

Groupe de travail Réseau  
**Request for Comments** : 3321  
 Catégorie : Information

H. Hannu & J. Christoffersson, Ericsson  
 S. Forsgren & K.-C. Leung, Texas Tech University  
 Z. Liu, Nokia  
 R. Price, Siemens/Roke Manor  
 mai 2003

Traduction Claude Brière de L'Isle

## Compression de signalisation (SigComp) – Fonctionnement étendu

### Statut de ce mémoire

Le présent mémoire apporte des informations pour la communauté de l'Internet. Le présent mémoire ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de copyright

Copyright (C) The Internet Society (2003).

### Résumé

Le présent document décrit comment mettre en œuvre certains mécanismes de compression de signalisation (SigComp, *Signaling Compression*) (RFC 3320) qui peuvent significativement améliorer l'efficacité de compression comparée à l'utilisation de la simple compression par message.

SigComp utilise une machine virtuelle de décompresseur universel (UDVM, *Universal Decompressor Virtual Machine*) pour la décompression, et les mécanismes décrits dans ce document peuvent être mis en œuvre en utilisant les instructions pour l'UDVM définies dans la RFC 3320.

### Table des Matières

1. Introduction.....	1
2. Terminologie.....	2
3. Vue de l'architecture des retours.....	2
4. Modèle de référence d'état.....	3
4.1 Généralités sur la référence d'état avec compression dynamique.....	3
5. Mécanismes étendus.....	4
5.1 Schéma d'accusé de réception explicite.....	4
5.2 Compression partagée.....	5
5.3 Conservation des données d'état à travers les sessions d'application.....	5
5.4 Utilisation d'un dictionnaire spécifique de l'utilisateur.....	6
5.5 État au point de vérification.....	6
5.6. Suppression implicite pour mise à jour de dictionnaire.....	6
6. Implications sur SigComp.....	7
6.1 Implications sur les messages SigComp.....	7
6.2 Extension du format d'annonce/retour SigComp.....	8
6.3 Optimisation d'accusé de réception.....	8
7. Considérations sur la sécurité.....	9
8. Considérations relatives à l'IANA9.....	9
9. Remerciements.....	9
10. Considérations de droits de propriété intellectuelle.....	9
11. Références.....	9
12. Adresse des auteurs.....	9
13. Déclaration complète de droits de reproduction.....	10

## 1. Introduction

Le présent document décrit comment mettre en œuvre les mécanismes de la [RFC3320] pour améliorer significativement l'efficacité de la compression par rapport à celle par message.

Un de ces mécanismes est d'utiliser les messages envoyés précédemment dans le processus de compression SigComp, auquel on se réfère sous le nom de compression dynamique. Afin d'utiliser les informations provenant des messages envoyés précédemment, il est nécessaire qu'un compresseur obtienne la connaissance de la réception de ces messages. Pour



La fonction de retours de SigComp est utilisée dans ce document pour fournir un mécanisme pour qu'un point d'extrémité SigComp confirme quels états ont été établis par son point d'extrémité SigComp distant durant la vie d'un compartiment SigComp. Les confirmations d'état établi sont appelées des accusés de réception. Selon l'état établi, ce type particulier de retour peut être ou non utilisé pour augmenter l'efficacité de la compression.

Les sections qui suivent décrivent comment la fonction SigComp de fourniture d'informations de retours est utilisée pour prendre en charge les mécanismes décrits dans ce document. La Section 4 décrit le modèle de référence d'état de SigComp. La Section 5 continue avec une description générale du mécanisme et la Section 6 décrit les implications de certains des mécanismes sur le SigComp de base.

## 4. Modèle de référence d'état

Une UDVM peut vouloir sauvegarder l'état de sa mémoire, et cet état est appelé un état. Comme expliqué dans la [RFC3320] une demande de sauvegarde d'état peut ou non être accordée par l'application. Pour se référer ultérieurement à un état sauvegardé, par exemple, si l'UDVM doit être chargée avec cet état, une référence est nécessaire pour localiser l'état spécifique. Cette référence est appelée un identifiant d'état.

### 4.1 Généralités sur la référence d'état avec compression dynamique

Lorsque le compresseur 1 compresse un message  $m$ , il utilise les informations correspondantes à un état SigComp que son décompresseur 2 distant a établi et dont il a accusé réception. Si le compresseur 1 souhaite utiliser le nouvel état pour la compression de messages ultérieurs, il doit sauvegarder le nouvel état. Le nouvel état contient des informations provenant de l'état antérieur et de  $m$ . Lorsque un accusé de réception est reçu pour ce nouvel état, le compresseur 1 peut utiliser le nouvel état dans le processus de compression. Ci-dessous figure une vue d'ensemble du modèle ainsi qu'un exemple de flux de messages.

