Groupe de travail Réseau
Request pour Comments : 5049
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

C. Bormann, Universitaet Bremen TZI
Z. Liu, Nokia Research Center
R. Price, EADS Defence et Security Systems Ltd
G. Camarillo, éd., Ericsson
décembre 2007

Application de la compression de signalisation (SigComp) au protocole d'initialisation de session (SIP)

Résumé
Le présent document décrit des spécificités de l'application de la compression de signalisation (SigComp, *Signaling Compression*) au protocole d'initialisation de session (SIP, *Session Initiation Protocol*) telles que les valeurs par défaut minimum des paramètres de SigComp, de la gestion de compartiment et d'état, et de quelques problèmes de SigComp sur TCP. Toute mise en œuvre de SigComp utilisée avec SIP doit se conformer au présent document et à SigComp, et de plus, prendre en charge le dictionnaire statique de SIP et du protocole de description de session (SDP, *Session Description Protocol*).

Table des Matières

1. Introduction

SigComp [RFC3320] est une solution pour compresser les messages générés par les protocoles d'application. Bien que son principal objet soit de compresser les messages SIP [RFC3261], la solution elle même a été intentionnellement conçue pour être neutre à l'égard de l'application afin qu'elle puisse être appliquée à tout protocole d'application ; ceci est noté par ANY/SigComp. Par conséquent, de nombreuses spécificités dépendant de l'application sont laissées en dehors de la norme de base. Il est prévu qu'une spécification distincte soit utilisée pour décrire ces spécificités quand SigComp est appliqué à un protocole d'application particulier.

Le présent document lie SigComp et SIP ; ceci est noté SIP/SigComp.

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].

3. Conformité à cette spécification

Toute mise en œuvre de SigComp qui est utilisée pour la compression de messages SIP DOIT se conformer au présent document, ainsi qu'à la [RFC3320]. De plus, elle doit prendre en charge le dictionnaire statique SIP/SDP, comme spécifié dans la [RFC3485], et le mécanisme pour découvrir la prise en charge de SigComp à la couche SIP , comme spécifié dans la [RFC3486].

4. Valeurs minimum des paramètres SigComp pour SIP/SigComp

Afin de prendre en charge une large gamme de capacités entre les points d'extrémité qui mettent en œuvre SigComp, SigComp définit des paramètres pour décrire le comportement de SigComp (voir le paragraphe 3.3 de la [RFC3320]). Pour chaque paramètre, la [RFC3320] spécifie une valeur minimum que tout point d'extrémité SigComp DOIT prendre en charge pour ANY/SigComp. Ces valeurs minimum ont été déterminées en considérant tous les appareils imaginables dans lesquels SigComp peut être mis en œuvre. L'adaptabilité a aussi été considérée comme étant un facteur clé.

Cependant, certaines des valeurs minimum spécifiées dans la [RFC3320] sont trop petites pour permettre de bonnes performances pour la compression de message SIP. Donc, elles sont augmentées pour SIP/SigComp comme spécifié dans les paragraphes suivants. Pour être complet, les paramètres qui sont les mêmes pour SIP/SigComp que pour ANY/SigComp sont aussi mentionnés.

Les nouvelles valeurs minimum sont spécifiques de SIP/SigComp et donc, ne s'appliquent à aucun autre protocole d'application. Un point d'extrémité SIP/SigComp PEUT offrir des ressources supplémentaires au dessus des valeurs minimum spécifiées dans le présent document si il en est de disponibles ; ces ressources peuvent être annoncées aux points d'extrémité distants comme décrit au paragraphe 9.4.9 de la [RFC3320].

4.1 Taille de mémoire de décompression (DMS) pour SIP/SigComp

Valeur minimum pour ANY/SigComp : 2048 octets, comme spécifié au paragraphe 3.3.1 de la [RFC3320].

Valeur minimum pour SIP/SigComp : 8192 octets.

