

Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

G.722.1

(05/2005)

SÉRIE G: SYSTÈMES ET SUPPORTS DE
TRANSMISSION, SYSTÈMES ET RÉSEAUX
NUMÉRIQUES

Equipements terminaux numériques – Codage des
signaux analogiques par des méthodes autres que la MIC

**Codage à faible complexité aux débits de 24 et
32 kbit/s pour utilisation en mains-libres sur les
systèmes à faible perte de trames**

Recommandation UIT-T G.722.1



RECOMMANDATIONS UIT-T DE LA SÉRIE G
SYSTÈMES ET SUPPORTS DE TRANSMISSION, SYSTÈMES ET RÉSEAUX NUMÉRIQUES

CONNEXIONS ET CIRCUITS TÉLÉPHONIQUES INTERNATIONAUX	G.100–G.199
CARACTÉRISTIQUES GÉNÉRALES COMMUNES À TOUS LES SYSTÈMES ANALOGIQUES À COURANTS PORTEURS	G.200–G.299
CARACTÉRISTIQUES INDIVIDUELLES DES SYSTÈMES TÉLÉPHONIQUES INTERNATIONAUX À COURANTS PORTEURS SUR LIGNES MÉTALLIQUES	G.300–G.399
CARACTÉRISTIQUES GÉNÉRALES DES SYSTÈMES TÉLÉPHONIQUES INTERNATIONAUX HERTZIENS OU À SATELLITES ET INTERCONNEXION AVEC LES SYSTÈMES SUR LIGNES MÉTALLIQUES	G.400–G.449
COORDINATION DE LA RADIOTÉLÉPHONIE ET DE LA TÉLÉPHONIE SUR LIGNES	G.450–G.499
CARACTÉRISTIQUES DES SUPPORTS DE TRANSMISSION	G.600–G.699
EQUIPEMENTS TERMINAUX NUMÉRIQUES	G.700–G.799
Généralités	G.700–G.709
Codage des signaux analogiques en modulation par impulsions et codage	G.710–G.719
Codage des signaux analogiques par des méthodes autres que la MIC	G.720–G.729
Principales caractéristiques des équipements de multiplexage primaires	G.730–G.739
Principales caractéristiques des équipements de multiplexage de deuxième ordre	G.740–G.749
Caractéristiques principales des équipements de multiplexage d'ordre plus élevé	G.750–G.759
Caractéristiques principales des équipements de transcodage et de multiplication numérique	G.760–G.769
Fonctionnalités de gestion, d'exploitation et de maintenance des équipements de transmission	G.770–G.779
Caractéristiques principales des équipements de multiplexage en hiérarchie numérique synchrone	G.780–G.789
Autres équipements terminaux	G.790–G.799
RÉSEAUX NUMÉRIQUES	G.800–G.899
SECTION NUMÉRIQUES ET SYSTÈMES DE LIGNES NUMÉRIQUES	G.900–G.999
QUALITÉ DE SERVICE ET DE TRANSMISSION – ASPECTS GÉNÉRIQUES ET ASPECTS LIÉS À L'UTILISATEUR	G.1000–G.1999
CARACTÉRISTIQUES DES SUPPORTS DE TRANSMISSION	G.6000–G.6999
DONNÉES SUR COUCHE TRANSPORT – ASPECTS GÉNÉRIQUES	G.7000–G.7999
ASPECTS RELATIFS AU PROTOCOLE ETHERNET SUR COUCHE TRANSPORT	G.8000–G.8999
RÉSEAUX D'ACCÈS	G.9000–G.9999

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

Recommandation UIT-T G.722.1

Codage à faible complexité aux débits de 24 et 32 kbit/s pour utilisation en mains-libres sur les systèmes à faible perte de trames

Résumé

La présente Recommandation décrit un codeur et un décodeur peu complexes, pouvant traiter un signal audiofréquence d'une largeur de bande de 7 kHz et fonctionner à 24 kbit/s ou 32 kbit/s. Par ailleurs, cet algorithme est recommandé pour les applications mains-libres, par exemple de conférence, qui ne comportent qu'une faible probabilité de perte de trames. Il peut s'utiliser avec des signaux d'entrée de parole ou de musique et permet de changer le débit à l'une quelconque des limites de trame de 20 ms. La nouvelle Annexe C contient la description d'un mode d'extension à faible complexité de la Rec. UIT-T G.722.1, qui permet de doubler l'algorithme afin d'obtenir une largeur de bande audio de 14 kHz utilisant une fréquence de 32 kHz pour les échantillons audio, aux débits de 24, 32 et 48 kbit/s. Adapté à la visioconférence, à la téléconférence ainsi qu'aux applications de lecture en temps réel, ce mode utilise la même longueur de trame de 20 ms, le même temps de traitement associé à l'algorithme de 40 ms et les mêmes étapes algorithmiques que le mode à 7 kHz. Le codage et le décodage nécessitent moins de 5,5 millions d'opérations pondérées par seconde (WMOPS) dans le mode à 7 kHz et moins de 11 WMOPS dans le mode à 14 kHz de l'Annexe C.

La présente Recommandation comporte un logiciel contenant le code source du codeur et du décodeur ainsi qu'un ensemble de vecteurs test pour le développement d'application. Ces vecteurs sont un outil qui fournit une indication sur la validité des implémentations. Le code à virgule fixe implémente à la fois le mode à 7 kHz (corps du texte) et le mode à 14 kHz (Annexe C). Le code à virgule flottante implémente uniquement le mode à 7 kHz.

La présente édition inclus le Corrigendum 1 (06/2008) qui corrige le code source du fichier decoder.c dans l'annexe B. Aucun changement n'a été apporté au texte de la recommandation. Le Code source de l'édition intégrée est numéroté Version 2.1 par souci d'homogénéité.

Source

La Recommandation UIT-T G.722.1 a été approuvée le 14 mai 2005 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Le Corrigendum 1 a été approuvé le 13 juin 2008 par la Commission d'études 16 de l'UIT-T (2005-2008) selon la procédure définie dans la Rec. UIT-T A.8.

AVANT-PROPOS

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

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

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

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

NOTE

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

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

		Page
1	Domaine d'application	1
2	Références normatives.....	2
3	Codeur	2
	3.1 Transformation MLT (à modulation et chevauchement)	4
	3.2 Calcul et quantification de l'enveloppe d'amplitude.....	5
	3.3 Codage de l'enveloppe d'amplitude	5
	3.4 Procédure de catégorisation.....	6
	3.5 Codage de Huffman à combinaison en vecteurs et quantification scalaire (SQVH, <i>scalar quantized vector Huffman coding</i>)	9
	3.6 Commande de débit.....	11
	3.7 Transmission des indices vectoriels MLT	11
	3.8 Flux binaire.....	11
4	Décodeur.....	12
	4.1 Décodage de l'enveloppe d'amplitude	12
	4.2 Détermination de la catégorisation.....	12
	4.3 Décodage des coefficients MLT.....	13
	4.4 Remplissage de bruit	13
	4.5 Bits insuffisants	13
	4.6 Effacement de trame.....	14
	4.7 Transformation MLT inverse (IMLT, <i>inverse MLT</i>)	14
5	Code C.....	14
6	Organigramme de la procédure de catégorisation	16
Annexe A – Format des paquets, identificateurs de capacité et paramètres de capacités.....		20
	A.1 Références	20
	A.2 Structure des paquets pour les trames G.722.1.....	20
	A.3 Identificateurs et paramètres de capacité utilisés avec la Rec. UIT-T H.245.....	22
Annexe B – Implémentation en virgule flottante de la Rec. UIT-T G.722.1.....		25
	B.1 Introduction	25
	B.2 Description algorithmique	25
	B.3 Code en langage C de l'ANSI.....	25
Annexe C – Mode à 14 kHz aux débits de 24, 32 et 48 kbit/s.....		26
	C.1 Introduction	26
	C.2 Description algorithmique	26
	C.3 Code en langage C de l'ANSI.....	28

Recommandation UIT-T G.722.1

Codage à faible complexité aux débits de 24 et 32 kbit/s pour utilisation en mains-libres sur les systèmes à faible perte de trames¹

1 Domaine d'application

La présente Recommandation décrit un algorithme de codeur numérique à large bande, produisant une largeur de bande audiofréquence de 50 Hz à 7 kHz et un débit de 24 kbit/s ou 32 kbit/s. L'entrée numérique du codeur peut être représentée en complément à 2 de 14, 15 ou 16 bits, à la fréquence d'échantillonnage de 16 kHz (le traitement est identique à celui décrit dans la Rec. UIT-T G.722). Les circuits d'interface analogiques et numériques à l'entrée du codeur et à la sortie du décodeur devraient être conformes aux spécifications de la Rec. UIT-T G.722.

L'algorithme se fonde sur la technologie des transformations à modulation et chevauchement (MLT, *modulated lapped transform*) et s'applique à des trames de 20 ms (320 échantillons) du signal audiofréquence. Comme la fenêtre de transformation (longueur de la fonction de base) correspond à 640 échantillons et qu'un chevauchement de 50% (320 échantillons) est prévu entre les trames, la capacité effective du tampon permet une durée d'anticipation de 20 ms. Ainsi, le retard algorithmique total de 40 ms est égal à la somme de la durée de trame et de la durée d'anticipation. Tous les autres retards sont liés à la durée des calculs et des transmissions sur le réseau.

La description de l'algorithme de codage de la présente Recommandation se fonde sur des opérations mathématiques à exactitude binaire en virgule fixe. Le code C indiqué au paragraphe 5, qui fait partie intégrante de la présente Recommandation, reflète cette approche descriptive à exactitude binaire en virgule fixe et, en cas d'irrégularité, il doit avoir la priorité sur les descriptions mathématiques des paragraphes 3 et 4.

Les descriptions mathématiques du codeur (paragraphe 3) et du décodeur (paragraphe 4) auraient pu être appliquées de plusieurs autres façons, mais le code C du paragraphe 5 a été choisi comme référence. Pour satisfaire à la présente Recommandation, toute implémentation doit donc produire les mêmes résultats que le code C du paragraphe 5, quel que soit le signal d'entrée.

Afin d'assurer l'atteinte de cet objectif, les implémentations doivent être conformes aux détails des calculs, aux tables de constantes, à l'adaptation du séquençement des variables et à l'utilisation indiquée par le code C du paragraphe 5. Il est toutefois reconnu que de nombreux éléments de l'algorithme sont essentiels à l'exécution correcte des opérations à exactitude binaire. Dans le cas de ces éléments, les applications doivent reproduire les détails des calculs, les tables de constantes, l'adaptation du séquençement des variables et l'utilisation indiquée par le code C du paragraphe 5.