États sauvegardés : états dont on s'attend à ce qu'ils soient utilisés pour la compression/décompression de messages ultérieurs.

États acquittés : un état acquitté est un état sauvegardé pour lequel le compresseur a reçu un accusé de réception, c'est-à-dire que l'état a été établi chez le décompresseur distant. Le compresseur ne doit utiliser que les états qui ont été établis chez le décompresseur distant, autrement un échec de décompression va se produire. Pour cette raison, les accusés de réception sont nécessaires, au moins pour un transport non fiable.

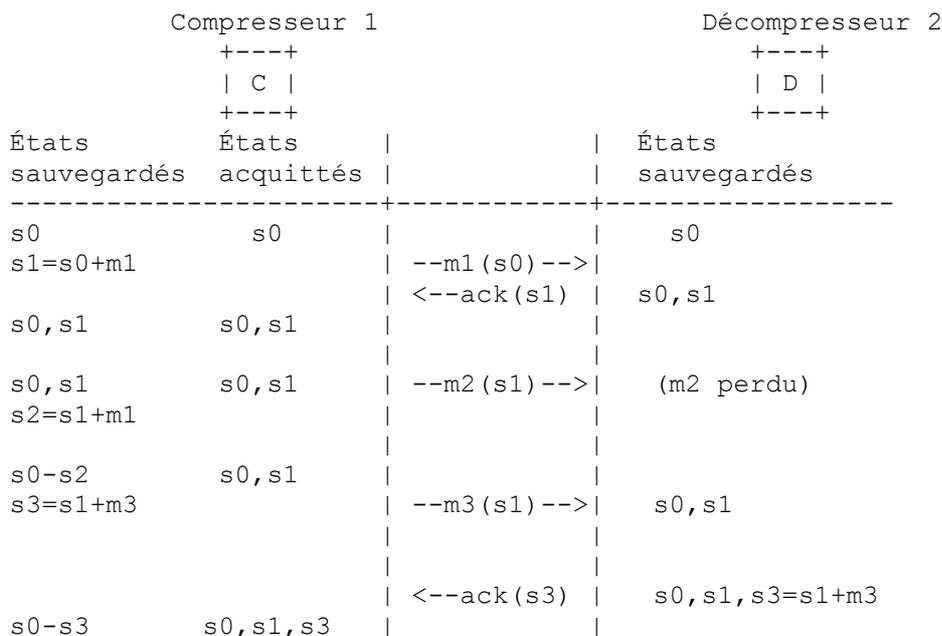


Figure 2 : Exemple de flux de message pour compression dynamique

Légende : le message 1 compressé en utilisant l'état  $s_0$  est noté  $m_1(s_0)$ . La notation  $s_1=s_0+m_1$  signifie que l'état  $s_1$  est créé en utilisant les informations provenant de l'état  $s_0$  et du message  $m_1$ .  $ack(s_1)$  signifie que l'accusé de réception

de la création de l'état s1 est porté sur un message qui voyage dans la direction inverse (qui n'est pas montrée sur la figure).

## 5. Mécanismes étendus

Les paragraphes qui suivent donnent une description générale des mécanismes étendus.

### 5.1 Schéma d'accusé de réception explicite

Pour qu'un compresseur soit capable d'utiliser un certain état, il doit savoir que le décompresseur distant a accès à cet état.

Dans le cas où le message compressé pourrait être perdu ou déclassé sur le chemin entre compresseur et décompresseur, un schéma d'accusé de réception doit être utilisé pour notifier au compresseur distant qu'un certain état a été établi.

Des accusés de réception explicites peuvent être initiés soit par du code UDVM téléchargé au décompresseur par le compresseur distant, soit par le point d'extrémité où les états ont été établis. Les deux paragraphes qui suivent expliquent plus en détails ces deux cas.

#### 5.1.1 Accusés de réception à l'initiative du compresseur distant

C'est le cas lorsque, par exemple, le compresseur 1 a téléchargé le code d'octet d'UDVM sur le décompresseur 2. Le code d'octet d'UDVM va utiliser le champ de retours demandé dans les informations d'annonce et le champ de retours de l'en-tête SigComp retourné pour obtenir les connaissances des états établis au point d'extrémité 2.