Raison : une DMS de 2048 octets est trop petite pour la compression de message SIP car cela limite sérieusement le taux de compression et rend même la compression impossible pour certains messages. Par exemple, la condition établie par la [RFC3320] pour SigComp sur UDP signifie : C + 2*B + R + 2*S + 128 < DMS (chaque terme est décrit ci-dessous). Donc, si la DMS est trop petite, au moins un de C, B, R, ou S va être sévèrement restreint. Par ailleurs, DMS est une mémoire qui est seulement temporairement nécessaire durant la décompression d'un message SigComp (la mémoire peut être reprise quand le message a été décompressé). Donc, une exigence de 8 k octets ne

devrait pas causer de problème pour un point d'extrémité qui met déjà en œuvre SIP, SigComp, et les applications qui utilisent SIP.

C : taille du message d'application compressé, dépend de R.
B : taille du code d'octet. Note : deux copies -- une au titre du message SigComp et une en mémoire de machine virtuelle de décompresseur universel (UDVM, *Universal Decompressor Virtual Machine*).
R : taille de la mémoire tampon circulaire dans la mémoire d'UDVM.
S : tout état supplémentaire chargé autre que celui créé à partir du contenu de la mémoire tampon circulaire à la fin de la décompression (comme pour B, deux copies de S sont nécessaires).
128 : la plus petite adresse dans la mémoire d'UDVM pour y copier le code d'octet.

4.2 Taille de mémoire d'état (SMS) pour SIP/SigComp

Valeur minimum pour ANY/SigComp : 0 (zéro) octet, comme spécifié au paragraphe 3.3.1 de la [RFC3320].

Valeur minimum pour SIP/SigComp : 2048 octets.

Raison : une SMS non de zéro permet à un point d'extrémité de charger un état dans le premier message SIP envoyé à un point d'extrémité distant sans incertitude sur la question de savoir si le point d'extrémité distant va avoir assez de mémoire pour conserver cet état. Une SMS non zéro exige évidemment que la mise en œuvre de SIP/SigComp conserve l'état. Sur la base de l'observation qu'il y a peu à gagner à une compression SigComp sans état, l'hypothèse est que des mises en œuvre de SIP purement sans état ont peu de chances de fournir une fonction SigComp. Les mises en œuvre à états pleins devraient avoir peu de problèmes à conserver 2 k d'état supplémentaire pour chaque compartiment (voir la Section 9).

Note : SMS est un paramètre qui s'applique à chaque compartiment individuel. Un point d'extrémité PEUT offrir des valeurs de SMS différentes pour différents compartiments tant que la valeur de SMS n'est pas inférieure à 2048 octets.

4.3 Cycles par bit (CPB) pour SIP/SigComp

Valeur minimum pour ANY/SigComp: 16, comme spécifié au paragraphe 3.3.1 de la [RFC3320].

Valeur minimum pour SIP/SigComp : 16 (comme ci-dessus).

4.4 Version SigComp (SV) pour SIP/SigComp

Pour ANY/SigComp : 0x01, comme spécifié au paragraphe 3.3.2 de la [RFC3320].
Pour SIP/SigComp : ≥ 0x02 (au moins SigComp + NACK).

Noter que cela implique que les dispositions de la [RFC4077] s'appliquent. C'est-à-dire, que les messages d'échec de décompression résultent en le renvoi de messages SigComp NACK au compresseur d'origine. Cela implique aussi que le compresseur n'a pas besoin d'utiliser les méthodes détaillées au paragraphe 2.4 de la [RFC4077] (Détection de la prise en charge du NACK) ; par exemple, il peut utiliser des méthodes de compression optimistes directememnt à partir du résultat.

4.5 État disponible en local (LAS) pour SIP/SigComp

LAS minimum pour ANY/SigComp : aucun, voir le paragraphe 3.3.3 de la [RFC3320].
LAS minimum pour SIP/SigComp : le dictionnaire statique SIP/SDP défini dans la [RFC3485].

Noter que comme la prise en charge du dictionnaire statique SIP/SDP est obligatoire, elle n'a pas besoin d'être annoncée.

