ; c -- component
                    / %x72 ; r -- resolution level
                    / %x70 ; p -- position
index-range-spec = first-index-pos "-" last-index-pos
first-index-pos = UINT
last-index-pos = UINT
data-uid = UINT / "*"
;=====
; C.8.3 Modèle d'élément de pavé utilisant un flux JPT (tpmodel)
;=====
tpmodel = "tpmodel" "=" 1#tpmodel-item
tpmodel-item = [codestream-qualifier "," ] tpmodel-element
tpmodel-element = ["-"] tp-descriptor
tp-descriptor = tp-range / tp-number
tp-range = tp-number "-" tp-number
tp-number = tile-number "." part-number
tile-number = UINT
part-number = UINT

```

```

;=====
; C.8.4 Besoin de requêtes sans description d'état (need)
;=====

need = "need" "=" 1#need-item

need-item = [codestream-qualifier "," ] bin-descriptor

;=====
; C.8.5 Besoin d'élément de pavé dans le cas des requêtes sans description
d'état (tpneed)
;=====

tpneed = "tpneed" "=" 1#tpneed-item

tpneed-item = [codestream-qualifier "," ] tp-descriptor

;=====
; C.8.6 Ensemble de modèles dans le cas des requêtes émises au cours d'une
session (mset)
;=====

mset = "mset" "=" 1#sampled-range

;=====
; C.9.1 Téléchargement vers l'amont (upload)
;=====

upload = "upload" "=" upload-type

upload-type = image-return-type ; C.7.3

;=====
; C.10.1 Capacité du client (cap)
;=====

cap = "cap" "=" 1#capability-group

capability-group = processing-capability
                  / depth-capability
                  / config-capability

processing-capability = compatibility-capability
                      / vendor-capability

compatibility-capability = "cc." compatibility-code

vendor-capability = "vc." vendor-code [":" vendor-value]

vendor-code = 1*(LOWER / DIGIT / "." / "-")

vendor-value = TOKEN

depth-capability = "depth:" UINT

config-capability = "config:" UINT

;=====
; C.10.2.1 Généralités
;=====

pref = "pref" "=" 1#(related-pref-set ["/r"])

related-pref-set = view-window-pref ; C.10.2.2
                  / colour-meth-pref ; C.10.2.3
                  / max-bandwidth ; C.10.2.4
                  / bandwidth-slice ; C.10.2.5
                  / placeholder-pref ; C.10.2.6
                  / codestream-seq-pref ; C.10.2.7
                  / other

other = TOKEN

;=====

```

```

; C.10.2.2 Préférences en matière de manipulation de la fenêtre de
visualisation
;=====
view-window-pref = "fullwindow" / "progressive"

;=====
; C.10.2.3 Préférences en matière de méthode de détermination de l'espace
colorimétrique
;=====
color-meth-pref = 1$(color-meth [":" meth-limit])
color-meth = "color-enum" / "color-ricc" / "color-icc" / "color-vend"
meth-limit = UINT

;=====
; C.10.2.4 Largeur de bande maximale
;=====
max-bandwidth = "mbw:" mbw
mbw = UINT ["K" / "M" / "G" / "T"]

;=====
; C.10.2.5 Tranche spectrale
;=====
bandwidth-slice = "slice:" slice
slice = NONZERO

;=====
; C.10.2.6 Préférence en terme de générique
;=====
placeholder-pref = "meta:" placeholder-branch
placeholder-branch = "incr" / "equiv" / "orig"

;=====
; C.10.2.7 Séquencement des flux codés
;=====
codestream-seq-pref = "codeseq:" codestream-seq-option
codestream-seq-option = "sequential" / "reverse-sequential"
                        / "interleaved"

;=====
; C.10.3 Sensibilité différentielle (csf)
;=====
csf = "csf" "=" 1#csf-sample-line
csf-sample-line = csf-density [";" csf-angle] ";" 1$sensitivity
csf-density = "density" ":" UFLOAT
csf-angle = "angle" ":" UFLOAT
sensitivity = UFLOAT

```

L.2 Formalisme BNF d'une réponse JPIP

```

;=====
; D.1.1 Structure de réponse
;=====
Status-Code = 3DIGIT
Reason-Phrase = *<TEXT, excluding CR and LF>

```