Considérons la Figure 3. Un flux d'accusés de réception d'événements d'utilisation réussie initiés par le compresseur distant peut être vu comme suit :

- (1) Le compresseur 1 sauvegarde, par exemple, l'état(A).
- (2) Le code d'octet d'UDVM pour initier une sauvegarde d'état pour l'état (A) est soit porté dans le message compressé, soit peut être restitué par le décompresseur 2 à partir d'un état déjà sauvegardé au point d'extrémité 2.
- (3) Comme le compresseur 1 est l'initiateur de cet accusé de réception, il peut utiliser un identifiant arbitraire à retourner pour indiquer que cet état (A) a été établi. L'identifiant doit contenir suffisamment de bits pour éviter d'accuser réception du mauvais état. Pour éviter d'avoir à appliquer un bourrage aux éléments de retours et pour un maximum de simplicité, un minimum de 1 octet devrait être utilisé pour l'identifiant. L'identifiant est placé à la localisation de `requested_feedback_item` [RFC3320]. L'instruction END-MESSAGE est utilisée pour indiquer la localisation du `requested_feedback_item` au gardien d'état.
- (4) Les données de retour demandées sont maintenant appelées données de retour retournées lorsque elles sont placées dans le message SigComp au compresseur 2.
- (5) L'élément de retour retourné est porté dans le message SigComp conformément à la Figure 4 : voir le paragraphe 6.1 et la [RFC3320].
- (6) L'élément de retour retourné est traité conformément à la Section 7 de la [RFC3320]

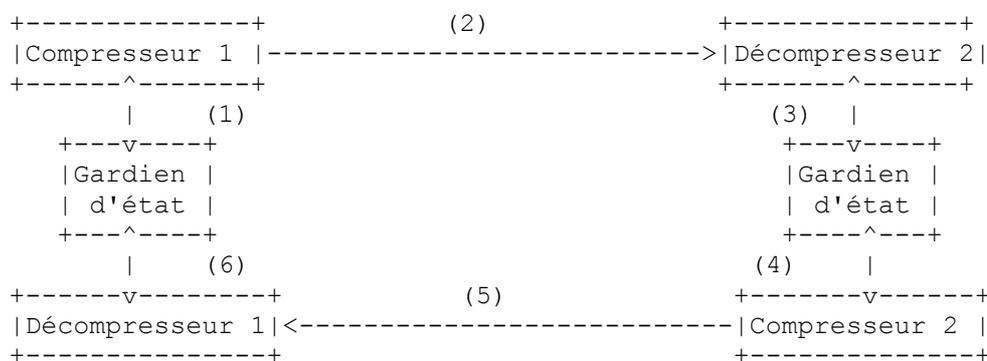


Figure 3 : Points d'extrémité SigComp simplifiés

#### 5.1.2 Accusés de réception à l'initiative du point d'extrémité local

Lorsque des accusés de réception explicites sont fournis par un point d'extrémité, le message SigComp va aussi porter les accusés de réception, appelés "`acked_state_id`" (voir la Section 2). À la Figure 3, un flux d'événements pour l'utilisation

réussie d'accusés de réception explicites initiés par le point d'extrémité peut être vu comme suit :

- (1) Le compresseur 1 sauvegarde, par exemple, l'état (A).
- (2) Le codage binaire d'UDVM pour initier une sauvegarde d'état pour l'état (A) est soit porté dans le message compressé, soit peut être restitué par le décompresseur 2 à partir d'un état déjà sauvegardé au point d'extrémité 2.
- (3) Une demande de sauvegarde d'état pour l'état (A) est passée au gardien d'état en utilisant l'instruction END-MESSAGE. L'application peut alors accorder au gardien d'état la permission de sauvegarder l'état (A) (voir la [RFC3320]).
- (4) Le point d'extrémité 2 décide d'accuser réception de l'état (A) au point d'extrémité 1. L'identifiant d'état (acked\_state\_id) pour l'état (A) est placé dans le message SigComp envoyé du compresseur 2 au décompresseur 1.
- (5) Le codage binaire d'UDVM pour initier (passer) l'accusé de réception explicite au point d'extrémité 1 est soit porté dans le message compressé, soit peut être restitué par le décompresseur 1 à partir d'un état déjà sauvegardé au point 1.
- (6) Le acked\_state\_id pour l'état (A) est passé au gardien d'état en plaçant le acked\_state\_id à la localisation des "paramètres SigComp retournés" [RFC3320], dont la localisation est donnée au gardien d'état en utilisant l'instruction END-MESSAGE.