Il est entendu que le code C est fourni à titre de référence et qu'il n'a pas été optimisé (du point de vue de la mémoire exigée, de la complexité, etc.) en fonction d'une plate-forme d'application spécifique. Il pourrait donc exiger une telle optimisation, selon les conditions particulières des implémentations.

La présente Recommandation spécifie un ensemble non exhaustif de signaux d'essai, qui pourrait aider les concepteurs à faire en sorte que leurs implémentations de codeur et de décodeur satisfassent à la présente Recommandation.

En pratique, les acheteurs de matériel à large bande ou de logiciel d'implémentation s'attendent à ce que ce matériel et ce logiciel soient conformes à la présente norme et qu'ils permettent donc

¹ La présente Recommandation inclut un logiciel comprenant le code source du codeur et du décodeur et un ensemble de vecteurs de test aux fins de développement.

l'interfonctionnement. Les concepteurs pourront décider d'optimiser ou de modifier autrement le code C de référence. Ils devront alors veiller à ce que leurs applications produisent la même sortie que le code C du paragraphe 5, quelles que soient les entrées.

2 Références normatives

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

- [1] Recommandation CCITT G.722 (1988), *Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s.*
- [2] Recommandation UIT-T G.192 (1996), *Interface parallèle numérique commune pour la normalisation des signaux vocaux.*
- [3] ISO/CEI 9899:1999, *Langages de programmation – C.*

3 Codeur

La Figure 1 présente un schéma de principe du codeur.

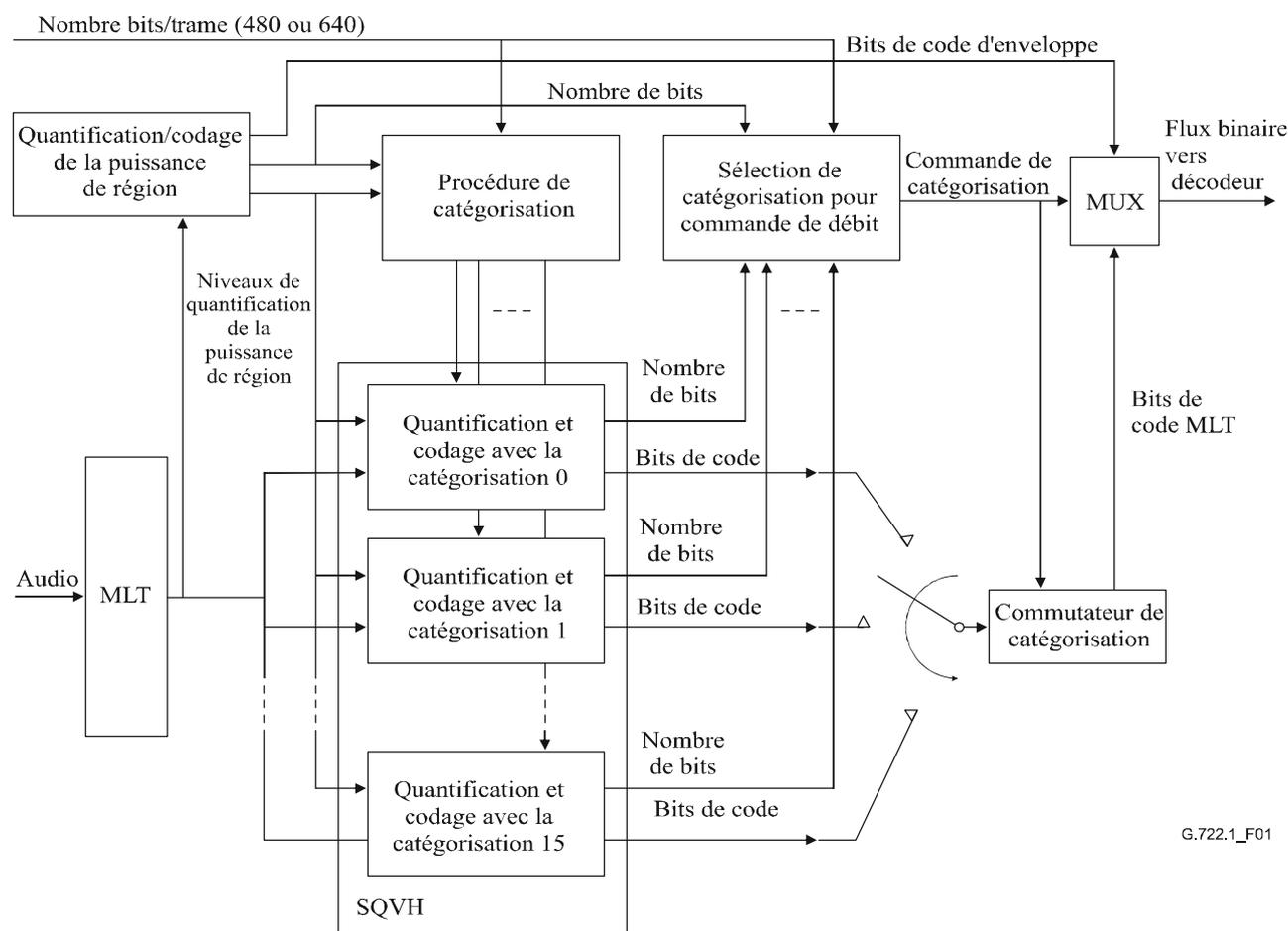
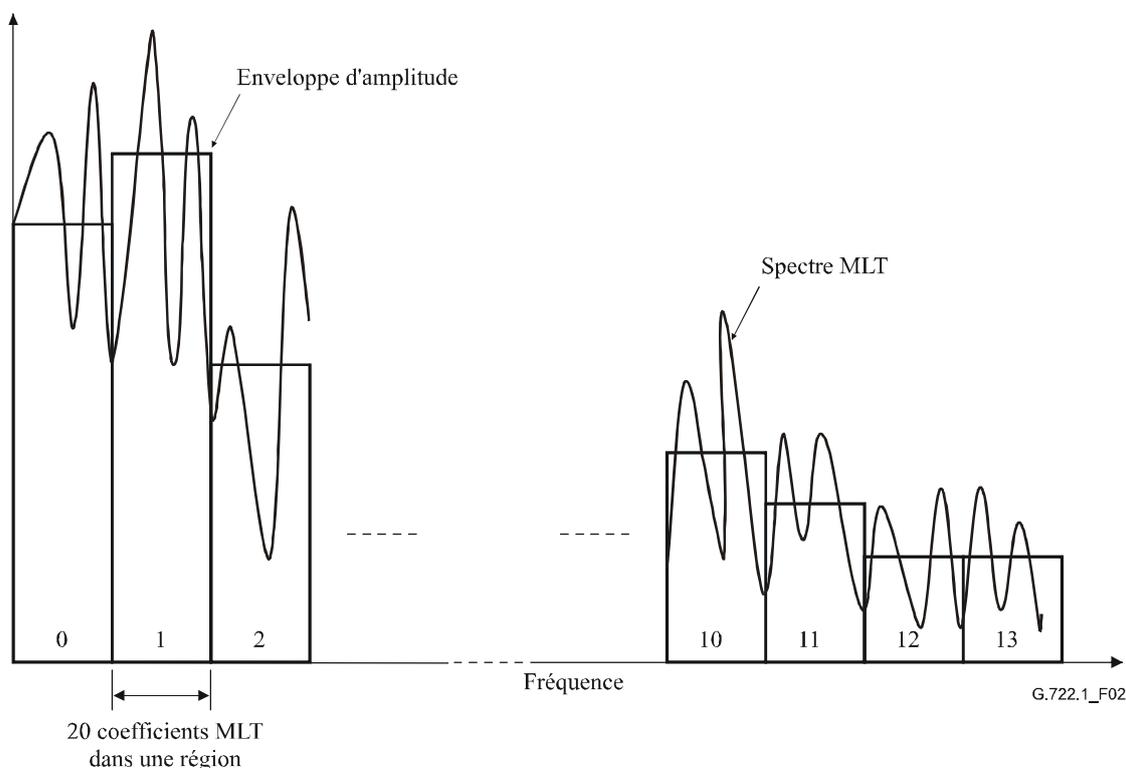


Figure 1/G.722.1 – Schéma de principe du codeur

Toutes les 20 millisecondes, ce qui correspond à 320 échantillons, les 640 derniers échantillons audiofréquence dans le domaine temporel sont dirigés vers un dispositif de transformation MLT (à modulation et chevauchement). Chaque transformation produit une trame de 320 coefficients MLT, et chaque trame de coefficients MLT est codée indépendamment, c'est-à-dire qu'il ne reste aucune – information d'état liée à la trame précédente. Les bits affectés à chaque trame sont respectivement au nombre de 480 et de 640, pour les débits de 24 kbit/s et de 32 kbit/s.

Les coefficients que produit la transformation MLT sont d'abord appliqués à un module qui calcule l'enveloppe d'amplitude et qui la quantifie (voir Figure 2). L'enveloppe d'amplitude constitue une représentation grossière du spectre MLT. Le spectre se divise en blocs de 20 coefficients MLT, appelés *régions*. Chaque région représente une largeur de bande de 500 Hz. Comme la largeur de bande est de 7 kHz, le nombre de régions (*number_of_regions*) est fixé à quatorze. Il n'est pas tenu compte des coefficients MLT représentant des fréquences supérieures à 7 kHz. Les bits de code représentant l'enveloppe d'amplitude sont dirigés vers le multiplexeur (MUX), puis transmis au décodeur. Les bits qui restent après la quantification et le codage de l'enveloppe d'amplitude servent à coder les coefficients MLT au cours de la procédure de *catégorisation*.



NOTE – Chaque point de l'enveloppe d'amplitude représente la valeur quadratique moyenne des coefficients MLT dans la région.

Figure 2/G.722.1 – Illustration de la répartition du spectre entre quatorze régions contenant chacune 20 coefficients MLT

A l'aide de l'enveloppe d'amplitude quantifiée et du nombre de bits qui restent dans la trame après le codage de l'enveloppe d'amplitude (et l'établissement de quatre bits de commande de catégorisation), la procédure de catégorisation produit un ensemble de seize *catégorisations* (*catégorisation 0* à *catégorisation 15*). Les différentes catégorisations nécessitent des nombres de bits différents pour le codage des mêmes coefficients MLT.