5. Délimitation des messages SIP et SigComp sur le même accès

Afin de limiter le nombre d'accès requis par un point d'extrémité à capacité SigComp, il est possible de permettre qu'arrivent sur le même accès à la fois des messages SigComp et des messages SIP "ordinaires" (c'est-à-dire, des messages SIP non compressés sans en-tête SigComp).

Pour un transport fondé sur le message comme UDP ou le protocole de transmission de commandes de flux (SCTP, *Stream Control Transmission Protocol*) distinguer entre les messages SigComp et non SigComp peut être fait par message. Le point d'extrémité receveur vérifie le premier octet de la charge utile UDP/SCTP pour déterminer si le message a été compressé en utilisant SigComp. Si les bits de poids fort (MSB, *Most Significant Bit*) de l'octet sont "11111", alors le message est considéré comme étant un message SigComp et est analysé conformément à la [RFC3320]. Si les MSB de l'octet ont toute autre valeur, le message est alors supposé être un message SIP non compressé, et il est passé directement à l'application sans autre effet sur la couche SigComp.

Pour un transport fondé sur le flux comme TCP, distinguer entre les messages SigComp et non SigComp doit être fait par connexion. Le point d'extrémité receveur vérifie le premier octet du flux de données TCP pour déterminer si le flux a été compressé en utilisant SigComp. Si les MSB de l'octet sont "11111", le flux est alors considéré comme contenant des messages SigComp et est analysé selon la [RFC3320]. Si les MSB de l'octet prennent une autre valeur, le flux est alors supposé contenir des messages SIP non compressés, et il est passé directement à l'application sans autre effet sur la couche SigComp. Noter que les délimiteurs de message SigComp NE DOIVENT PAS être utilisés si le flux contient des messages SIP non compressés.

Les applications NE DOIVENT PAS mélanger des messages SIP et des messages SigComp sur une même connexion TCP. Si la connexion TCP est utilisée pour porter des messages SigComp, alors tous les messages envoyés sur la connexion DOIVENT avoir un en-tête SigComp et être délimités par l'utilisation de 0xFFFF, comme décrit dans la [RFC3320].

La Section 11 de la [RFC4896] détaille un ensemble simple de codes d'octets, destinés à être "bien connus", qui mettent en œuvre un algorithme de décompression nulle. Ces codes d'octets permettent effectivement aux homologues SigComp d'envoyer des messages SigComp choisis avec des données non compressées. Si une mise en œuvre de SIP a des raisons d'envoyer à la fois des messages SIP compressés et non compressés sur une seule connexion TCP, le compressor peut recevoir pour instruction d'utiliser ces codes d'octets pour envoyer des messages SIP non compressés qui sont aussi des messages SigComp valides.

6. Mode continu sur TCP

Le mode continu est une caractéristique particulière de SigComp, qui est destinée à améliorer le ratio global de compression pour les connexions à longue durée de vie. Son utilisation exige un accord préalable entre le compresseur et le décompresseur SigComp. Le mode continu n'est pas utilisé avec SIP/SigComp.

Raison : le mode continu exige que le transport lui-même fournisse un certain niveau de protection contre les attaques de déni de service. TCP seul n'est pas considéré comme fournissant une protection suffisante.

7. Messages SIP trop grands

SigComp ne prend pas en charge la compression de messages de plus de 64 k. Donc, si une application SIP qui envoie des messages SIP compressés à une autre application SIP sur une connexion de transport (par exemple, une connexion TCP) a besoin d'envoyer un message SIP de plus de 64 k, l'application SIP NE DOIT PAS envoyer le message sur la même connexion TCP. L'application SIP DEVRAIT envoyer le message sur une connexion de transport différente (pour ce faire, l'application SIP peut avoir besoin d'établir une nouvelle connexion de transport).

8. Retransmissions SIP

Quand des messages SIP sont retransmis, ils ont besoin d'être recompressés, en tenant compte de tous les états SigComp qui peuvent avoir été créés ou invalidés depuis la précédente transmission. Les mises en œuvre NE DOIVENT PAS mettre en antémémoire le résultat de la compression de message et retransmettre un tel résultat d'antémémoire.

