

Groupe de travail Réseau  
**Request for Comments : 4207**  
 Catégorie : Sur la voie de la normalisation  
 Traduction Claude Brière de L'Isle

J. Lang, Sonos, Inc.  
 D. Papadimitriou, Alcatel

octobre 2005

# Codage de réseau optique synchrone (SONET)/hiérarchie numérique synchrone (SDH) pour les messages d'essai du protocole de gestion de liaisons (LMP)

## Statut de ce mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

## Notice de copyright

Copyright (C) The Internet Society (2005).

## Résumé

Le présent document détaille les informations spécifiques de la technologie de réseau optique synchrone (SONET, *Synchronous Optical Network*)/hiérarchie numérique synchrone (SDH, *Synchronous Digital Hierarchy*) nécessaires lors de l'envoi de messages d'essai du protocole de gestion de liaisons (LMP, *Link Management Protocol*).

## Table des matières

1. Introduction.....	1
2. Terminologie.....	2
3. Vérification de la connexité de liaison.....	2
3.1 Vérification du mécanisme de transport.....	3
4. Surveillance de trace.....	4
5. Considérations sur la sécurité.....	7
6. Considérations relatives à l'IANA.....	8
7. Références.....	8
7.1 Références normatives.....	8
7.2 Références pour information.....	8
8. Remerciements.....	8
Adresse des auteurs.....	9
Déclaration complète de droits de reproduction.....	9

## 1. Introduction

Pour des besoins d'adaptabilité, plusieurs ressources physiques qui interconnectent les routeurs de commutation d'étiquettes (LSR, *Label Switching Router*) peuvent être combinées pour former une seule liaison d'ingénierie du trafic (TE, *traffic engineering*) pour les besoins du calcul de chemin et de la signalisation. Ces ressources peuvent représenter une ou plusieurs liaisons physiques qui connectent les LSR, ou elles peuvent représenter un chemin de commutation d'étiquettes (LSP, *Label Switched Path*) si la hiérarchie de LSP [RFC4206] est utilisée. La gestion des liaisons TE ne se restreint pas à la messagerie dans la bande, mais peut plutôt être faite en utilisant des techniques hors bande.

Le protocole de gestion de liaisons (LMP, *Link Management Protocol*) [RFC4204] a été développé au titre de la suite de protocole MPLS généralisé (GMPLS) pour gérer les liaisons TE. LMP consiste actuellement en quatre procédures principales, dont les deux premières sont obligatoires et les deux dernières facultatives :

1. Gestion du canal de contrôle
2. Corrélation des propriétés de liaison
3. Vérification de liaison
4. Gestion des fautes

La gestion du canal de contrôle est utilisée pour établir et entretenir la connexité du canal de contrôle entre les nœuds adjacents. C'est fait en utilisant un échange de messages Config suivi par un échange de messages de garde en vie légers. La corrélation des propriétés de liaisons est utilisée pour agréger plusieurs liaisons de données en une seule liaison TE et pour synchroniser les propriétés des liaisons. La vérification de liaison est utilisée pour vérifier la connexité physique des liaisons de données et pour échanger les identifiants d'interface des liaisons de données. La gestion des fautes est principalement utilisée pour supprimer les alarmes et pour localiser les défaillances dans les réseaux aussi bien opaques que transparents. Quand LMP est utilisé avec SONET/SDH, les procédures de gestion de fautes peuvent cependant n'être pas nécessaires car les mécanismes existants de SONET/SDH peuvent être utilisés.

Dans le présent document, on définit les informations spécifiques de la technologie SONET/SDH pour LMP. Précisément, les procédures d'essai de SONET/SDH utilisées pour la vérification de liaison et la corrélation des propriétés de liaison sont détaillées. Ces procédures incluent le mécanisme de transport de corrélation de trace (défini pour J0, J1, J2) qui prend en charge une séparation des identifiants de plan de transport et de contrôle. Cette dernière procédure exige une nouvelle fonction de surveillance de trace qui est discutée dans le présent document. Une fois que les liaisons de données ont été vérifiées, elles peuvent être groupées pour former des liaisons TE.