Chaque catégorisation comprend un ensemble constitué de quatorze *assignations de catégorie*, soit une assignation pour chacune des quatorze régions. Une *catégorie* définit un ensemble de paramètres de quantification et de codage pour une région. Afin de permettre le codage d'une

région, chaque catégorie est associée à un nombre de bits prévu. Comme le codeur repose sur le codage de Huffman à longueur variable, le nombre final des bits utilisés dépend de la séquence particulière des coefficients MLT dans la région.

Ensuite, les coefficients MLT sont quantifiés et codés différemment pour chacune des seize catégorisations calculées. Le nombre réel des bits de code requis est déterminé pour chaque catégorisation.

La quantification et le codage s'effectuent région par région. Une catégorisation détermine les assignations de catégorie pour l'ensemble des quatorze régions, et l'assignation de catégorie ainsi que l'enveloppe d'amplitude pour chaque région établissent tous les paramètres de quantification et de codage à utiliser pour l'ensemble des vingt coefficients MLT de la région.

Les coefficients MLT d'une région sont d'abord normalisés par l'enveloppe d'amplitude quantifiée de la région, puis soumis à une quantification scalaire. Les indices résultant de la quantification scalaire sont combinés en indices vectoriels. Ces indices vectoriels subissent ensuite un codage de Huffman, c'est-à-dire qu'ils sont codés avec un nombre variable de bits. Les indices vectoriels fréquents nécessitent moins de bits que les indices vectoriels moins fréquents.

Comme ce codec utilise le codage de Huffman à longueur variable et qu'un débit de transmission constant est nécessaire, il est essentiel d'imposer une contrainte au débit, selon le débit du canal. Quatre bits de *commande de catégorisation* indiquent au décodeur la catégorisation qui a été sélectionnée. Le commutateur de catégorisation dirige les bits de code (représentant les coefficients MLT quantifiés produits à partir de la catégorisation sélectionnée) vers le multiplexeur pour la transmission. La catégorisation dont le nombre de bits est le plus rapproché du débit du canal est sélectionnée pour la transmission.

3.1 Transformation MLT (à modulation et chevauchement)

La transformation MLT est une transformation linéaire à reconstitution parfaite et échantillonnage critique, présentant un chevauchement de 50% entre les fonctions de base des trames MLT adjacentes. Les entrées de chaque dispositif MLT sont constituées des 640 échantillons audiofréquence les plus récents, $x(n)$,

où:

$$x(0) \text{ est l'échantillon le plus ancien,}$$

et:

$$0 \leq n < 640$$

Le dispositif MLT produit 320 coefficients de transformation, $mlt(m)$,

où:

$$0 \leq m < 320$$

La transformation MLT est donnée par:

$$mlt(m) = \sum_{n=0}^{639} \sqrt{\frac{2}{320}} \sin\left(\frac{\pi}{640}(n+0,5)\right) \cos\left(\frac{\pi}{320}(n-159,5)(m+0,5)\right) x(n)$$

La transformation MLT peut se décomposer en opération de fenêtrage, de chevauchement et d'addition, suivie d'une transformée discrète en cosinus (DCT, *discrete cosine transform*) de type IV. L'opération de fenêtrage, de chevauchement et d'addition est donnée par:

$$v(n) = w(159-n)x(159-n) + w(160+n)x(160+n) \quad \text{pour } 0 \leq n \leq 159$$

$$v(n+160) = w(319-n)x(320+n) - w(n)x(639-n) \quad \text{pour } 0 \leq n \leq 159$$

où:

$$w(n) = \sin\left(\frac{\pi}{640}(n+0,5)\right) \text{ pour } 0 \leq n < 320$$

Si l'on combine $v(n)$ avec une transformée DCT de type IV, l'expression résultant de la transformation MLT est:

$$mlt(m) = \sum_{n=0}^{319} \sqrt{\frac{2}{320}} \cos\left(\frac{\pi}{320}(n+0,5)(m+0,5)\right) v(n)$$

A noter que les techniques de transformation rapide réduisent considérablement la complexité de la transformation DCT.

3.2 Calcul et quantification de l'enveloppe d'amplitude

Les coefficients MLT se divisent en régions de vingt coefficients. Ainsi, le nombre total de régions $number_of_regions = 14$. La région r comprend les coefficients MLT $20r$ à $20r + 19$,

où:

$$0 \leq r < number_of_regions$$

Les quarante coefficients MLT des fréquences les plus élevées, qui représentent des fréquences supérieures à 7 kHz, ne sont pas utilisés du fait qu'ils se situent à l'extérieur de la largeur de bande d'intérêt.

L'enveloppe d'amplitude dans la région r se définit comme l'écart quadratique moyen (RMS, *root-mean-square*) des coefficients MLT de la région, et elle se calcule comme suit:

$$rms(r) = \sqrt{\frac{1}{20} \sum_{n=0}^{19} mlt(20r+n) mlt(20r+n)}$$

Elle est ensuite quantifiée. L'indice de sortie du quantificateur est $rms_index(r)$. L'ensemble autorisé des valeurs de reconstitution de la quantification est le suivant:

$$2^{\left(\frac{i+2}{2}\right)} \text{ pour les valeurs entières de } i, -8 \leq i \leq 31$$

et $rms_index(0)$ est contraint à nouveau de façon que

$$1 \leq rms_index(0) \leq 31$$

Une mesure logarithmique est utilisée pour que les valeurs quantifiées à $2^{\left(\frac{i+2}{2}\right)}$ se situent de $2^{\left(\frac{i-0,5+2}{2}\right)}$ à $2^{\left(\frac{i+0,5+2}{2}\right)}$.

Par exemple, si $rms(r) = 310$, alors le niveau de quantification correspondant est de $2^{\left(\frac{15+2}{2}\right)}$, soit 362,04, et $rms_index(r) = 15$, car $2^{\left(\frac{15-0,5+2}{2}\right)} = 304,43$.

3.3 Codage de l'enveloppe d'amplitude

La première valeur transmise dans chaque trame est celle de $rms_index(0)$. Cinq bits sont utilisés. Le bit le plus significatif de l'indice est transmis en premier. La valeur, $rms_index(0) = 0$, est réservée et demeure inutilisée.

Les indices des treize autres régions sont soumis à un codage différentiel, puis à un codage de Huffman, avant d'être transmis. Les plus grandes différences qu'il est possible de coder sont +11 et -12. Afin de maintenir les différences à l'intérieur de cette plage, on élève d'abord les valeurs de creux afin que les valeurs de crête subséquentes puissent être représentées avec exactitude. C'est ce que décrit le pseudocode C suivant:

```

pour (r = number_of_regions - 2; r >= 0; r--)
{
    si (rms_index[r] < rms_index[r + 1] - 11)
        rms_index[r] = rms_index[r + 1] - 11;
}
pour (r = 1; r < number_of_regions; r++)
{
    j = rms_index[r] - rms_index[r - 1];
    si (j < -12)
    {
        j = -12
        rms_index[r] = rms_index[r - 1] + j;
    }
    differential_rms_index[r] = j;
}

```

Les différences, *differential_rms_index[r]*, sont transmises par ordre de région. Elles sont codées conformément aux codes de Huffman à longueur variable définis dans la table *differential_region_power_codes[r][j + 12]*, ainsi que dans la table *differential_region_power_bits[r][j + 12]* qui définit le nombre de bits de chaque code de Huffman. Ces tables se trouvent dans la partie de la présente Recommandation qui traite du code C. Chaque région est associée à un ensemble unique de codes de Huffman. Le bit le plus à gauche (ou le plus significatif) est toujours transmis en premier.

3.4 Procédure de catégorisation

La *procédure de catégorisation* détermine les échelons de quantification (et les autres paramètres connexes de quantification et de codage) utilisés pour quantifier les coefficients MLT.

Le processus de *catégorisation* assigne une catégorie à chacune des régions. Il existe huit catégories, numérotées de 0 à 7. Seize catégorisations différentes sont calculées, mais une seule est sélectionnée pour la transmission.

La même procédure de catégorisation est exécutée dans le décodeur. Il importe donc, pour les fins de l'interfonctionnement, que différentes implémentations de cette procédure produisent des catégorisations identiques lorsqu'elles reçoivent les mêmes entrées. Les entrées de cette procédure sont les suivantes:

- *number_of_available_bits*: nombre réel des bits de la trame qui demeurent inutilisés, une fois qu'il a été tenu compte de l'enveloppe d'amplitude et des bits de commande de catégorisation.
- *rms_index()*: ensemble des valeurs quantifiées de *rms(r)* pour toutes les régions.

La catégorie assignée à une région détermine les paramètres de quantification et de codage pour la région et le nombre total prévu des bits nécessaires pour représenter les coefficients MLT quantifiés de la région. En raison du codage de Huffman à longueur variable, le nombre réel des bits variera selon les statistiques des coefficients MLT d'une région. Ainsi, parmi les seize catégorisations calculées selon les critères décrits plus loin, la catégorisation la mieux adaptée sera sélectionnée pour les fins de la transmission.

Le nombre de bits prévu pour chaque catégorie (0-7) est prédéterminé dans le Tableau 1.

Tableau 1/G.722.1 – Nombre de bits prévu pour chaque catégorie

Catégorie	Bits de code par région selon la catégorie (voir <code>expected_bits_table[]</code> dans le code C)
0	52
1	47
2	43
3	37
4	29
5	22
6	16
7	0

3.4.1 Détermination du nombre estimé de bits disponibles

A partir du nombre réel de bits (*number_of_available_bits*), le programme calcule comme suit le nombre estimé de bits disponibles (*estimated_number_of_available_bits*):

si:

$$number_of_available_bits > 320,$$

alors:

$$estimated_number_of_available_bits = 320 + ((number_of_available_bits - 320) * 5/8)$$

La valeur *estimated_number_of_available_bits* est toujours inférieure au nombre réel de bits, ce qui laisse une marge de manœuvre dans le processus de catégorisation.