```

jpip-response-header =
    / JPIP-tid ; D.2.2
    / JPIP-cnew ; D.2.3
    / JPIP-qid ; D.2.4
    / JPIP-fsiz ; D.2.5
    / JPIP-rsiz ; D.2.6
    / JPIP-roff ; D.2.7
    / JPIP-comps ; D.2.8
    / JPIP-stream ; D.2.9
    / JPIP-context ; D.2.10
    / JPIP-roi ; D.2.11
    / JPIP-layers ; D.2.12
    / JPIP-srate ; D.2.13
    / JPIP-metareq ; D.2.14
    / JPIP-len ; D.2.15
    / JPIP-quality ; D.2.16
    / JPIP-type ; D.2.17
    / JPIP-mset ; D.2.18
    / JPIP-cap ; D.2.19
    / JPIP-pref ; D.2.20

;=====
; D.2.2 Target ID (JPIP-tid)
;=====
JPIP-tid = "JPIP-tid" ":" LWSP target-id

;=====
; D.2.3 Nouveau canal (JPIP-cnew)
;=====
JPIP-cnew = "JPIP-cnew" ":" LWSP "cid" "=" channel-id
            [", " 1#(transport-param "=" TOKEN)]
transport-param = TOKEN

;=====
; D.2.4 Identificateur de requête (JPIP-qid)
;=====
JPIP-qid = "JPIP-qid" ":" LWSP UINT

;=====
; D.2.5 Longueur de trame (JPIP-fsiz)
;=====
JPIP-fsiz = "JPIP-fsiz" ":" LWSP fx ", " fy

;=====
; D.2.6 Taille de la région (JPIP-rsiz)
;=====
JPIP-rsiz = "JPIP-rsiz" ":" LWSP sx ", " sy

;=====
; D.2.7 Décalage (JPIP-roff)
;=====
JPIP-roff = "JPIP-roff" ":" LWSP ox ", " oy

;=====
; D.2.8 Composants (JPIP-comps)
;=====
JPIP-comps = "JPIP-comps" ":" LWSP 1#UINT-RANGE

;=====
; D.2.9 Flux codé (JPIP-stream)
;=====
JPIP-stream = "JPIP-stream" ":" LWSP 1#(prefixed-range / sampled-range)

```

```

prefixed-range = "<" ctxt-id ":" ctxt-elt ">" sampled-range
ctxt-id = UINT
ctxt-elt = UINT

;=====
; D.2.10 Contexte du flux codé (JPIP-context)
;=====

JPIP-context = "JPIP-context" ":" LWSP 1$(context-range "=" 1#sampled-range)

;=====
; D.2.11 ROI (JPIP-roi)
;=====

JPIP-roi = "JPIP-roi" ":" LWSP
         "roi" "=" region-name ";"
         "fsiz" "=" UINT "," UINT ";"
         "rsiz" "=" UINT "," UINT ";"
         "roff" "=" UINT "," UINT ";"

region-name = 1*(DIGIT / ALPHA / "_")

;=====
; D.2.12 Couches (JPIP-layers)
;=====

JPIP-layers = "JPIP-layers" ":" LWSP UINT

;=====
; D.2.13 Fréquence d'échantillonnage (JPIP-srate)
;=====

JPIP-srate = "JPIP-srate" ":" LWSP UFLOAT

;=====
; D.2.14 Requête de métadonnées (JPIP-metareq)
;=====

JPIP-metareq = "JPIP-metareq" ":" LWSP
              1#( "[" 1$(req-box-prop) "]" [root-bin] [max-depth] )
              [metadata-only]

req-box-prop = box-type [limit] [metareq-qualifier] [priority]

;=====
; D.2.15 Longueur maximale de réponse (JPIP-length)
;=====

JPIP-len = "JPIP-len" ":" LWSP UINT

;=====
; D.2.16 Qualité (JPIP-quality)
;=====

JPIP-quality = "JPIP-quality" ":" LWSP (1*2DIGIT / "100" / "-1")