Note : Lorsque la longueur du retour demandé n'est pas zéro, les accusés de réception initiés par le point d'extrémité ne devraient pas être utilisés, à cause d'un possible gaspillage de bande passante. Lorsque on décide de mettre ce mécanisme en œuvre, on devrait examiner si cela en vaut la peine, car toutes les mises en œuvre de SigComp vont prendre en charge le mécanisme de retour et ont donc la possibilité de mettre en œuvre le mécanisme du paragraphe 5.1.1.

## 5.2 Compression partagée

Pour utiliser la compression partagée un point d'extrémité qui compresse sauvegarde la version non compressée du message compressé comme un état (état partagé). Comme décrit à la Section 2, la référence à un état partagé est appelée *shared\_state\_id* (identifiant d'état partagé). Les paramètres d'état partagé *state\_address* et *state\_instruction* doivent être réglés à zéro. La *state\_retention\_priority* (priorité de rétention d'état) doit être réglée à 65535, et les autres paramètres d'état sont réglés conformément à la [RFC3320]. Cela parce que différents algorithmes de compression peuvent être utilisés pour compresser les messages d'application qui voyagent dans les différentes directions. L'état partagé est aussi créé sur la base du compartiment, c'est-à-dire que l'état partagé est mémorisé dans la même mémoire que les états créés par le compresseur distant particulier. Le choix de la façon de diviser la mémoire d'état entre les états "ordinaires" et les états partagés est une décision de mise en œuvre chez le compresseur. Noter que de nouveaux éléments d'état partagé ne doivent pas être créés sauf si le compresseur a assez de mémoire d'état disponible (car un échec de décompression pourrait se produire si l'état partagé poussait l'état existant hors de la mémoire tampon d'état).

Un point d'extrémité qui compresse doit aussi indiquer au compresseur distant que l'état partagé est disponible, mais seulement si le décompresser local peut restituer l'état partagé. La restitution de l'état partagé est faite conformément aux instructions de restitution d'état de l'UDVM.

Considérons la Figure 3. Un flux d'événements pour l'utilisation réussie de la compression partagée peut être vu comme suit :

- (1) Le compresseur 1 sauvegarde, par exemple, l'état (M), qui est la version non compressée du message d'application en cours à compresser et envoyer.
- (2) Le code d'octet d'UDVM pour indiquer la présence de l'état (M) au point d'extrémité 1 est soit porté dans le message compressé, soit peut être restitué par le décompresseur 2 à partir d'un état déjà sauvegardé au point d'extrémité 2.
- (3) L'instruction SHA-1 est utilisée au point d'extrémité 2 pour calculer le *shared\_state\_id* pour l'état (M). L'indication est passée au gardien d'état, en plaçant l'identifiant partagé à la localisation de "paramètres SigComp retournés" [RFC3320]. La localisation de "paramètres SigComp retournés" est donnée au gardien d'état en utilisant l'instruction END-MESSAGE.
- (4) Si le point d'extrémité 2 utilise la compression partagée, il compare les valeurs d'identifiant d'état dans les informations de "paramètres SigComp retournés" avec la valeur qu'il a calculé pour le message décompressé en cours reçu du point d'extrémité 1. Si il y a correspondance, le point d'extrémité 2 utilise alors l'état partagé avec l'état qu'il aurait normalement utilisé si la compression partagée n'avait pas été acceptée pour compresser le prochain message.
- (5) Le code d'octet d'UDVM qui va utiliser l'état partagé (état (M)) dans le processus de décompression au décompresseur 1 est soit porté dans le message compressé, soit peut être restitué par le décompresseur 1 à partir d'un état déjà sauvegardé au point d'extrémité 1.

## 5.3 Conservation des données d'état à travers les sessions d'application

Généralement, les protocoles de signalisation (par exemple, SIP) emploient le concept de session. Cependant, du point de vue de la compression, les messages envoyés par la même source contiennent des redondances qui vont au delà des limites d'une session. Par conséquent, il est naturel de conserver les données d'état provenant de la même source à travers les

sessions de sorte que de bonnes performances puissent être réalisées et conservées, la surcharge étant amortie sur une plus longue période que celle d'une session d'application.

Conserver les états à travers les sessions d'application peut être fait simplement en rendant la durée de vie d'un compartiment plus longue que la durée d'une seule session d'application. Noter qu'ici les états auxquels on se réfère sont ceux qui sont mémorisés sur la base du compartiment, et non les états disponibles localement qui sont mémorisés de façon globale (c'est-à-dire pas spécifiques d'un compartiment).

#### 5.4 Utilisation d'un dictionnaire spécifique de l'utilisateur

Le concept de dictionnaire spécifique de l'utilisateur se fonde sur l'observation que pour des protocoles comme SIP, une certaine combinaison utilisateur/appareil va produire des messages contenant des champs qui sont toujours remplis des mêmes données.