## 2. Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Le lecteur est supposé familier avec la terminologie de la [RFC4204], [G.707], et de [T1.105]. Les abréviations suivantes sont utilisées ici :

CRC-N (*Cyclic Redundancy Check-N*) : contrôle de redondance cyclique N.

DCC (*Data communications channel*) : canal de communication de données.

LOVC (*Lower-order virtual container*) : conteneur virtuel d'ordre inférieur.

HOVC (*Higher-order virtual container*) : conteneur virtuel d'ordre supérieur.

MS (*Multiplex section*) : section de multiplexage.

MSOH (*Multiplex section overhead*) : frais généraux de section de multiplexage.

POH (*Path overhead*) : frais généraux de chemin.

RS (*Regenerator section*) : section de régénération.

RSOH (*Regenerator section overhead*) : frais généraux de section de régénération.

SDH (*Synchronous digital hierarchy*) : hiérarchie numérique synchrone.

SOH (*Section overhead*) : frais généraux de section.

SONET (*Synchronous Optical Network*) : réseau optique synchrone.

STM(-N) (*Synchronous Transport Module (-N)*) : module (-N) de transport synchrone (en SDH).

STS(-N) (*Synchronous Transport Signal-Level N*) : niveau de signal N de transport synchrone (en SONET).

VC-n (*Virtual Container-n*) : conteneur virtuel n (en SDH).

VTn (*Virtual Tributary-n*) : tributaire virtuel n (en SONET).

## 3. Vérification de la connexité de liaison

Dans la [RFC4204], une procédure de vérification de liaison est définie par le fait que des messages d'essai sont transmis dans la bande sur les liaisons de données. C'est utilisé pour la découverte du plan des données, l'échange des identifiants d'interface (les identifiants d'interface sont utilisés dans la signalisation GMPLS, soit comme étiquettes d'accès [RFC3471], soit comme identifiants de liaison composante [RFC4201], selon la configuration) et la vérification de la connexité physique. Plusieurs liaisons de données peuvent être vérifiées en utilisant une seule procédure de vérification ; la corrélation est faite en utilisant le "Verify\_Id" qui est alloué à la procédure.

Au titre de la procédure de vérification de liaison, un échange de message BeginVerify est utilisé pour s'accorder sur les paramètres de la procédure d'essai. Cela peut être initié par l'envoi d'un message BeginVerify sur le canal de contrôle. Ce message comporte un objet BEGIN\_VERIFY qui contient un certain nombre de champs spécifiant, entre autres choses, le taux de transmission (en bits) le type de codage, et un mécanisme de transport pour les messages Test. Si le nœud distant reçoit un message BeginVerify et si il est prêt à commencer la procédure, il envoie un message BeginVerifyAck qui spécifie le mécanisme de transport désiré pour les messages Test. Le nœud distant alloue aussi un Verify\_Id à la procédure et l'inclut dans le message BeginVerifyAck.

Le taux de transmission de la liaison de données sur laquelle les messages Test vont être transmis est représenté en format IEEE à virgule flottante en utilisant un champ numérique de 32 bits et est exprimé en octets par seconde. Voir dans la [RFC3471] les valeurs définies pour SONET/SDH.

Le type de codage identifie le codage pris en charge par une interface. Le codage défini est cohérent avec le type de codage LSP défini dans la [RFC3471]. Pour SONET/SDH, cette valeur doit être égale à la valeur donnée pour "SDH UIT-T G.707 / SONET ANSI T1.105".

Le mécanisme de transport est défini en utilisant le gabarit binaire de mécanisme de transport Verify. La portée de ce gabarit binaire est restreinte au type de codage de liaison. Plusieurs bits peuvent être établis quand ce champ est inclus dans le message BeginVerify ; cependant, seul un bit peut être établi quand il est inclus dans le message BeginVerifyAck.