3.4.2 Calcul de la catégorisation initiale

Pour n'importe quelle valeur entière de décalage (*offset*) dans la plage -32 à 31, l'assignation des catégories est donnée par:

$$category(r) = \text{MAX} \{0, \text{MIN} \{7, (offset - rms_index(r))/2 \}\}$$

où:

$$0 \leq r < number_of_regions.$$

La même valeur *offset* est utilisée pour toutes les régions. Le nombre prévu total des bits de code MLT est le suivant:

$$expected_number_of_code_bits = \sum_{r=0}^{13} expected_bits_table(category(r))$$

La valeur *offset* est ensuite déterminée en fonction du décalage maximal qui satisfait à la formule:

$$expected_number_of_code_bits \geq estimated_number_of_available_bits - 32$$

3.4.3 Production des quinze autres catégories

Une fois que la catégorisation initiale a été calculée, les quinze autres doivent également l'être. Dans le cas de chaque nouvelle catégorisation, la catégorie est établie pour une seule région par rapport à la catégorisation précédente. La méthode de détermination des autres catégorisations est la suivante:

$$initial_categorization(r) = \text{MAX} \{0, \text{MIN} \{7, (offset - rms_index(r))/2 \}\}$$

où:

$$0 \leq r < \text{number_of_regions}$$

création des variables temporaires:

max_category(r)

max_bits

min_category(r)

min_bits

max_category(r) = initial_categorization (r)

min_category(r) = initial_categorization (r)

max_bits = expected_number_of_code_bits

min_bits = expected_number_of_code_bits.

Pour chacune des quinze autres catégorisations, la comparaison suivante est ensuite effectuée:

si:

$$\text{max_bits} + \text{min_bits} \leq 2 * \text{estimated_number_of_available_bits}$$

alors:

une nouvelle catégorisation est requise pour un nombre de bits prévu supérieur.

Pour les régions *r* où:

$$\text{max_category}(r) > 0$$

la région où la fonction se situe au minimum est déterminée comme suit:

$$\text{offset} - \text{rms_index}(r) - 2 * \text{max_category}(r)$$

S'il existe plusieurs régions où cette fonction se situe au même minimum, mettre *r* à la valeur qui correspond à la fréquence la plus basse pour la région.

La catégorie de cette région dans *max_category(r)* est ensuite décrétementée de 1, après quoi le nombre prévu de bits correspondant à la nouvelle catégorisation est calculé de nouveau et *max_bits* est mis à la valeur obtenue.

Sinon:

une catégorisation est requise pour un nombre de bits prévu inférieur.

Pour les régions *r* où:

$$\text{min_category}(r) < 7$$

la région où la fonction se situe au maximum est déterminée comme suit:

$$\text{offset} - \text{rms_index}[r] - 2 * \text{min_category}(r)$$

S'il existe plusieurs régions où cette fonction se situe au même maximum, mettre *r* à la valeur qui correspond à la fréquence la plus élevée pour la région.

La catégorie de cette région dans *min_category(r)* est ensuite incrémentée de 1, après quoi le nombre prévu de bits correspondant à la nouvelle catégorisation est calculé de nouveau et *min_bits* est mis à la valeur obtenue.

De cette façon, seize catégorisations uniques sont produites, ordonnées selon le nombre de bits prévu (voir paragraphe 6). La catégorisation 0 correspond au nombre de bits prévu supérieur et la catégorisation 15, au nombre inférieur. Chaque catégorisation est identique à la voisine, sauf dans une région où l'entrée de catégorie différera de 1. Par exemple, la région 5 de la catégorisation 7

pourra comporter une catégorie 2, alors que la région 5 de la catégorisation 8 pourra comporter une catégorie 3 tout en demeurant identique à la catégorisation 7 pour ce qui est des autres régions.

Le paragraphe 6 présente un organigramme détaillé de la procédure de catégorisation.

3.5 Codage de Huffman à combinaison en vecteurs et quantification scalaire (SQVH, *scalar quantized vector Huffman coding*)

Dans les régions à valeurs de catégorie 0-6, les coefficients MLT sont constitués de parties indiquant le signe et de parties indiquant la grandeur. Les parties indiquant la grandeur sont normalisées par la valeur quantifiée de $rms(r)$, puis soumises à une quantification scalaire avec extension de zone morte, à une combinaison en vecteurs et à un codage de Huffman. Les régions auxquelles une catégorie 7 a été assignée ne sont pas traitées de cette façon et ne contiennent aucun bit à transmettre.

Pour chaque région r , le codeur commence par normaliser et quantifier la valeur absolue de chaque coefficient MLT, $mlt(i)$, afin de produire l'indice de quantification, $k(i)$:

$$k(i) = \text{MIN} \{ \text{partie entière de } (x * \text{valeur absolue de } (mlt(20r + i)) + \text{deadzone_rounding}), k_{\text{max}} \}$$

où l'indice dans une région particulière est:

$$0 \leq i < 20$$

et:

$$x = 1/(\text{stepsize} * (\text{valeur quantifiée de } rms(r)))$$

et:

les valeurs $stepsize$, $deadzone_rounding$ et k_{max} sont indiquées au Tableau 2.

Tableau 2/G.722.1 – Table des constantes utilisées par la procédure SQVH

Catégorie	stepsize (pas)	deadzone_rounding (arrondi)	kmax
0	$2^{-1,5}$	0,3	13
1	$2^{-1,0}$	0,33	9
2	$2^{-0,5}$	0,36	6
3	$2^{0,0}$	0,39	4
4	$2^{0,5}$	0,42	3
5	$2^{1,0}$	0,45	2
6	$2^{1,5}$	0,5	1

Les indices $k()$ sont combinés en indices vectoriels; les propriétés des vecteurs diffèrent selon la catégorie. Chaque région contient vpr vecteurs prédéfinis à dimension vd , conformément à la définition du Tableau 3 et à l'illustration de la Figure 3. L'ensemble des valeurs scalaires $k()$ correspond à un vecteur unique indiqué par un indice, de la façon suivante:

$$vector_index(n) = \sum_{j=0}^{vd-1} k(n \times vd + j)(k_{\text{max}} + 1)^{(vd-(j+1))}$$

où: $0 \leq n \leq vpr - 1$ représente le n^{e} vecteur dans la région r

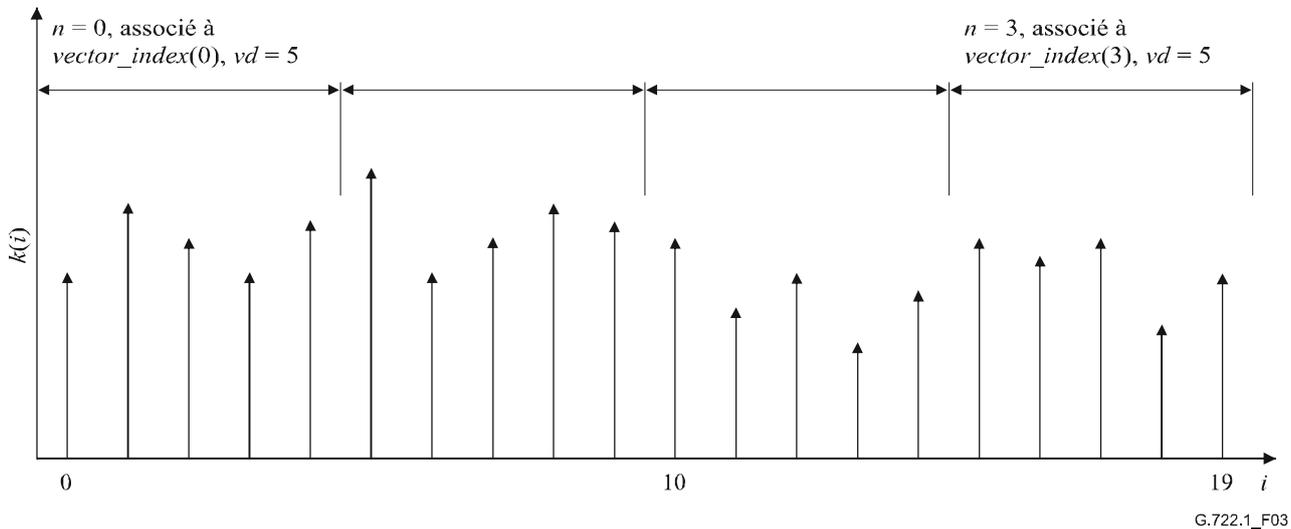
et:

j = indice de la j^e valeur de $k()$ pour une région et un vecteur donnés

vd = dimension du vecteur pour une catégorie donnée

vpr = nombre de vecteurs par région pour une catégorie donnée

$kmax$ = valeur maximale de $k()$ pour une catégorie donnée, selon le Tableau 2



NOTE – Chaque indice vectoriel représente vd coefficients MLT quantifiés.

Figure 3/G.722.1 – Illustration de la répartition d'une région à catégorie 6 selon une série de cinq vecteurs de dimension ($vd = 5$), à raison de quatre vecteurs par région ($vpr = 4$ et $0 \leq n < 4$)

Le Tableau 3 donne les valeurs de vd , vpr et u , où $u = (kmax + 1)^{vd}$ représente le nombre de valeurs distinctes qu'un vecteur peut prendre dans une catégorie quelconque.

Tableau 3/G.722.1 – Définition des constantes vd , vpr et u

Catégorie	vd	vpr	u
0	2	10	196
1	2	10	100
2	2	10	49
3	4	5	625
4	4	5	256
5	5	4	243
6	5	4	32

Les tables `mlt_sqvh_bitcount_category_0[]` à `mlt_sqvh_bitcount_category_6[]` précisent le nombre de bits nécessaires pour représenter un vecteur, $vector_index(n)$, d'une catégorie donnée. Elles indiquent aussi le nombre de bits nécessaires pour les entrées de mot de code correspondantes dans les tables `mlt_sqvh_code_category_0[]` à `mlt_sqvh_code_category_6[]`. Le nombre de bits ne tient pas compte des bits de signe. Les valeurs de $k() = 0$ ne requièrent pas de bit de signe.

Le nombre de bits (y compris les bits de signe) réellement nécessaires afin de représenter les coefficients MLT, pour une région r et une catégorie y , est obtenu comme suit:

$$\text{number_of_region_bits}(r) = \sum_{n=0}^{vpr-1} \text{mlt_svqh_bitcount_category_y}(\text{vector_index}(n))$$

+ (nombre de bits de signe du n^{e} vecteur)

3.6 Commande de débit

Le nombre total de bits réellement nécessaires pour représenter la trame est calculé pour chaque catégorisation, y compris les bits servant à représenter l'enveloppe d'amplitude, les quatre bits de commande de catégorisation et les bits utilisés pour les coefficients MLT. Il reste ensuite à sélectionner la meilleure catégorisation de transmission et à indiquer cette sélection à l'aide des bits de commande de catégorisation.

En premier lieu, les catégorisations dont le total des bits dépasse le nombre alloué sont éliminées. Parmi les catégorisations restantes, celle dont l'indice est le plus bas est sélectionnée. Par exemple, si les catégorisations 0 à 3 utilisent trop de bits et que la catégorisation 4 respecte le nombre alloué, la catégorisation 4 sera sélectionnée.