Prenons SIP en exemple. Les capacités des points d'extrémité SIP sont communiquées durant l'initialisation de session, et ne tendent pas à changer sauf si les capacités de l'appareil changent. De même, des informations spécifiques de l'utilisateur, comme son URL, son nom, et son adresse de messagerie vont probablement ne pas changer fréquemment, et vont apparaître régulièrement dans les échanges de signalisation SIP qui impliquent un utilisateur spécifique.

Donc, un compresseur SigComp pourrait inclure le dictionnaire spécifique de l'utilisateur au titre des messages initiaux au décompresseur, même avant que soient générés de messages de signalisation sensibles au temps provenant d'une application particulière. Cela permet une augmentation de l'efficacité de compression une fois que les messages commencent à s'écouler.

Évidemment, le dictionnaire spécifique de l'utilisateur est un élément d'état qu'il serait bon d'avoir dans un état persistant à travers les sessions (voir au paragraphe 5.3).

#### 5.5 État au point de vérification

Le mécanisme suivant peut être utilisé pour éviter un échec de décompression dû à une référence à un état non existant. Cela peut arriver dans trois cas : a) un état n'est pas établi au point d'extrémité SigComp distant à cause de la perte d'un message SigComp ; b) un état n'est pas établi à cause d'une mémoire insuffisante ; c) un état a été établi mais a été supprimé ultérieurement à cause d'une insuffisance de mémoire.

Lorsque un compresseur envoie un message SigComp qui va créer un nouvel état sur le côté décompresseur, il peut indiquer que l'état nouvellement créé sera un état de point de vérification en réglant `state_retention_priority` [RFC3320] à la plus haute valeur envoyée par le même compresseur. De plus, un état de point de vérification doit être explicitement acquitté par le décompresseur receveur au compresseur envoyeur.

Considérons la Figure 3. Un flux d'événement pour cette sorte de gestion d'état peut être vu comme suit :

- (1) Le compresseur 1 sauvegarde, par exemple, l'état (A), qu'il aimerait avoir comme état de point de vérification au décompresseur 2.
- (2) Le code d'octet d'UDVM pour indiquer la priorité d'état (`state_retention_priority` dans la [RFC3320]) de l'état (A) et initier une sauvegarde d'état pour l'état(A) est soit portée dans le message compressé, soit peut être restitué par le décompresseur 2 à partir d'un état déjà sauvegardé au point d'extrémité 2.
- (3) Une demande de sauvegarde d'état pour l'état (A) est passée au gardien d'état en utilisant l'instruction END-MESSAGE, incluant l'indication de la priorité d'état. L'application accorde la sauvegarde de l'état (A) (voir la [RFC3320]).
- (4) Un accusé de réception pour l'état (A) (l'état de point de vérification) est retourné au point d'extrémité 2 en utilisant un des mécanismes décrits au paragraphe 5.1.

Note : Pour éviter d'utiliser un état qui a été supprimé à cause d'une mémoire insuffisante, un compresseur doit garder trace de la mémoire disponible pour sauvegarder les états au point d'extrémité distant. Le paramètre SigComp `state_memory_size` (*taille de mémoire d'état*) qui est annoncé par le mécanisme de retours SigComp peut être utilisé pour déduire si un état de point de vérification précédent a été supprimé (par la demande ultérieure de création d'un état de point de vérification) à cause d'un manque de mémoire.

#### 5.6 Suppression implicite pour mise à jour de dictionnaire

Généralement, un état consiste en deux parties : un code d'octet d'UDVM et un dictionnaire. Lorsque la compression dynamique est appliquée, un nouveau contenu doit être ajouté au dictionnaire. Pour garder une limite supérieure à la consommation de mémoire comme dans le cas d'un terminal mobile à l'extrémité distante, le contenu existant du

dictionnaire doit être supprimé pour faire de la place au nouveau contenu.

Au lieu de signaler explicitement quelles parties du dictionnaire doivent être supprimées message par message, une approche de suppression implicite peut être appliquée. Précisément, certaines parties du dictionnaire sont choisies pour être supprimées conformément à un algorithme bien défini qui est connu et appliqué de la même façon au compresseur et au décompresseur. Par exemple, l'algorithme peut faire partie du code d'octet d'UDVM prédéfini qui a fait l'objet d'un accord entre les deux points d'extrémité SigComp. Comme entrée à l'algorithme, on fournit le nombre total d'octets à supprimer. L'algorithme spécifie alors quelles parties du dictionnaire sont à supprimer. Comme le même algorithme est appliqué aux deux points d'extrémité SigComp, il n'y a pas besoin d'une signalisation explicite message par message. Cela peut conduire à une meilleure efficacité de compression due à l'évitement des frais de signalisation. Cela signifie aussi plus de robustesse car il n'y a pas de bits de signalisation soumis à de possibles erreurs/pertes de transmission dans le réseau.