La raison de ce comportement est qu'il est impossible de savoir si l'échec qui a causé la retransmission s'est produit sur le message retransmis ou sur la réponse à ce message. Si la réponse a été perdue, tous les changements d'état effectués par la première instance du message retransmis vont déjà avoir eu lieu. Si ces changements d'état suppriment un état sur lequel le message précédemment transmis s'appuie, alors la retransmission du même message compressé conduirait à un échec de

décompression.

Noter qu'une retransmission SIP peut être causée par la perte du message original ou de sa réponse par un échec de décompression. Dans ce cas, un NACK aurait été envoyé par le décompresseur au compresseur, qui peut utiliser les informations de ce message NACK pour ajuster ses paramètres de compression. Noter que, sur un transport non fiable, un tel message NACK peut aussi être perdu, de sorte que si un compresseur a utilisé une forme de compression optimiste, il PEUT vouloir passer à une méthode qui ait moins de chances de causer une forme d'échec de décompression quand il compresse une retransmission SIP.

9. Compartimentage et gestion d'état pour SIP/SigComp

Une application qui échange du trafic compressé avec une application distante a un compartiment qui contient les informations d'état nécessaires pour compresser les messages sortants et décompresser les messages entrants. Pour augmenter l'efficacité de la compression, l'application doit allouer des compartiments distincts aux différentes applications distantes.

9.1 Identification de l'application distante

Les applications SIP/SigComp identifient les applications distantes par leurs identifiants SIP/SigComp. Chaque application SIP/SigComp DOIT avoir un nom de ressource universel (URN, *Uniform Resource Name*) d'identifiant SIP/SigComp qui identifie de façon univoque l'application. L'usage d'un URN fournit un nom persistant et unique pour l'identifiant SIP/SigComp. Cela donne aussi un moyen aisé de garantir l'unicité. Cet URN DOIT être persistant tant que l'application mémorise l'état du compartiment relatif aux autres applications SIP/SigComp.

Une application SIP/SigComp DEVRAIT utiliser un URN d'identifiant universellement unique (UUID, *Universally Unique IDentifier*) comme identifiant SIP/SigComp, à cause des difficultés de comparaison d'égalité pour les autres sortes d'URN. L'URN UUID [RFC4122] permet un calcul non centralisé d'un URN fondé sur l'heure, des noms uniques (comme une adresse de contrôle d'accès au support (MAC, *Media Access Control*)) ou un générateur de nombres aléatoires. Si un schéma d'URN autre qu'un UUID est utilisé, l'URN DOIT être choisi de telle sorte que l'application puisse être certaine qu'aucune autre application SIP/SigComp ne va choisir la même valeur d'URN.

Noter que la définition de l'identifiant SIP/SigComp est similaire à celle de l'identifiant d'instance dans la [RFC5626]. Une différence est qu'il est seulement exigé que les identifiants d'instance soient uniques dans leur adresse d'enregistrement (AoR, *Address of Record*) tandis qu'il est exigé des identifiants SIP/SigComp qu'ils soient uniques au monde.

Même si les identifiants d'instance sont seulement obligés d'être uniques au sein de leur AoR, des appareils peuvent choisir de générer des identifiants d'instance uniques au monde. Un appareil avec un identifiant d'instance unique au monde DEVRAIT utiliser son identifiant d'instance comme identifiant SIP/SigComp.

Note : utiliser la même valeur pour identifiant d'une instance d'entité et SIP/SigComp améliore le taux de compression des champs d'en-tête qui portent les deux identifiants (par exemple, un champ d'en-tête Contact dans une demande REGISTER).

Des goupes de serveurs qui partagent un état SIP/SigComp entre eux DOIVENT utiliser le même identifiant SIP/SigComp pour tous leurs serveurs.