Si aucune catégorisation n'utilise un total de bits respectant le nombre alloué, la catégorisation la plus rapprochée (normalement 15) sera sélectionnée. Les codes de bit seront transmis jusqu'à ce que soit atteint le total alloué pour la trame.

Il peut arriver que le nombre de bits dont le codeur a besoin pour représenter une trame audiofréquence de 20 ms soit inférieur au nombre alloué de bits par trame (480 ou 640 bits). Dans ce cas, les bits qui demeurent inutilisés à la fin du flux binaire sont tous mis à 1.

3.7 Transmission des indices vectoriels MLT

Les indices vectoriels sont transmis dans l'ordre des fréquences, des basses fréquences aux hautes fréquences. Ils sont soumis à un codage de longueur variable, défini par les tables de code $\text{mlt_svqh_bitcount_category_x}[]$ et $\text{mlt_svqh_code_category_x}[]$ (où x représente la valeur de la catégorie, $0 \leq x \leq 6$). Les bits les plus à gauche (ou les plus significatifs) sont transmis en premier. Les bits de signe, associés aux coefficients MLT autres que zéro de chaque vecteur, sont transmis immédiatement après le code à longueur variable de leur indice vectoriel respectif. Ils sont également transmis dans l'ordre des fréquences. Le bit de signe est mis à "1" pour les nombres positifs.

3.8 Flux binaire

Le nombre total de bits d'une trame est de 480 ou de 640, respectivement pour les débits de 24 kbit/s et de 32 kbit/s. Bien que le nombre de bits d'une trame soit fixe, sauf pour le paramètre des bits de commande de catégorisation, tous les autres paramètres sont représentés par des codes à longueur variable ou par un nombre variable de bits. La Figure 4 illustre cet aspect, ainsi que l'ordre des champs des paramètres transmis. Tous les codes à longueur variable et les bits de commande de catégorisation sont transmis dans l'ordre, du bit le plus à gauche (le plus significatif) au bit le plus à droite (le moins significatif).

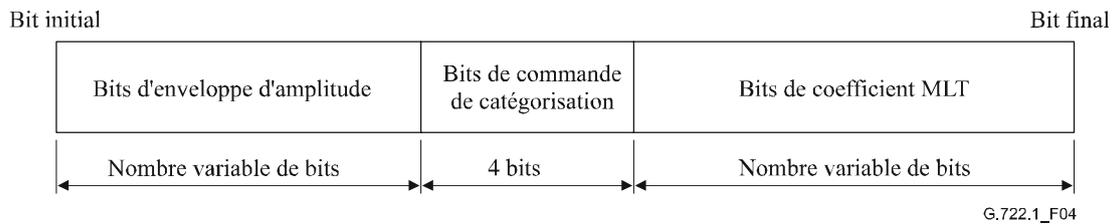


Figure 4/G.722.1 – Flux binaire et ordre des principaux champs transmis

4 Décodeur

Pour chacune des trames, les cinq premiers bits, qui représentent l'indice d'amplitude de la région zéro, sont tout d'abord décodés. Par la suite, les autres régions font l'objet d'un décodage de Huffman et d'une reconstitution. Les quatre bits de commande de catégorisation sont ensuite décodés, ce qui permet de déterminer quelle catégorisation parmi les seize possibles a été sélectionnée et transmise par le codeur. Les bits de code qui restent dans la trame représentent les coefficients MLT quantifiés et sont décodés selon l'information de catégorie de chaque région. Comme dans le cas du codeur, la procédure de catégorisation du décodeur utilise l'enveloppe d'amplitude avec le nombre de bits qu'il reste à décoder (dans la trame courante) et calcule l'ensemble des seize catégorisations possibles.

Il se peut que le codeur ait assigné une catégorie 7 à certaines régions. Aucun coefficient MLT n'est alors transmis pour représenter ces régions, et les régions à catégorie 7 doivent être reconstituées selon une technique appelée *remplissage de bruit*. La grandeur MLT moyenne de ces régions est tirée de l'enveloppe d'amplitude. Plutôt que de remettre les coefficients MLT de la catégorie 7 à zéro, le décodeur fixe la valeur de leur amplitude en proportion de la valeur moyenne du coefficient MLT pour la région, et le signe de chaque coefficient est établi de façon aléatoire. Plusieurs méthodes permettent de déterminer les signes des coefficients, notamment le recours à un simple générateur de nombres pseudo-aléatoires.

Le remplissage de bruit s'applique aussi aux catégories 5 et 6, car bon nombre des coefficients MLT peuvent être quantifiés à zéro dans ces catégories. Les valeurs transmises à zéro sont mises à de petites fractions de la grandeur moyenne pour la région. A nouveau, les signes sont déterminés de façon aléatoire.

Une table prédéterminée renferme les valeurs de reconstitution des coefficients normalisés qui avaient fait l'objet d'une quantification scalaire à des valeurs autres que zéro. Les valeurs reconstituées sont ensuite mises à l'échelle, à l'aide de la valeur appropriée de $rms(r)$. Les quarante coefficients MLT représentant des fréquences supérieures à 7 kHz sont mis à zéro. Après la reconstitution des coefficients MLT, la transformation inverse à modulation et chevauchement (IMLT, *inverse modulated lapped transform*) produit 320 nouveaux échantillons dans le domaine temporel.

Exception faite de l'opération finale de chevauchement et d'addition de l'IMLT, l'information que reçoit le décodeur dans chaque trame est indépendante de l'information de la trame précédente.

4.1 Décodage de l'enveloppe d'amplitude

Les cinq premiers bits de la trame représentent $rms_index(0)$. Pour les autres régions, les codes à longueur variable de $differential_rms_index(r)$ sont ensuite décodés selon les tables $differential_region_power_bits[][]$ et $differential_region_power_codes[][]$ spécifiées dans le code C; les indices de quantification de ces régions sont reconstitués comme suit:

$$rms_index(r) = rms_index(r - 1) + differential_rms_index(r),$$

où:

$$1 \leq r < \text{number_of_regions}.$$

4.2 Détermination de la catégorisation

Après avoir décodé l'enveloppe d'amplitude, le décodeur détermine le nombre de bits qui restent pour représenter les coefficients MLT, de la façon suivante:

$$\text{bits disponibles} = \text{bits par trame} - \text{bits d'enveloppe d'amplitude} - 4 \text{ (bits de commande de catégorisation)}$$

Selon la même procédure de catégorisation que le codeur, le décodeur calcule le même ensemble de seize catégorisations possibles. Les quatre bits de commande de catégorisation indiquent quelle catégorisation a servi à coder les coefficients MLT et devrait donc être utilisée par le décodeur.

4.3 Décodage des coefficients MLT

Pour chaque région, les codes à longueur variable représentant les vecteurs MLT sont décodés selon les tables de catégories appropriées. Les tables `mlt_svqh_bitcount_category_x[]` et `mlt_svqh_code_category_x[]` sont utilisées à cette fin dans le code C (où `_x` représente la valeur de catégorie, $0 \leq x \leq 6$). Les différents indices de quantification des coefficients MLT, $k(i)$, d'une région sont récupérés à partir de l'indice vectoriel, de la façon suivante:

$$k(i) = \left\lfloor \frac{\text{vector_index}(n)}{(kmax+1)^j} \right\rfloor \text{MOD}(kmax+1)$$

où:

$\lfloor z \rfloor$ désigne la valeur entière la plus élevée qui soit inférieure ou égale à z

$$i = (n+1)vd - j - 1$$

$$0 \leq j \leq vd - 1$$

$0 \leq n \leq vpr-1$, représente le n^e vecteur dans la région r ,

et:

vd = dimension du vecteur pour une catégorie donnée

$kmax$ = valeur maximale de $k()$ pour une catégorie donnée, selon le Tableau 2.

La reconstitution des coefficients MLT repose sur les tables de centroïdes contenues dans `mlt_quant_centroid[][]` (code C). Les amplitudes des coefficients MLT sont reconstituées par le calcul du produit de $rms(r)$, dans la région d'intérêt, et du centroïde que spécifie l'indice vectoriel décodé. Le signe des valeurs autres que zéro est établi à partir du bit de signe.

4.4 Remplissage de bruit

Aucune amplitude de coefficient MLT n'est codée pour les régions auxquelles la catégorie 7 a été assignée. Dans les catégories 5 et 6, la valeur élevée des échelons est telle que la plupart des coefficients MLT codés sont mis à zéro; les codes zéro sont remplacés par des coefficients de signe aléatoire et d'amplitude proportionnelle à $rms(r)$. Le Tableau 4 définit les constantes de proportionnalité.

Tableau 4/G.722.1 – Constantes de proportionnalité du remplissage de bruit

Catégorie	Constante par défaut de proportionnalité du remplissage de bruit
5	0,176777
6	0,25
7	0,707107

4.5 Bits insuffisants

Il est possible qu'il existe des trames pour lesquelles le codeur a manqué de bits avant de terminer le codage de la dernière région autre que de catégorie 7. Dans ces cas, le décodeur traite la région en question et toutes les régions restantes comme ayant une assignation de catégorie 7.

4.6 Effacement de trame

Si le décodeur est informé (au moyen d'un mécanisme de signalisation externe non défini dans la présente Recommandation) qu'une trame s'est perdue ou corrompue, il répète les coefficients MLT décodés de la trame précédente. Pour ce faire, il les convertit au domaine temporel et exécute l'opération de chevauchement et d'addition en utilisant l'information décodée des trames précédente et suivante. Si la trame précédente s'est également perdue ou corrompue, le décodeur remet à zéro tous les coefficients MLT des trames courantes.

4.7 Transformation MLT inverse (IMLT, *inverse MLT*)

Chaque opération IMLT est appliquée à 320 coefficients afin de produire 320 échantillons audio dans le domaine temporel. L'opération IMLT peut se décomposer en une transformation DCT de type IV, suivie d'une opération de fenêtrage, de chevauchement et d'addition.