;=====
; D.2.17 Type de retour d'image (JPIP-type)
;=====

JPIP-type = "JPIP-type" ":" LWSP image-return-type

;=====
; D.2.18 Ensemble de modèles (JPIP-mset)
;=====

JPIP-mset = "JPIP-mset" ":" LWSP 1#sampled-range

;=====
; D.2.19 Capacité recherchée (JPIP-cap)
;=====

```

```
JPIP-cap = "JPIP-cap" ":" LWSP 1#capability-code  
;=====  
; D.2.20 Préférence indisponible (JPIP-pref)  
;=====  
JPIP-pref = "JPIP-pref" ":" LWSP 1#related-pref-set
```

Annexe M

Revendication de droits de propriété intellectuelle

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

L'Organisation internationale de normalisation (ISO, *International Organization for Standardization*) et la Commission électrotechnique internationale (CEI, *International Electrotechnical Commission*) appellent l'attention sur le fait qu'il est revendiqué que la conformité à cette partie de l'ISO/CEI 15444 peut impliquer l'utilisation de brevets.

L'ISO et la CEI ne prennent aucune position concernant l'évidence, la validité et le domaine d'application de ces droits de brevet.

Les détenteurs de ces droits de brevet ont assuré l'ISO et la CEI qu'ils étaient disposés à négocier des licences à des termes ou conditions raisonnables et non discriminatoires en faveur de requérants situés dans le monde entier. A cet égard, les déclarations des détenteurs de ces droits de brevet sont enregistrées par l'ISO et par la CEI. Des informations peuvent être obtenues auprès des sociétés énumérées ci-dessous.

L'attention est appelée sur la possibilité que certains éléments de cette partie de l'ISO/CEI 15444 puissent faire l'objet de droits de brevet autres que ceux qui sont identifiés dans la présente annexe. L'ISO et la CEI ne doivent pas être tenues responsables de l'identification de tout ou partie de tels droits de brevet.

	Société
1	Canon Inc.
2	Ricoh Company, Limited

Annexe N

Bibliographie

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

- [1] TAUBMAN (D.): Remote Browsing of JPEG 2000 Images (Exploration à distance d'images JPEG 2000), *Proc. Int. Conf. on Image Processing*, Vol. 1, pp. 229-232, septembre 2002.
- [2] LI (J.), SUN (H.), LI (H.), ZHANG (Q.), LIN (X.): Vfile – A Virtual File Media Access Mechanism and its Application in JPEG2000 Images for Browsing over Internet (Vfile: un mécanisme d'accès à des médias de fichiers virtuels et son application aux images JPEG 2000 pour la navigation par Internet), *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N1473*, novembre 1999.
- [3] BOLIEK (M.), WU (G.K.), GORMISH (M.J.): JPEG 2000 for Efficient Imaging in a Client/Server Environment (Format JPEG 2000 pour une imagerie efficace dans un environnement client-serveur), *Proc. SPIE Conf. on Applications of Digital Image Processing*, Vol. 4472, pp. 212-223, décembre 2001.
- [4] DESHPANDE (S.), ZENG (W.): Scalable Streaming of JPEG2000 Images Using Hypertext Transfer Protocol (Commande modulable du débit d'images JPEG 2000 au moyen du protocole HTTP), *Proc. ACM Conf. on Multimedia*, pp. 372-381, octobre 2001.
- [5] WRIGHT (A.), CLARK (R.), COLYER (G.): An Implementation of JPIP Based on HTTP (Mise en œuvre du protocole JPIP au-dessus du protocole HTTP), *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N2426*, février 2002.
- [6] GORMISH (M.), BANERJEE (S.): Tile-Based Transport of JPEG 2000 (Transport en mode pavé d'image JPEG 2000), N. Garcia, J.M. Martinez, L. Salgado (Eds.), VLVB03, LNCS 2849, pp. 217-224, 2003.
- [7] TAUBMAN (D.), ROSENBAUM (R.): Rate-Distortion Optimized Interactive Browsing of JPEG2000 Images (Exploration interactive d'image JPEG 2000, optimisée en terme de distorsion de débit), *Proc. Int. Conf. on Image Processing*, septembre 2003.
- [8] TAUBMAN (D.), PRANDOLINI (R.): Architecture, Philosophy and Performance of JPIP: Internet Protocol Standard for JPEG2000 (Architecture, fondamentaux et performances du protocole JPIP: norme de protocole Internet pour JPEG 2000), presented at *Visual Communications and Image Processing*, Lugano, Suisse, 2003.
- [9] GORMISH (M.J.): TRUEW: Transport of Reversible and Unreversible EmbeddedWavelets (A JPIP Proposal), (Transport réversible et irréversible d'ondelettes imbriquées (proposition JPIP)) *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N2602*, juillet 2002.
- [10] CANON: Proposal for JPIP Tier 2 protocol (Proposition pour l'étape 2 du protocole JPIP), *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N2608*, juin 2002.
- [11] TAUBMAN (D.), MARCELLIN (M.): JPEG2000: image compression fundamentals, standards and practice (JPEG 2000: Bases fondamentales, normes et procédés de la compression d'image), *Kluwer Academic Publishers*, Boston, 2001.

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