Dans le paragraphe suivant, les diverses options pour le mécanisme de transport Verify sont définies quand le codage est SONET/SDH. Le mécanisme de transport de corrélation de trace (défini pour J0, J1, J2) prend en charge une séparation des identifiants de transport et de plan de contrôle.

### 3.1 Vérification du mécanisme de transport

Ce champ fait 16 bits.

Dans ce document, on définit les fanions pour le codage de SONET/SDH. Noter que toutes les valeurs sont définies dans l'ordre des octets du réseau (c'est-à-dire, l'ordre gros boutien).

0x0001 : Réservé

0x0002 DCCS : Message Test sur la Section/RS DCC

Capable de transmettre des messages Test en utilisant les octets de frais généraux DCC de la Section/RS avec le format de tramage en mode binaire de commande de liaison des données de haut niveau (HDLC, *High-Level Data Link Control*) [RFC1662]. Le message Test est envoyé comme défini dans la [RFC4204].

0x0004 DCCL : Message Test sur la ligne/MS DCC

Capable de transmettre des messages Test en utilisant les octets de frais généraux DCC de la ligne/MS avec le format de tramage en mode binaire HDLC [RFC1662]. Le message Test est envoyé comme défini dans la [RFC4204].

0x0008 J0-trace : Corrélation de trace de section J0

Capable de transmettre une trace de section/RS SONET/SDH sur l'octet de frais généraux de section/RS J0 comme défini dans [T1.105] et [G.707]. Le message Test n'est pas transmis en utilisant les octets J0 (c'est-à-dire, sur la liaison de données) mais est envoyé sur le canal de contrôle et corrélé pour la cohérence du schéma J0 reçu. Afin d'obtenir la transposition entre l'Interface\_Id sur laquelle le message Test J0 est envoyé et le schéma J0 envoyé dans la bande, le nœud émetteur doit fournir la corrélation entre ce schéma et le message Test J0. Cette corrélation est faite en utilisant l'objet TRACE comme défini dans la Section 4.

Le format du message Test est comme suit :

<message Test> ::= <En-tête commun> <Identifiant d'interfac locale> <VERIFY\_ID> <TRACE>

0x0010 : Réservé

0x0020 : Réservé

0x0040 J1-trace : Corrélation de trace de chemin J1

Capable de transmettre une trace de chemin SPE/HOVC STS SONET/SDH sur l'octet de frais généraux de chemin J1 comme défini dans [T1.105] et [G.707]. Le message Test n'est pas transmis en utilisant les octets J1 (c'est-à-dire, sur la liaison de données) mais est envoyé sur le canal de contrôle et corrélé pour sa cohérence avec le schéma J1 reçu. Afin d'obtenir la transposition entre l'Interface\_Id sur laquelle le message Test J1 est envoyé et le schéma J1 envoyé dans la bande, le nœud émetteur doit fournir la corrélation entre ce schéma et le message Test J1. Cette corrélation est faite en utilisant l'objet TRACE comme défini dans la Section 4. Le format du message Test est identique à celui défini ci-dessus dans J0-trace.

0x0080 J2-trace : Corrélation de trace de chemin J2

Capable de transmettre une trace de chemin SPE/LOVC VT SONET/SDH sur l'octet de frais généraux de chemin J2 comme défini dans [T1.105] et [G.707]. Le message Test n'est pas transmis en utilisant les octets J2 (c'est-à-dire, sur la

liaison de données) mais est envoyé sur le canal de contrôle et corrélé quant à sa cohérence au schéma J2 reçu. Afin d'obtenir la transposition entre l'identifiant de l'interface sur laquelle le message Test J2 est envoyé et le schéma J2 envoyé dans la bande, le nœud émetteur doit fournir la corrélation entre ce schéma et le message Test J2. Cette corrélation est faite en utilisant l'objet TRACE comme défini dans la Section 4.