La transformation DCT de type IV est la suivante:

$$u(n) = \sum_{m=0}^{319} \sqrt{\frac{2}{320}} \cos\left(\frac{\pi}{320}(m+0,5)(n+0,5)\right) mlt(m) \quad \text{pour } 0 \leq n \leq 319$$

L'opération de fenêtrage, de chevauchement et d'addition s'applique à la moitié des échantillons provenant de la sortie DCT de la trame courante et à la moitié de ceux provenant de la sortie DCT de la trame précédente:

$$y(n) = w(n)u(159 - n) + w(319 - n)u_old(n) \quad \text{pour } 0 \leq n \leq 159$$

$$y(n + 160) = w(160 + n)u(n) - w(159 - n)u_old(159 - n) \quad \text{pour } 0 \leq n \leq 159,$$

où:

$$w(n) = \sin\left(\frac{\pi}{640}(n+0,5)\right) \quad \text{pour } 0 \leq n \leq 319$$

La moitié inutilisée de $u()$ est stockée sous le nom $u_old()$ et utilisée dans la trame suivante:

$$u_old(n) = u(n + 160) \quad \text{pour } 0 \leq n \leq 159$$

5 Code C

Le code source en langage C ANSI en virgule fixe à 16/32 bits, qui fait partie intégrante de la présente Recommandation se divise en un certain nombre de fichiers. La description algorithmique donnée en langage C a priorité sur les textes contenus dans le corps de la présente Recommandation et dans l'Annexe C. Ces fichiers sont énumérés dans le Tableau 5.

NOTE – Pour plus de précisions sur le code source en langage C en virgule flottante, voir l'Annexe B.

Tableau 5/G.722.1 – Liste des fichiers logiciels propres au code source en virgule fixe de la Rec. UIT-T G.722.1

Nom du fichier	Description
basop32.c	Opérateurs arithmétiques de base
coef2sam.c	MLT inverse
common.c	Routines utilisées par le codeur et le décodeur
count.c	Fonctions de calcul automatique de la complexité
dct4_a.c	DCT directe
dct4_s.c	DCT inverse
decode.c	Programme principal du décodeur
decoder.c	Routines du décodeur
encode.c	Programme principal du codeur
encoder.c	Routines du codeur
huff_tab.c	Codage de Huffman pour le codeur et le décodeur
sam2coef.c	MLT directe
tables.c	Tableaux relatifs à la MLT directe et inverse
basop32.h	Définitions des opérateurs arithmétiques de base
count.h	Définitions des fonctions de mesure de la complexité
dct4_a.h	Définitions des tableaux relatifs à la DCT directe
dct4_s.h	Définitions des tableaux relatifs à la DCT inverse
defs.h	Définitions des paramètres
huff_defs.h	Définitions du codage de Huffman
huff_tab.h	Déclaration des tables de Huffman
tables.h	Définitions des tableaux relatifs à la MLT directe et inverse
typedefs.h	Définitions des types de données et des constantes

Une fois que le programme autonome a été compilé dans le fichier du codeur, *encode*, et dans le fichier du décodeur, *decode*, la ligne de commande permettant d'utiliser le codeur s'écrit comme suit:

```
encode <type de flux binaire> <fichier audio d'entrée> <fichier du flux de sortie> <débit>
<largeur de bande>
```

```
decode <type de flux binaire> <fichier du flux d'entrée> <fichier audio de sortie> <débit>
<largeur de bande>
```

où:

- <type de flux binaire> =
- 0, indique l'utilisation du flux binaire compacté;
 - 1, indique l'utilisation du format de flux binaire de la Rec. UIT-T G.192 aux fins de test.
- <fichier audio d'entrée> = nom du fichier audio MIC linéaire à 16 bits dans lequel sont lus les échantillons.
- <fichier audio de sortie> = nom du fichier audio MIC linéaire à 16 bits dans lequel est enregistrée la sortie décodée.

- <fichier du flux d'entrée> = nom du fichier dans lequel est lu le flux binaire d'entrée.
- <fichier du flux de sortie> = nom du fichier dans lequel est enregistré le flux binaire codé de sortie.
- <débit> =
- pour le mode audio avec largeur de bande de 7 kHz (corps du texte): 24000 ou 32000 (en bit/s);
 - pour le mode audio avec largeur de bande de 14 kHz (Annexe C): 24000, 32000 ou 48000 (en bit/s).
- <largeur de bande> =
- 7000 pour le mode audio à 7 kHz (corps du texte);
 - 14000 pour le mode audio à 14 kHz (Annexe C).

6 Organigramme de la procédure de catégorisation

Le présent paragraphe décrit en détail la procédure du calcul de catégorisation, utilisée par le codeur et le décodeur. La procédure se subdivise en vingt-cinq étapes, qu'il est possible de représenter sous forme d'organigramme.

Le présent paragraphe définit la variable ci-dessous et y fait référence:

$$category.X[r]$$

est la catégorie assignée à la région r pour la catégorisation X ,

où:

$$0 \leq X < 16$$

$$0 \leq r < 14 \text{ (number_of_regions)}$$

$$0 \leq category.X[r] \leq 7$$

ETAPE (0)

Calculer le nombre de bits disponibles en prenant le nombre alloué de bits par trame (par exemple 480 pour un débit de 24 kbit/s). Soustraire le nombre de bits utilisés pour représenter l'enveloppe d'amplitude, puis les quatre bits de commande de catégorisation représentant la sélection de catégorisation. Modifier ensuite le résultat comme suit:

si:

$$number_of_available_bits > 320,$$

alors:

$$estimated_number_of_available_bits = 320 + ((number_of_available_bits - 320) * 5/8)$$

(On compense ainsi les différences dans les statistiques relatives aux assignations de catégorie pour différents débits.)

ETAPE (1)

Affecter les tables temporaires:

initial_categorization[number_of_regions]
max_category[number_of_regions]
min_category[number_of_regions]
temp_category_balances[32]

Affecter les variables temporaires:

offset
delta
expected_bits
max_expected_bits
min_expected_bits
max_pointer
min_pointer
categorization_count

ETAPE (2)

Initialiser:

offset = -32
delta = 32

ETAPE (3)

Calculer les assignations de catégorie non limitées pour chaque région:

$initial_categorization[r] = (offset + delta - rms_index[r]) / 2$

ETAPE (4)

Limiter *initial_categorization*[] pour chaque région:

si:

$initial_categorization[r] < 0$

alors:

$initial_categorization[r] = 0$

si:

$initial_categorization[r] > 7$

alors:

$initial_categorization[r] = 7$

ETAPE (5)

La table prédéterminée *expected_bits_table*[8] contient le nombre moyen des bits de chaque catégorie. Elle permet de calculer le nombre total prévu de bits pour cette catégorisation:

$$expected_bits = \sum_{r=0}^{number_of_regions-1} expected_bits_table[initial_categorization[r]]$$

ETAPE (6)

si:

$expected_bits \geq estimated_number_of_available_bits - 32,$

alors:

$offset = offset + delta.$

ETAPE (7)

$delta = delta / 2$

ETAPE (8)

si:

$$\delta > 0,$$

alors:

aller à l'ETAPE (3),

sinon:

aller à l'ETAPE (9).

ETAPE (9)

$$initial_categorization[r] = (offset - rms_index[r])/2$$

ETAPE (10)Limiter $initial_categorization[r]$ pour chaque région comme à l'ETAPE (4).**ETAPE (11)**Calculer $expected_bits$ pour $initial_categorization[]$ comme à l'ETAPE (5).**ETAPE (12)**

Initialiser:

$$max_category[r] = initial_categorization[r]$$

$$min_category[r] = initial_categorization[r]$$

$$max_bits = expected_bits$$

$$min_bits = expected_bits$$

$$max_pointer = 16$$

$$min_pointer = 16$$

$$categorization_count = 1$$

ETAPE (13)

si:

$$max_bits + min_bits \leq 2 * estimated_number_of_available_bits$$

alors:

aller à l'ETAPE (14)

sinon:

aller à l'ETAPE (16)

ETAPE (14)Pour les régions r où

$$max_category[r] > 0$$

trouver la région où la fonction se situe au minimum:

$$offset - rms_index[r] - 2 * max_category[r]$$

S'il existe plusieurs régions où cette fonction se situe au même minimum, mettre r à la valeur qui correspond à la fréquence la plus basse pour la région.

ETAPE (15)

$max_pointer = max_pointer - 1$
 $temp_category_balances[max_pointer] = r$
 $max_bits = max_bits - expected_bits_table[max_category[r]]$
 $max_category[r] = max_category[r] - 1$
 $max_bits = max_bits + expected_bits_table[max_category[r]]$
 aller à l'ETAPE (18)

ETAPE (16)

Pour les régions r où

$min_category[r] < 7$

trouver la région où la fonction se situe au maximum:

$offset - rms_index[r] - 2 * min_category[r]$

S'il existe plusieurs régions où cette fonction se situe au même maximum, mettre r à la valeur qui correspond à la fréquence la plus élevée pour la région.

ETAPE (17)

$temp_category_balances[min_pointer] = r$
 $min_pointer = min_pointer + 1$
 $min_bits = min_bits - expected_bits_table[min_category[r]]$
 $min_category[r] = min_category[r] + 1$
 $min_bits = min_bits + expected_bits_table[min_category[r]]$

ETAPE (18)

$categorization_count = categorization_count + 1$

si:

$categorization_count < 16$

alors:

aller à l'ETAPE (13)

sinon:

aller à l'ETAPE (19)

ETAPE (19)

Copier $max_category[]$ dans $category.0 []$.

Pour toutes les régions

$Category.0 [r] = max_category[r]$

ETAPE (20)

$n = 1$

ETAPE (21)

Copier le contenu de la $(n - 1)^e$ catégorisation dans la n^e catégorisation.

Pour toutes les régions:

$category.n[r] = category.n - 1[r]$

ETAPE (22)

Incrémenter l'assignation de catégorie pour la région indiquée par la table *temp_category_balances*[*max_pointer*]:

$$\text{category.n} [\text{temp_category_balances}[\text{max_pointer}]] = \text{category.n} [\text{temp_category_balances} [\text{max_pointer}]] + 1$$

ETAPE (23)

$$\text{max_pointer} = \text{max_pointer} + 1$$
$$n = n + 1$$

ETAPE (24)

si:

$$n < 16$$

alors:

aller à l'ETAPE (21)

sinon:

FIN

Annexe A

Format des paquets, identificateurs de capacité et paramètres de capacités

A.1 Références

- [1] Recommandation UIT-T H.225.0 (2003), *Protocoles de signalisation d'appel et paquets des trains monomédias pour les systèmes de communication multimédias en mode paquet.*

A.2 Structure des paquets pour les trames G.722.1

L'algorithme de codage audiofréquence, défini dans la présente Recommandation, code des signaux audio large bande avec une bande passante de 50 Hz-7 kHz en un signal numérique de débit de 24 ou 32 kbit/s, en utilisant des trames de 20 ms et une fréquence d'échantillonnage de 16 kHz. Le débit peut être modifié à toute limite de trame de 20 ms, bien qu'aucune notification de changement de débit ne soit fournie dans la bande avec le flux binaire. Pour un débit de 24 kbit/s, 480 bits (60 octets) sont produits par trame, et pour un débit de 32 kbit/s, 640 bits (80 octets). En conséquence, les deux débits permettent l'alignement d'octets sans recours à des bits de remplissage.