## 6. Implications sur SigComp

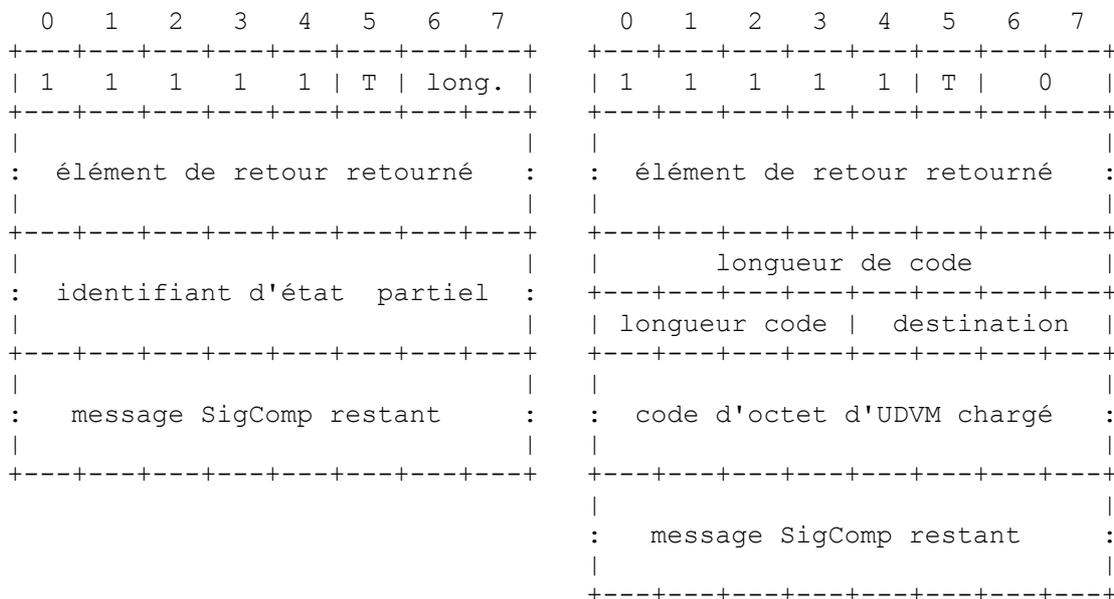
Les caractéristiques étendues auront des implications sur les messages SigComp envoyés entre le compresseur et son décompresseur distant, et sur la façon d'interpréter, par exemple, les paramètres SigComp retournés [RFC3320]. Cependant, excepté pour les octets obligatoires des messages SigComp [RFC3320], les formats de message final utilisés sont une affaire de mise en œuvre. Noter qu'une mise en œuvre qui n'utilise pas les accusés de réception explicites et/ou la compression partagée n'est pas affectée, même si elle reçoit ce type de retours.

### 6.1 Implications sur les messages SigComp

Pour prendre en charge les caractéristiques étendues, les messages SigComp doivent porter les indications et informations traitées dans la Section 5. Par exemple pour prendre en charge la compression partagée et les accusés de réception explicites, les messages SigComp doivent porter les informations suivantes :

- Le `acked_state_id` comme décrit à la Section 2 et au paragraphe 5.1.
- Le `shared_state_id` comme décrit à la Section 2 et au paragraphe 5.2.