Le format du message Test est identique à celui défini ci-dessus pour J0-trace.

## 4. Surveillance de trace

Les caractéristiques de surveillance de trace décrites dans cette Section permettent à un nœud de faire la surveillance de trace en utilisant les capacités SONET/SDH.

- o Un nœud peut demander à son voisin (le nœud distant) de surveiller une liaison pour un schéma spécifique dans le surdébit en utilisant le message TraceMonitor. Un exemple de ce surdébit est le message Trace de section SONET transmis dans l'octet J0. Si le message de trace réel ne correspond pas au message de trace attendu, le nœud distant DOIT rapporter la condition de discordance.
- o Un nœud peut demander la valeur du message de trace en cours sur une certaine liaison en utilisant le message TraceReq.
- o Un nœud peut demander à un nœud distant d'envoyer un message de trace spécifique sur une liaison de données en utilisant le message InsertTrace.

### 4.1.1 Message TraceMonitor

Le message TraceMonitor (type de message 21) est envoyé sur le canal de contrôle et est utilisé pour demander au nœud distant de surveiller une liaison de données pour une valeur de trace spécifique. Cette valeur est insérée dans l'objet <TRACE>. Le format du message TraceMonitor est le suivant :

```
<Message TraceMonitor> ::= <En-tête commun> <Identifiant de message> <LOCAL_INTERFACE_ID> <TRACE>
```

L'ordre de transmission ci-dessus DEVRAIT être suivi.

Le nœud distant DOIT répondre à un message TraceMonitor par un message TraceMonitorAck ou TraceMonitorNack.

#### 4.1.1.1 Classe d'objets TRACE

Classe = 21

C-Type = 1, Trace

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|N|  C-Type  |      Classe      |      Longueur      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type de trace      |      Longueur de trace      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
//                      Message Trace                      //
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Type de trace : 16 bits

Les valeurs suivante de type de message trace sont définies ; toutes les autres valeurs sont réservées.

- 1 = trace de section SONET (octet J0)
- 2 = trace de chemin SONET (octet J1)
- 3 = trace de chemin SONET (octet J2)
- 4 = trace de section SDH (octet J0)
- 5 = trace de chemin SDH (octet J1)
- 6 = trace de chemin SDH (octet J2)

Longueur de trace : 16 bits

C'est la longueur en octets du message trace (comme spécifié par le type de trace).

Message Trace : C'est la valeur du message qu'on s'attend à recevoir dans la bande. Les combinaisons valides de longueur et de valeur sont déterminées par la technologie utilisée : pour SONET voir [T1.105] et pour SDH, voir [G.707]. Le message DOIT être bourré de zéros jusqu'à une limite de 32 bits, si nécessaire. La longueur de trace n'inclut pas les zéros de bourrage.

Cet objet est non négociable.

#### 4.1.2 Message TraceMonitorAck

Le message TraceMonitorAck (type de message 22) est utilisé pour accuser réception du message TraceMonitor et indiquer que tous les objets TRACE qui étaient dans le message TraceMonitor ont été reçus et traités correctement (c'est-à-dire, pas de discordance de trace).

Le format est le suivant :

< Message TraceMonitorAck > ::= <En-tête commun> <MESSAGE\_ID\_ACK>

L'ordre de transmission ci-dessus DEVRAIT être suivi.

L'objet MESSAGE\_ID\_ACK est défini dans la [RFC4204]. Le contenu de l'objet MESSAGE\_ID\_ACK DOIT être obtenu du message TraceMonitor dont il accuse réception.