Les identifiants SIP/SigComp sont portés dans le paramètre d'identifiant de ressource universel (URI, *Uniform Resource Identifier*) SIP "sigcomp-id" ou le champ d'en-tête Via. Le paramètre d'URI SIP "sigcomp-id" est un "uri-parameter", comme défini par l'ABNF *(Augmented Backus-Naur Form)* SIP du paragraphe 25.1 de la [RFC3261]. Voici son ABNF [RFC4234]:

```
uri-sip-sigcomp-id = "sigcomp-id=" 1*paramchar
```

Le paramètre d'URI SIP "sigcomp-id" DOIT contenir un URN [RFC2141].

Le paramètre Via "sigcomp-id" est une "via-extension", comme défini par l'ABNF SIP (paragraphe 25.1 de la [RFC3261]). Voici son ABNF [RFC4234] :

via-sip-sigcomp-id = "sigcomp-id" EQUAL LDQUOT *(qdtext / quoted-pair) RDQUOT

Le paramètreVia "sigcomp-id" DOIT contenir un URN [RFC2141].

Voici un exemple de paramètre d'URI SIP "sigcomp-id" :

sigcomp-id=urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128

Voici un exemple d'un champ d'en-tête Via avec un paramètre "sigcomp-id" :

```
Via: SIP/2.0/UDP server1.example.com:5060
  ;branch=z9hG4bK87a7
  ;comp=sigcomp
  ;sigcomp-id="urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128"
```

Voici un exemple de demande REGISTER qui porte des paramètres"sigcomp-id" dans une entrée Via et dans le champ d'en-tête Contact. De plus, elle porte aussi un paramètre de champ d'en-tête Contact "+sip.instance".

```
REGISTER sip:example.net SIP/2.0
Via: SIP/2.0/UDP 192.0.2.247:2078;branch=z9hG4bK-et736vsjirav;
 rport;sigcomp-id="urn:uuid:2e5fdc76-00be-4314-8202-1116fa82a473"
From: "Joe User" <sip:2145550500@example.net>;tag=6to4gh7t5j
To:  "Joe User" <sip:2145550500@example.net>
Call-ID: 3c26700c1adb-lu1lz5ri5orr
CSeq: 215196 REGISTER
Max-Forwards: 70
Contact: <sip:2145550500@192.0.2.247:2078;
 sigcomp-id=urn:uuid:2e5fdc76-00be-4314-8202-1116fa82a473>;
 q=1.0; expires=3600;
 +sip.instance="<urn:uuid:2e5fdc76-00be-4314-8202-1116fa82a473>"
Content-Length: 0
```

Les messages SIP sont confrontés aux identifiants d'application distante comme suit :

Demandes sortantes : l'identifiant d'application distante est l'identifiant SIP/SigComp de l'URI auquel la demande est envoyée. Si l'URI ne contient pas identifiant SIP/SigComp, l'identifiant d'application distante est l'adresse IP plus l'accès du datagramme qui porte la demande pour les protocoles de transport sans connexion, et la connexion de transport (par exemple, une connexion TCP) portant la demande pour les protocole de transport en mode connexion (c'est pour prendre en charge les applications SIP/SigComp traditionnelles).

Réponses entrantes : l'identifiant d'application distante est le même que celui de la demande envoyée précédemment qui a initié la transaction à laquelle appartient la réponse.

Demandes entrantes : l'identifiant d'application distante est l'identifiant SIP/SigComp de l'entrée Via supérieure. Si le champ d'en-tête Via ne contient pas d'identifiant SIP/SigComp, l'identifiant d'application distante est l'adresse IP de source plus l'accès du datagramme qui porte la demande pour les protocoles de transport sans connexion, et la connexion de transport (par exemple, une connexion TCP) portant la demande pour les protocoles de transport en mode connexion (c'est pour prendre en charge les applications SIP/SigComp traditionnelles).