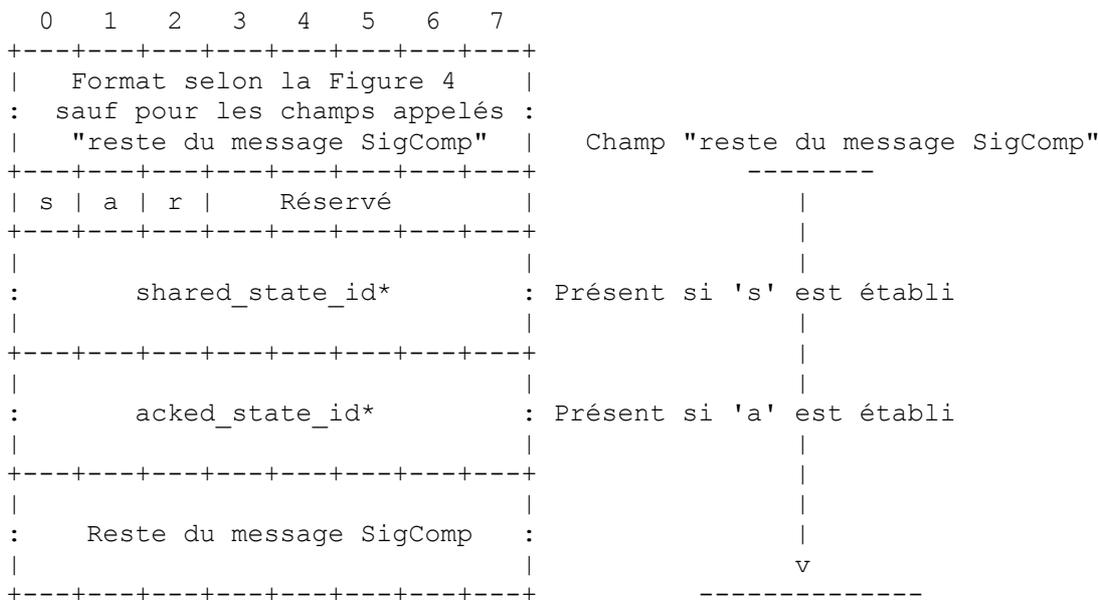
La Figure 4 décrit le format d'un message SigComp conformément à la [RFC3320] :



**Figure 4 : Format d'un message SigComp**

Le format du champ "reste du message SigComp" est une décision de mise en œuvre du compresseur qui fournit le code d'octet d'UDVM. Il n'est donc pas besoin de spécifier un format de message pour porter les informations nécessaires pour les caractéristiques étendues décrites dans ce document.

La Figure 5 décrit un exemple de ce à quoi pourrait ressembler le "reste du message SigComp" prenant en charge la compression partagée et les accusés de réception explicites. Noter que ceci n'est qu'un exemple ; le format est une décision de mise en œuvre.



**Figure 5 : Exemple de message SigComp pour certaines des extension de caractéristiques**

'r' : s'il est établi (à 1), un état correspondant à la version décompressée de ce message compressé (état partagé) a été sauvegardé au compresseur.

\* : la longueur des champs `shared_state_id` et `acked_state_id` est la même que celle de l'identifiant d'état partiel.

## 6.2 Extension du format d'annonce/retour SigComp

Ce paragraphe décrit comment les informations de "returned\_SigComp\_parameters" [RFC3320] sont interprétées pour fournir des retours conformément aux paragraphes 5.1 et 5.2.

L'identifiant d'état partiel correspond à la valeur du hachage pour les états qui ont été établis au point d'extrémité distant après la réception des messages SigComp, c'est-à-dire, ce sont des accusés de réception pour les états établis et qui peuvent être utilisés pour la compression. Les `partial_state_identifiers` peuvent aussi annoncer un "état global" qui n'est pas transposé en un compartiment particulier et n'est pas établi à réception d'un message SigComp.

Il appartient à la mise en œuvre de déduire à quel sorte d'état chaque `partial_state_identfier` se réfère, par exemple, un état acquitté ou un état partagé. Au cas où un message SigComp qui inclurait des identifiants d'état pour des états partagés et/ou acquittés serait reçu par une mise en œuvre SigComp de base, ces identifiants seront ignorés.

Le bit I du format de retour exigé est fourni comme commutateur de la liste des éléments d'état localement disponibles. Un point d'extrémité qui souhaite recevoir un `shared_state_id` ne doit pas établir le bit I à 1. Le point d'extrémité qui mémorise les états partagés et qui envoie la liste des états localement disponibles à son point d'extrémité distant doit faire attention quand il prend la décision d'exclure ou inclure des types différents d'états disponibles localement (c'est-à-dire des états partagés ou des états, par exemple, d'algorithmes bien connus) de la liste.

## 6.3 Optimisation d'accusé de réception

Si la compression partagée est utilisée entre deux points d'extrémité (voir la Figure 1) il existe alors une optimisation, qui, si elle est mise en œuvre, rend inutile un `acked_state_id` dans le message SigComp :

Le compresseur 1 sauvegarde un état partagé (M), qui est la version non compressée du message actuellement compressé (le message m) à envoyer. Le compresseur 1 établit aussi le bit 'r' (voir la Figure 5) pour signaler que l'état (M) peut être utilisé par le point d'extrémité 2 dans le processus de compression. Le `acked_state_id` pour l'état (S), qui a été créé au point d'extrémité 2 au moment de la décompression du message m, peut n'avoir pas été explicitement placé dans les messages compressés provenant du compresseur 2 si l'état partagé (M) est utilisé dans le processus de compression.