De même, l'algorithme étendu défini dans l'Annexe C code des signaux audio avec une bande passante de 50 Hz-14 kHz, la même structure de trame et les mêmes caractéristiques de format de paquets. Les principaux paramètres sont récapitulés dans le Tableau A.1.

Tableau A.1/G.722.1 – Paramètres pour les modes G.722.1

	Fréquence d'échantillonnage (kHz)	Longueur de trames (ms)	Taille de la trame (bits/octets)
24 kbit/s G.722.1	16	20	480/60
32 kbit/s G.722.1	16	20	640/80
24 kbit/s Annexe C/G.722.1	32	20	480/60
32 kbit/s Annexe C/G.722.1	32	20	640/80
48 kbit/s Annexe C/G.722.1	32	20	960/120

Le nombre de bits par trame est fixe. Toutefois, à l'intérieur de cette trame fixe, la présente Recommandation utilise un codage à longueur variable (par exemple le codage de Huffman) pour représenter la plupart des paramètres codés. A l'exception du paramètre de bits de commande de catégorisation, tous les autres paramètres du flux binaire sont représentés par des codes à longueur variable, c'est-à-dire comportant un nombre variable de bits. La Figure A.1 illustre cet aspect ainsi que l'ordre des champs des paramètres transmis. Tous les codes à longueur variable et les bits de commande de catégorisation sont transmis dans l'ordre, du bit le plus à gauche (le bit de plus fort poids (MSB, *most significant bit*)) au bit le plus à droite (le bit de plus faible poids (LSB, *least significant bit*)). L'utilisation du codage de Huffman signifie qu'il n'est pas possible d'identifier les divers paramètres/champs de codage convenus dans le flux binaire sans commencer par décoder entièrement toute la trame.

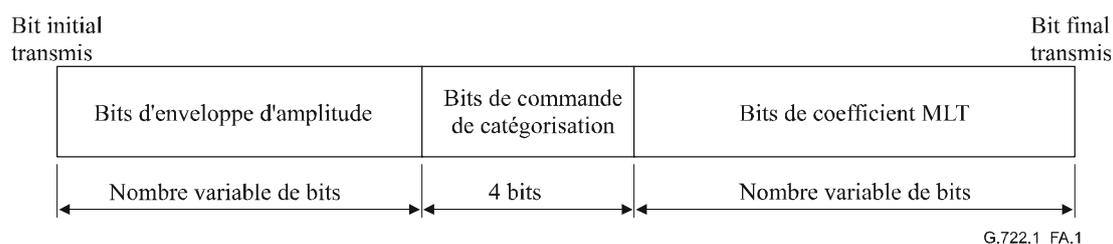


Figure A.1/G.722.1 – Principaux champs de flux binaire et leur ordre de transmission

La Figure A.2 représente la manière dont le flux binaire G.722.1 s'intègre à une charge utile RTP [1] dont les octets sont alignés conformément à la description donnée dans la Rec. UIT-T H.225.0.

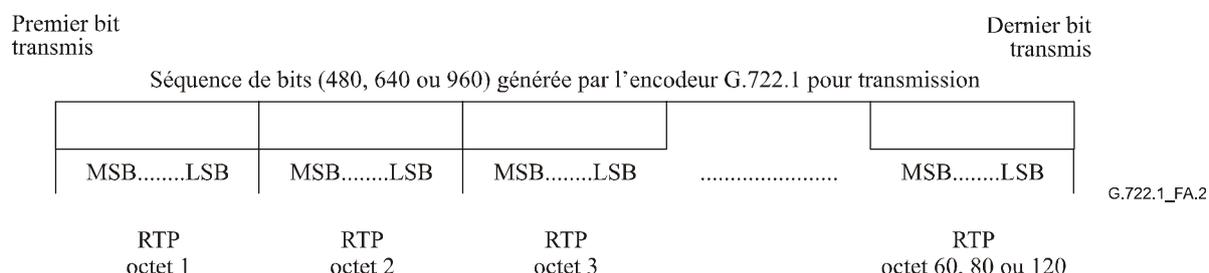


Figure A.2/G.722.1 – Flux binaire du codeur G.722.1

Tout paquet RTP contiendra uniquement des trames G.722.1 du même débit et de la même fréquence d'échantillonnage. Le timbre horodateur RTP doit être exprimé en unités de 1/16000^e de seconde, sauf pour les modes définis dans l'Annexe C/G.722.1, qui doivent utiliser des unités de 1/32000^e de seconde.

Le flux binaire est subdivisé en une séquence d'octets (60, 80 ou 120 en fonction du mode), chaque octet étant à son tour mappé en un octet RTP.

A.3 Identificateurs et paramètres de capacité utilisés avec la Rec. UIT-T H.245

Le paramètre **GenericCapability** est utilisé dans la Rec. UIT-T H.245 pour l'échange de capacités de la Rec. G.722.1 initiale. Les tableaux nécessaires d'identificateurs et paramètres de capacités sont définis ci-après.

A.3.1 Identificateurs et paramètres de capacité pour la Rec. UIT-T G.722.1 initiale

Dans les messages capability et OpenLogicalChannel H.245, le champ maxBitRate indiqué dans le Tableau A.2 ci-dessus doit être utilisé pour signaler un mode G.722.1 unique à un débit binaire exact (en dépit de la définition qui est donnée de ce champ dans la Rec. UIT-T H.245).

Tableau A.2/G.722.1 – Tableau des identificateurs de capacité pour G.722.1

Nom de la capacité	Rec. UIT-T G722.1
Classe de la capacité	Audio
Type d'identificateur de la capacité	Standard
Valeur de l'identificateur de la capacité	{ itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) 0 }
maxBitRate	Ce paramètre doit être mis à une valeur de 32 000, représentant 32 kbit/s, ou de 24 000, représentant 24 kbit/s
Collapsing	Ce champ doit contenir les paramètres de capacité indiqués ci-dessous
nonCollapsing	Ce champ ne doit pas être inclus
NonCollapsingRaw	Ce champ n'est pas utilisé
Transport	Ce champ n'est pas utilisé

Par exemple, un système qui inclut un identificateur de capacité de { itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) 0 } avec un champ maxBitRate d'une valeur de 32000 indique qu'il peut fonctionner conformément à la présente Recommandation à 32 kbit/s uniquement, et pas nécessairement aussi à tout débit binaire inférieur. Si un système peut fonctionner à la fois à 24 et à 32 kbit/s, il doit le faire savoir au moyen de deux messages GenericCapability, indiquant pour le champ maxBitRate une valeur de 24000 pour l'un de ces messages, et une valeur de 32000 pour l'autre.

NOTE – Les unités du champ maxBitRate indiquées dans le Tableau A.2, et l'utilisation de ce champ, diffèrent de la convention indiquée dans la Rec. UIT-T H.245 ainsi que des unités et de l'utilisation indiquées dans le Tableau A.4. Ces unités et utilisation, qui ont été établies il y a très longtemps, sont conservées pour assurer la continuité de l'interfonctionnement avec les systèmes mis en place.

Le Tableau A.3 ci-dessous définit le paramètre obligatoire maxFramesPerPacket. Dans une capacité de réception, ceci indique le nombre maximal de trames G.722.1 codées dans un seul paquet RTP qu'un récepteur est capable de décoder.

Tableau A.3/G.722.1 – Tableau de paramètres de capacité générique pour la Rec. UIT-T G.722.1, établissant le nombre maximal de trames autorisées dans un paquet RTP

Nom du paramètre	MaxFramesPerPacket
Description du paramètre	Il s'agit d'un paramètre Collapsing GenericParameter. La valeur du paramètre maxFramesPerPacket indique le nombre maximal de trames G.722.1 codées qui peuvent figurer dans un seul paquet RTP
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Obligatoire
Type de paramètre	unsignedMin
Remplace	Ce champ n'est pas utilisé

A.3.2 Identificateurs et paramètres de capacité pour les modes étendus de la Rec. UIT-T G.722.1

Les Tableaux A.4 à A.6 définissent les paramètres **GenericCapability** utilisés dans la Rec. UIT-T H.245 pour les modes étendus de la présente Recommandation, y compris l'Annexe C.

Tableau A.4/G.722.1 – Tableau des identificateurs de capacité pour les modes étendus de la présente Recommandation

Nom de la capacité	Extension de la Rec. UIT-T G.722.1
Classe de la capacité	Audio
Type d'identificateur de la capacité	Standard
Valeur de l'identificateur de la capacité	{ itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) extension (1) 0 }
maxBitRate	Ce paramètre doit être mis à une valeur de 480, représentant 48 kbit/s; de 320, représentant 32 kbit/s; ou de 240, représentant 24 kbit/s.
collapsing	Ce champ doit contenir les paramètres de capacité pour les modes étendus G.722.1 indiqués ci-dessous
nonCollapsing	Ce champ ne doit pas être inclus
NonCollapsingRaw	Ce champ n'est pas utilisé
Transport	Ce champ n'est pas utilisé

Le champ maxBitRate de cet identificateur de capacité est utilisé conformément à la Rec. UIT-T H.245. Il signale le débit binaire maximal pris en charge parmi les différents modes indiqués dans le paramètre supportedExtendedModes ci-dessous. Un seul identificateur de capacité { itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) extension (1) 0 } doit être utilisé pour signaler tous les modes G.722.1 étendus pris en charge.