#### 4.1.3 Message TraceMonitorNack

Le message TraceMonitorNack (type de message 23) est utilisé pour accuser réception du message TraceMonitor et indiquer que l'objet TRACE dans le message TraceMonitor n'a pas été traité correctement. Cela peut être parce que la surveillance de trace demandée n'est pas prise en charge ou qu'il y avait une erreur dans la ou les valeurs d'objet TRACE.

Le format est le suivant :

<Message TraceMonitorNack > ::= <En-tête commun> <MESSAGE\_ID\_ACK> <Code d'erreur>

L'ordre de transmission suivant DEVRAIT être respecté.

Les objets MESSAGE\_ID\_ACK et Code d'erreur sont définis dans la [RFC4204]. Le contenu de l'objet MESSAGE\_ID\_ACK DOIT être obtenu du message TraceMonitor dont il accuse réception.

Si le type de trace n'est pas pris en charge, le code d'erreur DOIT indiquer "Type de trace non pris en charge" défini au paragraphe 4.1.3.1.

Si l'objet TRACE n'était pas égal à la valeur figurant dans la trace, le message TraceMonitorNack DOIT inclure le code d'erreur indiquant "Message Trace Invalide". Le message TraceMismatch (voir le paragraphe 4.1.4) NE DEVRAIT PAS être envoyé suite à une discordance.

Le message TraceMonitorNack utilise un nouveau C-Type de code d'erreur défini au paragraphe 4.1.3.1.

##### 4.1.3.1 Classe ERROR\_CODE

C-Type = 3, TRACE\_ERROR

Les nouvelles valeurs binaires de code d'erreur sont définies :

0x01 = Type de trace non pris en charge

0x02 = Message Trace invalide

Toutes les autres valeurs sont réservées. Plusieurs bits peuvent être établis pour indiquer des erreurs multiples. Cet objet est non négociable.

#### 4.1.4 Message TraceMismatch

Le message TraceMismatch (type de message 24) est envoyé sur le canal de contrôle et est utilisé pour rapporter une discordance de trace sur une liaison de données pour laquelle la surveillance de trace était demandée. Le format est le suivant :

```
<Message TraceMismatch> ::= <En-tête commun> <Identifiant de message> <Identifiant d'interface locale> [<Identifiant d'interface locale> ...]
```

L'ordre de transmission ci-dessus DEVRAIT être respecté.

Un nœud voisin qui reçoit un message TraceMismatch DOIT répondre par un message TraceMismatchAck.

L'objet Identifiant d'interface locale est défini dans la [RFC4204]. L'identifiant d'interface locale dans ce message est celui de la liaison de données qui a une discordance de trace. Une discordance de trace pour plusieurs identifiant d'interface locale peut être rapportée dans le même message.

#### 4.1.5 Message TraceMismatchAck

Le message TraceMismatchAck (type de message 25) est utilisé pour accuser réception d'un message TraceMismatch. Le format est le suivant :

```
<Message TraceMismatchAck> ::= <En-tête commun> <Accusé de réception d'identifiant de message>
```

L'objet Accusé de réception d'identifiant de message est défini dans la [RFC4204]. Le contenu de l'objet Accusé de réception d'identifiant de message DOIT être obtenu du message TraceMismatch dont il accuse réception.

#### 4.1.6 Message TraceReq

Le message TraceReq (type de message 26) est envoyé sur le canal de contrôle et est utilisé pour demander la valeur de trace actuelle d'une liaison de données.

```
<Message TraceReq> ::= <En-tête commun> <Identifiant de message> <Identifiant d'interface locale> <TRACE_REQ>
```

L'ordre de transmission ci-dessus DEVRAIT être respecté.

Le format de l'objet TRACE\_REQ est le suivant :

Classe = 22

C-Type = 1, TraceReq

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
N  C-Type										Classe										Longueur											
Type de trace																(Réservé)															

Type de trace : 16 bits. Défini au paragraphe 4.1.1.1.

Réservé : 16 bits. Ce champ DOIT être réglé à zéro à l'émission et ignoré à réception.