Réponses sortantes : l'identifiant d'application distante est le même que celui de la demande reçue précédemment qui a initiée la transaction à laquelle appartient la réponse. Noter que, du fait du traitement standard de champ d'en-tête SIP Via , cet identifiant va être présent dans l'entrée Via supérieure dans de telles réponses (pour autant qu'il était présent dans l'entrée Via supérieure de la demande précédemment reçue).

Une application SIP/SigComp qui place son URI avec le paramètre "comp=sigcomp" dans un champ d'en-tête DOIT ajouter à cet URI un paramètre "sigcomp-id" avec son identifiant SIP/SigComp.

Une application SIP/SigComp qui génère sa propre entrée Via contenant le paramètre "comp=sigcomp" DOIT ajouter à cette entrée Via un paramètre "sigcomp-id" avec son identifiant SIP/SigComp.

Un identifiant d'application distante est transposé en un identifiant de compartiment SigComp particulier en suivant les

règles du paragraphe 9.3.

9.2 Règles de comparaison d'identifiants

Les comparaisons pour égalité entre les identifiants SIP/SigComp sont effectuées en utilisant les règles pour égalité d'URN spécifiques du schéma dans l'URN. Si l'élément qui effectue les comparaisons ne comprend pas le schéma d'URN, il effectue les comparaisons en utilisant les règles d'égalité lexicale définies dans la [RFC2141]. L'égalité lexicale peut résulter en deux URN qui sont considérés comme inégaux alors qu'ils sont en fait égaux. Dans cet usage spécifique des URN, le seul élément qui fournit l'URN est l'application SIP/SigComp identifiée par cet URN. Par suite, l'application SIP/SigComp DEVRAIT fournir des URN lexicalement équivalents dans chaque enregistrement qu'elle génère. Ceci est probablement le comportement normal dans tous les cas ; les applications ne vont probablement pas modifier la valeur de leurs identifiants SIP/SigComp afin qu'ls restent fonctionnellement équivalents bien que lexicographiquement différents des identifiants précédents.

9.3 Ouverture et fermeture de compartiment

Les applications SIP ont besoin de savoir quand ouvrir un nouveau compartiment et quand le clore. La durée de vie des compartiments SIP/SigComp est lée à l'état d'enregistrement. Les compartiments sont ouverts au moment de l'enregistrement SIP et sont normalement clos quand l'enregistrement arrive à expiration ou est annulé.

Note : lier la durée de vie des compartiments SIP/SigComp à l'état d'enregistrement limite l'applicabilité de cette spécification. En particulier, les agents d'utilisateur SIP qui ne s'enregistrent pas mais, par example, traitent seulement des transactions PUBLISH ou SUBSCRIBE/NOTIFY ne sont pas capables de créer des compartiments SIP/SigComp suivant la présente spécification. Les précédentes révisions de cette spécification définissaient aussi des compartiments valides durant une transaction ou dialogue SIP. Ces compartiments couvraient toutes les entités SIP possibles, incluant celles qui ne traitent pas les transactions REGISTER. Cependant, il a été décidé d'éliminer ces types de compartiments parce que la complexité qu'ils introduisent (par exemple, les serveurs mandataires de bordure étaient obligés de garder l'état de dialogue) est supérieure aux avantages qu'ils apportent dans la plupart des scénarios de déploiement.

Généralement, tous les états créés durant la vie d'un compartiment vont être "logiquement" supprimés quand le compartiment est fermé. Comme décrit au paragraphe 6.2 de la [RFC3320], une suppression logique ne peut devenir une suppression physique que quand aucun compartiment qui a créé l'état (le même) ne continue d'exister.

Un point d'extrémité SigComp peut offrir de conserver un état créé à la demande d'un point d'extrémité SigComp homologue au delà de la durée de vie par défaut d'un compartiment (c'est-à-dire, au delà de la durée de son enregistrement associé). Cela peut être utilisé pour améliorer l'efficacité de compression des messages SIP suivants générés par la même application distante de point d'extrémité SigComp homologue. Pour indiquer qu'un tel état va continuer d'être disponible, le point d'extrémité SigComp peut informer son point d'extrémité SigComp homologue en annonçant l'identifiant d'état (partiel) dans les paramètres SigComp retournés à la fin de l'enregistrement qui était supposé limiter la durée de vie de l'état SigComp. Cela signale que l'état va être maintenu. La prise en charge obligatoire du mécanisme d'accusé de réception négatif (NACK, *Negative Acknowledgement*) SigComp [RFC4077] dans SIP/SigComp assure qu'il est possible de récuperer des erreurs de synchronisation concernant les durées de vie de compartiment.