Lorsque le point d'extrémité 1 remarque que l'état partagé (M) est demandé par le décompresseur 1, il sait implicitement que l'état (S) a été créé au point d'extrémité 2. Il s'ensuit donc que :

- \* le compresseur 1 a donné pour instruction au décompresseur 2 de sauvegarder l'état (S) ;
- \* l'indication d'état partagé (M) n'aurait jamais été reçue par le compresseur 2 si l'état (S) n'avait pas été sauvegardé avec

succès, parce que si une demande de sauvegarde d'état est refusée, les informations d'annonce correspondantes sont éliminées par le gardien d'état.

Note : le gardien d'état du point d'extrémité 1 doit conserver une transposition entre l'état (M) et l'état (S) pour que cette optimisation fonctionne.

Note : le seul état qui soit acquitté par ce dispositif est l'état qui a été créé en combinant l'état utilisé pour la compression du message et le message lui-même. Pour tous les autres cas le `acked_state_id` doit être utilisé.

Note : il existe une possibilité que l'état (S) soit éliminé à cause d'un manque de mémoire d'état même si les informations d'annonce sont bien transmises. Cette possibilité doit être prise en compte (autrement un échec de décompression peut se produire) ; cela peut être fait en utilisant le paramètre SigComp `state_memory_size` qui est annoncé par le mécanisme de retours SigComp. Le point d'extrémité peut utiliser ce paramètre pour déduire si une demande de création d'état a échoué à cause d'un manque de mémoire.

## 7. Considérations sur la sécurité

Les caractéristiques du présent document n'ajoutent aucun risque à ceux mentionnés dans la [RFC3320].

## 8. Considérations relatives à l'IANA9

Le présent document n'exige aucune implication de la part de l'IANA.

## 9. Remerciements

Merci à Carsten Bormann, Christopher Clanton, Miguel Garcia, Lars-Erik Jonsson, Khiem Le, Mats Nordberg, Jonathan Rosenberg et Krister Svanbro de leurs précieux apports.

## 10. Considérations de droits de propriété intellectuelle

Des revendications de droits de propriété intellectuelle ont été notifiées à l'IETF à l'égard de tout ou partie de la spécification contenue dans ce document. Pour plus d'informations consulter la liste en ligne des revendications de droits.

## 11. Références

[RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (*Mise à jour par [RFC3265](#), [RFC3853](#), [RFC4320](#), [RFC4916](#), [RFC5393](#), [RFC6665](#), [RFC8217](#)*)

[RFC3320] R. Price, et autres, "[Compression de signalisation](#) (SigComp)", janvier 2003. (*MàJ par [RFC4896](#)*) (P.S.)

## 12. Adresse des auteurs

Hans Hannu  
Box 920  
Ericsson AB  
SE-971 28 Lulea, Sweden  
téléphone : +46 920 20 21 84  
mél : [hans.hannu@epl.ericsson.se](mailto:hans.hannu@epl.ericsson.se)

Jan Christoffersson  
Box 920  
Ericsson AB  
SE-971 28 Lulea, Sweden  
téléphone : +46 920 20 28 40  
mél : [jan.christoffersson@epl.ericsson.se](mailto:jan.christoffersson@epl.ericsson.se)

Stefan Forsgren  
mél : [StefanForsgren@alvishagglunds.se](mailto:StefanForsgren@alvishagglunds.se)

Ka-Cheong Leung  
Department of Computer Science  
Texas Tech University  
Lubbock, TX 79409-3104  
United States of America  
téléphone : +1 806 742-3527  
mél : [kcleung@cs.ttu.edu](mailto:kcleung@cs.ttu.edu)

Zhigang Liu  
Nokia Research Center  
6000 Connection Drive  
Irving, TX 75039, USA  
téléphone : +1 972 894-5935  
mél : [zhigang.c.liu@nokia.com](mailto:zhigang.c.liu@nokia.com)

Richard Price  
Roke Manor Research Ltd  
Romsey, Hants, SO51 0ZN,  
United Kingdom  
téléphone : +44 1794 833681  
mél : [richard.price@roke.co.uk](mailto:richard.price@roke.co.uk)

### 13. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2003). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de droits de reproduction ci-dessus et le présent paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou ses successeurs ou ayant droits.

Le présent document et les informations contenues sont fournis 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 encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

#### Remerciement

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