#### 4.1.7 Message TraceReport

Le message TraceReport (type de message 27) est envoyé sur le canal de contrôle après réception d'un message TraceReq.

```
<Message TraceReport> ::= <En-tête commun> <MESSAGE_ID_ACK> <TRACE>
```

Le message TraceReport DOIT inclure un objet TRACE (comme décrit au paragraphe 4.1.1.1) pour la liaison de données demandée.

L'objet MESSAGE\_ID\_ACK est défini dans la [RFC4204]. Le contenu de l'objet MESSAGE\_ID\_ACK DOIT être obtenu du message TraceReq dont on accuse réception.

#### 4.1.8 Message TraceReqNack

Le message TraceReqNack (type de message 28) est envoyé sur le canal de contrôle après réception d'un message TraceReq.

<Message TraceReqNack> ::= <En-tête commun> <MESSAGE\_ID\_ACK> <Code d'erreur>

L'ordre de transmission ci-dessus DEVRAIT être respecté.

L'objet MESSAGE\_ID\_ACK est défini dans la [RFC4204]. Le contenu de l'objet MESSAGE\_ID\_ACK DOIT être obtenu du message TraceReq dont il accuse réception.

Le message TraceReqNack DOIT inclure un objet Code d'erreur (comme défini au paragraphe 4.1.3.1) pour la liaison de données demandée.

#### 4.1.9 Message InsertTrace

Le message InsertTrace (type de message 29) est envoyé sur le canal de contrôle et est utilisé pour demander à un nœud distant d'envoyer un message de trace spécifique sur une liaison de données (cela suppose que le nœud distant sache la transposition entre les identifiants d'interface locale et distant avant de satisfaire une telle requête).

Le format est le suivant :

<Message InsertTrace> ::= <En-tête commun> <Identifiant de message> <Identifiant d'interface locale> <TRACE>

L'ordre de transmission ci-dessus DEVRAIT être respecté.

Un nœud qui reçoit un message InsertTrace DOIT répondre avec un message InsertTraceAck ou InsertTraceNack.

Une fois le message InsertTraceAck reçu, le message TraceMismatch (voir au paragraphe 4.1.4) est utilisé pour indiquer qu'une discordance de trace s'est produite.

L'objet Identifiant de message est défini dans la [RFC4204].

#### 4.1.10 Message InsertTraceAck

Le message InsertTraceAck (type de message 30) est utilisé pour accuser réception du message InsertTrace et indiquer que l'objet TRACE dans le message InsertTrace a bien été reçu et traité correctement (c'est-à-dire, pas de discordance de trace). Le format est le suivant :

<Message InsertTraceAck> ::= <En-tête commun> <Accusé de réception d'identifiant de message>

L'objet Accusé de réception d'identifiant de message est défini dans la [RFC4204]. Le contenu de l'objet Accusé de réception d'identifiant de message DOIT être obtenu du message InsertTrace dont on accuse réception.

#### 4.1.11 Message InsertTraceNack

Le message InsertTraceNack (type de message 31) est utilisé pour accuser réception du message InsertTrace et pour indiquer que l'objet TRACE dans le message InsertTrace n'a pas été traité correctement. Cela pourrait être parce que la surveillance de trace demandée n'est pas prise en charge ou qu'il y avait une erreur dans la valeur.

Le format est le suivant :

<Message InsertTraceNack> ::= <En-tête commun> <Accusé de réception d'identifiant de message> <Code d'erreur>

L'ordre de transmission ci-dessus DEVRAIT être respecté.

L'objet Accusé de réception d'identifiant de message est défini dans la [RFC4204].

Le message InsertTraceNack DOIT inclure un objet Code d'erreur (comme défini au paragraphe 4.1.3.1) pour la liaison de données demandée.

## 5. Considérations sur la sécurité

La sécurité de message LMP utilise IPsec comme décrit dans la [RFC4204]. Le présent document n'introduit aucune autre nouvelles considérations sur la sécurité qui ne soit couverte dans la [RFC4204].