Les erreurs de mise en œuvre de la gestion d'un compartiment sont un souci de fonctionnement qui va probablement conduire à des défaillances sporadiques difficiles à diagnostiquer. Les décompresseurs peuvent donc vouloir mettre en antémémoire l'ancien état et, si il est encore disponible, permettre l'accès lors de l'enregistrement d'informations de diagnostic. Les compresseurs et décompresseurs utilisent tous deux le mécanisme SigComp d'accusé de réception négatif (NACK) [RFC4077] pour récupérer de situations où un tel ancien état peut n'être plus disponible.

Une transaction REGISTER cause l'ouverture par une application d'un nouveau compartiment qui va être valide pour la durée de l'enregistrement établie par la transaction REGISTER.

Une application SIP qui a besoin d'envoyer un REGISTER SIP compressé (c'est-à-dire, un agent d'utilisateur qui génère un REGISTER ou un serveur mandataire qui en relaie un à son prochain bond) DEVRAIT ouvrir un compartiment pour l'identifiant d'application distante de la demande. Une application SIP qui reçoit un REGISTER SIP compressé (c'est-à-dire, le registraire ou un mandataire qui relaie le REGISTER à son prochain bond) DEVRAIT ouvrir un compartiment pour l'identifiant d'application distante de la demande.

Ces compartiments PEUVENT être clos si la demande REGISTER reçoit en réponse autre chose qu'un 2xx final, ou quand l'enregistrement expire ou est annulé. Cependant, les applications PEUVENT aussi choisir de garder ces compartiments ouverts pour une plus longue période, comme mentionné précédemment. Pour un enregistrement réussi donné, les applications NE DEVRAIENT PAS clore leurs compartiments associés avant la fin de l'enregistrement.

Note : un réseau SIP peut être configuré à ce que le trafic SIP régulier de et vers un agent d'utilisateur traverse un ensemble différent de mandataires que celui de la transaction REGISTER initiale. Le chemin que suit la transaction REGISTER est normalement déterminé par les données de configuration. Le chemin que traversent les demandes suivantes est déterminé par les champs d'en-tête Path [RFC3327] et Service-Route [RFC3308] dans la transaction REGISTER et par les champs d'en-tête Record-Route et Route dans les transactions de création de dialogue. Les précédentes révisions de ce document prenaient en charge l'utilisation de différents chemins pour différents types de trafic. Cependant, pour des raisons de simplicité, le présent document suppose mainteant que les réseaux qui utilisent la compression vont être configurés de telle sorte que les demandes suivantes suivent le même chemin que la transaction REGISTER initiale afin de réaliser la meilleure compression possible. La Section 10 donne aux administrateurs de réseau des recommandations pour configurer les réseaux de façon appropriée.

Si, suivant les règles ci-dessus, une application SIP est supposée ouvrir un compartiment pour un identifiant d'application distante pour lequel elle a déjà un compartiment (par exemple, l'application SIP s'enregistre auprès d'un second registraire en utilisant le même serveur mandataire de bordure que pour son enregistrement auprès de son premier registraire) l'application SIP DOIT utiliser le compartiment déjà existant. C'est-à-dire que l'application SIP NE DOIT PAS ouvrir un nouveau compartiment.