Tableau A.5/G.722.1 – Tableau de paramètres de capacité générique pour les modes étendus de la Rec. UIT-T G.722.1, établissant le nombre maximal de trames autorisées dans un paquet RTP

Nom du paramètre	MaxFramesPerPacket
Description du paramètre	Il s'agit d'un paramètre Collapsing GenericParameter. La valeur du paramètre maxFramesPerPacket indique le nombre maximal de trames codées pour les modes étendus G.722.1 qui peuvent figurer dans un seul paquet RTP.
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Obligatoire
Type de paramètre	unsignedMin
Remplace	Ce champ n'est pas utilisé

Tableau A.6/G.722.1 – Tableau de paramètres de capacité générique pour les modes étendus de la Rec. UIT-T G.722.1, exposant les modes pris en charge

Nom du paramètre	supportedExtendedModes
Description du paramètre	<p>Ce paramètre est un tableau Booléen</p> <p>Si le bit 2 (valeur 64) est 1, cela indique le mode de l'Annexe C à 24 kbit/s</p> <p>Si le bit 3 (valeur 32) est 1, cela indique le mode de l'Annexe C à 32 kbit/s</p> <p>Si le bit 4 (valeur 16) est 1, cela indique le mode de l'Annexe C à 48 kbit/s</p> <p>Tous les autres bits sont réservés et seront mis à 0 et ignorés par les récepteurs</p> <p>Dans une capacité, lorsqu'un bit est mis à 1, cela signifie que le système peut fonctionner conformément au(x) mode(s) indiqué(s)</p> <p>Dans un message OpenLogicalChannel, un seul bit exactement doit être mis à 1, ce qui indique le mode à utiliser sur la voie logique</p> <p>NOTE – Au cas où serait défini ultérieurement un nombre de modes supérieur au nombre de bits réservés qu'il est possible de prendre en charge, des modes additionnels pourraient être signalés en affectant un autre paramètre pour les modes additionnels</p>
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Obligatoire
Type de paramètre	booleanArray
Remplace	Ce champ n'est pas utilisé

Annexe B

Implémentation en virgule flottante de la Rec. UIT-T G.722.1

B.1 Introduction

La présente annexe décrit l'implémentation en arithmétique à virgule flottante pour le mode à 7 kHz défini dans le corps de la présente Recommandation.

B.2 Description algorithmique

Cette version en virgule flottante de la présente Recommandation contient les mêmes étapes algorithmiques de base que la version en virgule fixe pour le mode à 7 kHz.

B.3 Code en langage C de l'ANSI

Est joint à la présente annexe le code en langage C conforme à l'ANSI, qui simule la version en virgule flottante de la présente Recommandation. La description algorithmique donnée en langage C a priorité sur les textes contenus dans le corps de la présente Recommandation et dans la présente annexe.

Les fichiers du Tableau B.1 contiennent le code source en langage C en virgule flottante.

Tableau B.1/G.722.1 – Liste des fichiers logiciels propres au code source en virgule flottante de la Rec. UIT-T G.722.1

Nom du fichier	Description
common.c	Routines utilisées par le codeur et le décodeur
dct4.c	DCT directe et inverse
decode.c	Programme principal du décodeur
decoder.c	Routines du décodeur
encode.c	Programme principal du codeur
encoder.c	Routines du codeur
rmlt_coefs_to_samples.c	MLT inverse
samples_to_rmlt_coefs.c	MLT
defs.h	Définitions des paramètres
huff_defs.h	Définitions du codage de Huffman
huff_tables.h	Déclaration des tables de Huffman

Une fois que le programme autonome (en virgule flottante) a été compilé dans le fichier du codeur, *encode*, et dans le fichier du décodeur, *decode*, la ligne de commande permettant d'utiliser le codeur s'écrit comme suit:

encode <type de flux binaire> <fichier audio d'entrée> <fichier du flux de sortie> <débit>

decode <type de flux binaire> <fichier du flux d'entrée> <fichier audio de sortie> <débit>

où:

- <type de flux binaire> =
- 0, indique l'utilisation du flux binaire compacté;
 - 1, indique l'utilisation du format de flux binaire de la Rec. UIT-T G.192 aux fins de test;
- <fichier audio d'entrée> nom du fichier audio MIC à 16 bits dans lequel sont lus les échantillons;

<fichier audio de sortie>	nom du fichier audio MIC à 16 bits dans lequel est enregistrée la sortie décodée;
<fichier du flux d'entrée>	nom du fichier dans lequel est lu le flux binaire d'entrée;
<fichier du flux de sortie>	nom du fichier dans lequel est enregistré le flux binaire codé de sortie;
<débit> =	24000 ou 32000 pour une exploitation respectivement à 24 kbit/s ou 32 kbit/s.

Annexe C

Mode à 14 kHz aux débits de 24, 32 et 48 kbit/s

C.1 Introduction

La présente annexe décrit le mode à 14 kHz aux débits de 24, 32 et 48 kbit/s de la présente Recommandation.

C.2 Description algorithmique

Ce mode à 14 kHz de la présente Recommandation comporte les mêmes étapes algorithmiques que le mode (à largeur de bande audio de 7 kHz) du corps du texte de la présente Recommandation, à ceci près que l'algorithme est doublé pour tenir compte du doublement de la largeur de bande audio.

Concrètement, l'algorithme de la présente annexe diffère de celui du corps du texte de la présente Recommandation sur les points suivants:

- a) doublement de la fréquence d'échantillonnage, qui passe de 16 à 32 kHz;
- b) doublement du nombre d'échantillons par trame, qui passe de 320 à 640 échantillons (même longueur de trame de 20 ms);
- c) doublement de la fenêtre de transformation qui passe de 640 à 1280 échantillons comme suit:

les entrées de chaque dispositif MLT sont constituées des 1280 échantillons audiofréquence les plus récents, $x(n)$,

où:

$x(0)$ est l'échantillon le plus ancien,

et:

$$0 \leq n < 1280$$

Le dispositif MLT produit 640 coefficients de transformation, $mlt(m)$,

où:

$$0 \leq m < 640$$

La transformation MLT est donnée par:

$$mlt(m) = \sum_{n=0}^{1279} \sqrt{\frac{2}{640}} \sin\left(\frac{\pi}{1280}(n+0,5)\right) \cos\left(\frac{\pi}{640}(n-319,5)(m+0,5)\right) x(n)$$

La transformation MLT peut se décomposer en opération de fenêtrage, de chevauchement et d'addition, suivie d'une transformation en cosinus discrète (DCT, *discrete cosine transform*) de type IV. L'opération de fenêtrage, de chevauchement et d'addition est donnée par:

$$v(n) = w(319-n)x(319-n) + w(320+n)x(320+n) \text{ pour } 0 \leq n \leq 319$$

$$v(n+320) = w(639-n)x(640+n) - w(n)x(1279-n) \text{ pour } 0 \leq n \leq 319$$

où:

$$w(n) = \sin\left(\frac{\pi}{1280}(n+0,5)\right) \text{ pour } 0 \leq n < 640$$

Si l'on combine $v(n)$ avec une transformée DCT de type IV, l'expression résultant de la transformation MLT est:

$$mlt(m) = \sum_{n=0}^{639} \sqrt{\frac{2}{640}} \cos\left(\frac{\pi}{640}(n+0,5)(m+0,5)\right) v(n) \text{ pour } 0 \leq m < 640$$

Chaque opération IMLT est appliquée à 640 coefficients afin de produire 640 échantillons audio dans le domaine temporel. L'opération IMLT peut se décomposer en une transformation DCT de type IV, suivie d'une opération de fenêtrage, de chevauchement et d'addition.

La transformation DCT de type IV est la suivante:

$$u(n) = \sum_{m=0}^{639} \sqrt{\frac{2}{640}} \cos\left(\frac{\pi}{640}(m+0,5)(n+0,5)\right) mlt(m) \text{ pour } 0 \leq n \leq 639$$

L'opération de fenêtrage, de chevauchement et d'addition s'applique à la moitié des échantillons provenant de la sortie DCT de la trame courante et à la moitié de ceux provenant de la sortie DCT de la trame précédente:

$$y(n) = w(n)u(319-n) + w(639-n)u_{old}(n) \text{ pour } 0 \leq n \leq 319$$

$$y(n+320) = w(320+n)u(n) - w(319-n)u_{old}(319-n) \text{ pour } 0 \leq n \leq 319,$$

où:

$$w(n) = \sin\left(\frac{\pi}{1280}(n+0,5)\right) \text{ pour } 0 \leq n \leq 639$$

La moitié inutilisée de $u(\)$ est stockée sous le nom $u_{old}(\)$ et utilisée dans la trame suivante:

$$u_{old}(n) = u(n+320) \text{ pour } 0 \leq n \leq 319$$

d) Ajouter de nouveaux tableaux DCT pour tenir compte du doublement de la largeur de bande comme suit:

un nouveau tableau $a_cos_msin_64[320][2]$ est ajouté dans $dct4_a.h$ pour la DCT directe et deux nouveaux tableaux $s_cos_msin_64[320][2]$ et $max_dither[640]$ sont ajoutés dans $dct4_s.h$ pour la DCT inverse, respectivement (voir $dct4_a.h$ et $dct4_s.h$ pour plus de précisions).

e) Doublement du nombre de sous-bandes de 500 Hz, qui passe de 14 à 28.

f) Doublement des tables de codage de Huffman comme suit:

les tables de codage de Huffman suivantes s'agrandissent par rapport à celles qui sont utilisées dans le corps du texte de la présente Recommandation.

$differential_region_power_bits[28][24]$

$differential_region_power_codes[28][24]$

differential_region_power_decoder_tree[28][23][2]
mlt_quant_centroid[8][16]

Les trois premières tables susmentionnées sont obtenues en reprenant la première ligne des tableaux correspondant de la présente Recommandation et la quatrième table est obtenue en ajoutant une nouvelle ligne et deux colonnes de zéros à la table de la présente Recommandation (voir *huff_tab.c* pour plus de précisions).

- g) Le nombre total de bits d'une trame est de 480, 640 ou 960 bits, pour les débits binaires de 24, 32 et 48 kbit/s, respectivement. Ce nombre comprend les bits servant à représenter l'enveloppe d'amplitude, les quatre bits de commande de catégorisation et les bits utilisés pour les coefficients MLT. L'affectation des bits s'effectue dynamiquement dans toute la bande de fréquences de 14 kHz, trame par trame.
- h) Doublement du seuil pour la détermination du nombre estimé de bits disponibles comme suit:

à partir du nombre réel de bits (*number_of_available_bits*), le programme calcule comme suit le nombre estimé de bits disponibles (*estimated_number_of_available_bits*):

si:

$$number_of_available_bits > 640,$$

alors:

$$estimated_number_of_available_bits = 640 + ((number_of_available_bits - 640) * 5/8)$$

La valeur *estimated_number_of_available_bits* est toujours inférieure au nombre réel de bits, ce qui laisse une marge de manœuvre dans le processus de catégorisation.

C.3 Code en langage C de l'ANSI

Le code en langage C conforme à l'ANSI qui implémente le mode à 14 kHz défini dans la présente annexe a été intégré avec la source C du mode à 7 kHz défini dans le corps du texte de la présente Recommandation. Les fichiers et l'utilisation, qui sont communs aux deux modes, sont exposés de façon détaillée au § 5.

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