## 6. Considérations relatives à l'IANA

LMP [RFC4204] définit les espaces de noms suivants et la façon dont l'IANA effectue les allocations dans ces espaces de noms :

- type de message LMP.
- classe d'objet LMP.
- type de classe d'objet LMP (C-Type) unique au sein de la classe d'objet.
- type de classe de sous objet LMP (Type) unique au sein de la classe d'objet.

Le présent mémoire introduit les nouvelles allocations suivantes :

Type de message LMP :

- o Message TraceMonitor (type de message = 21)
- o Message TraceMonitorAck (type de message = 22)
- o Message TraceMonitorNack (type de message = 23)
- o Message TraceMismatch (type de message = 24)
- o Message TraceMismatchAck (type de message = 25)
- o Message TraceReq (type de message = 26)
- o Message TraceReport (type de message = 27)
- o Message TraceReqNack (type de message = 28)
- o Message InsertTrace (type de message = 29)
- o Message InsertTraceAck (type de message = 30)
- o Message InsertTraceNack (type de message = 31)

Espace de noms de classe d'objet LMP et de type de classe (C-Type) :

- o TRACE Nom de classe (21)
  - Type 1 (C-Type = 1)
- o TRACE REQ Nom de classe (22)
  - Type 1 (C-Type = 1)

## 7. Références

### 7.1 Références normatives

- [G.707] Recommandation UIT-T G.707, "Interface de nœud de réseau pour la hiérarchie numérique synchrone (SDH)," octobre 2000.
- [RFC1662] W. Simpson, éditeur, "[PPP en trames de style HDLC](#)", STD 51, juillet 1994. (*Remplace la RFC1549*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (*MàJ par RFC8174*)
- [RFC3471] L. Berger, éd., "[Commutation d'étiquettes multi-protocoles généralisée](#) (GMPLS) : description fonctionnelle de la signalisation", janvier 2003. (*MàJ par RFC4201, RFC4328, RFC4872, RFC8359*) (*P.S.*)
- [RFC4201] K. Kompella et autres, "[Faisceaux de liaisons](#) dans l'ingénierie du trafic MPLS", octobre 2005. (*P.S.*)
- [RFC4204] J. Lang, éd., "[Protocole de gestion de liaison](#) (LMP)", octobre 2005. (*P.S.*)

[T1.105] Recommandation UIT-T T1.105, "Revised Draft T105 SONET Base Standard," January 2001.

## 7.2 Références pour information

[RFC4206] K. Kompella, Y. Rekhter, "[Hiérarchie de chemins commutés par étiquettes](#) (LSP) avec l'ingénierie de trafic (TE) de la commutation généralisée d'étiquettes multi-protocoles (GMPLS)", octobre 2005. (P.S.)

## 8. Remerciements

Les auteurs tiennent à remercier Bernard Sales, Emmanuel Desmet, Gert Grammel, Jim Jones, Stefan Ansorge, John Drake, et James Scott de leurs nombreuses contributions au présent document.

Merci également à Greg Bernstein et Michiel van Everdingen de leurs précieux commentaires et pour leur action combinant fortement la fermeté, le professionnalisme, et la courtoisie.

## Adresse des auteurs

Jonathan P. Lang  
Sonos, Inc.  
223 E. De La Guerra St.  
Santa Barbara, CA 93101  
USA  
mél : [jplang@ieee.org](mailto:jplang@ieee.org)

Dimitri Papadimitriou  
Alcatel  
Francis Wellesplein 1  
B-2018 Antwerpen,  
Belgium  
mél : [dimitri.papadimitriou@alcatel.be](mailto:dimitri.papadimitriou@alcatel.be)

## Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2005).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à [www.rfc-editor.org](http://www.rfc-editor.org), et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

### Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

### Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par la Internet Society.