9.4 Absence de compartiment

L'utilisation d'une compression sans état (c'est-à-dire, la compression sans compartiment) n'est normalement pas intéressante et peut même résulter en l'expansion du message. Donc, si une application SIP n'a pas un compartiment pour un message qu'elle a besoin d'envoyer, elle PEUT choisir de ne pas le compresser même en présence du paramètre "comp=sigcomp". La Section 5 décrit comment une application SIP peut envoyer des messages compressés et non compressés sur la même connexion TCP. Noter que la [RFC3486] déclare : "Si l'URI de prochain bond contient le paramètre comp=sigcomp, le client DEVRAIT compresser la demande en utilisant SigComp".

L'expérience depuis la rédaction de la [RFC3486] a montré que la compression sans état est, dans la plupart des cas, sans intérêt. C'est pourquoi il n'est pas recommandé de continuer de l'utiliser plus longtemps.

10. Recommandations pour les administrateurs de réseau

Les administrateurs de réseau peuvent configurer leurs réseaux de façon à augmenter l'efficacité de la compression réalisée. Les recommandations suivantes aident les administrateurs de réseau à effectuer cette tâche.

Pour un agent d'utilisateur donné, le chemin établi pour les demandes entrantes (créées par un champ d'en-tête Path) et pour les demandes sortantes (créées par un champ d'en-tête Service-Route) est normalement le même. Cependant, les registraires peuvent, si ils le souhaitent, insérer des mandataires dans le dernier chemin qui n'apparaît pas dans le premier chemin et vice-versa. ll est RECOMMANDÉ que les registraires soient configurés de telle façon que les mandataires qui effectuent la compression SigComp apparaissent dans les deux chemins.

Les chemins décrits précédemment s'appliquent aux demandes envoyées en-dehors d'un dialogue. Les demandes à l'intérieur d'un dialogue suivent un chemin construit en utilisant les champs d'en-tête Record-Route. Il est RECOMMANDÉ que les mandataires effectuant SigComp qui sont sur le chemin des demandes en dehors d'un dialogue soient configurés à se placer eux-mêmes (en s'insérant dans les champs d'en-tête Record-Route) dans les chemins utilisés pour les demandes à l'intérieur des dialogues.

Quand l'enregistrement d'un agent d'utilisateur arrive à expiration, les serveurs mandataires qui effectuent la compression peuvent clore leur compartiment SIP/SigComp associé. Si l'agent d'utilisateur est impliqué dans un dialogue établi avant l'expiration de l'enregistrement, les demandes suivantes au sein du dialogue ne peuvent plus être compressées. Afin d'éviter cette situation, il est RECOMMANDÉ que les agents d'utilisateur soient enregistrés tant qu'ils sont impliqués dans un dialogue.

11. Accords privés

Les mises en œuvre de SIP/SigComp qui sont soumises à des accords privés PEUVENT s'écarter de la présente spécification, si les accords privés le spécifient sans ambiguïté. Les candidats plausibles pour de tels écarts incluent :
- o les valeurs minimum (Section 4) ;
- o l'utilisation du mode continu (Section 6) ;
- o la définition de compartiment (Section 9).

12. Rétro-compatibilité

SigComp a un certain nombre de paramètres qui peuvent être configurés par point d'extrémité. Le présent document spécifie un profil pour SigComp quand il est utilisé pour la compression de SIP qui contraint plus la gamme que peuvent prendre certains de ces paramètres. Des exemples de cela sont la taille de mémoire de décompresseur, la taille de mémoire d'état, et la version SigComp (prise en charge du NACK). De plus, le présent document spécifie comment les applications SIP/SigComp devraient effectuer la transposition de compartiment.

Lors de la rédaction du présent document, il existait déjà quelques déploiements de SIP/SigComp. Les règles du présent document ont été conçues pour maximiser l'interopérabilité avec ces mises en œuvre anciennes de SIP/SigComp. Néanmoins, les mises en œuvre devraient être conscientes que les mises en œuvre anciennes de SIP/SigComp peuvent ne pas se conformer à la présente spécification. Des exemples de problèmes avec les applications anciennes vont être une plus petite DMS que ce que le présent document rend obligatoire, la non prise en charge du NACK, ou une transposition différente de compartiment.

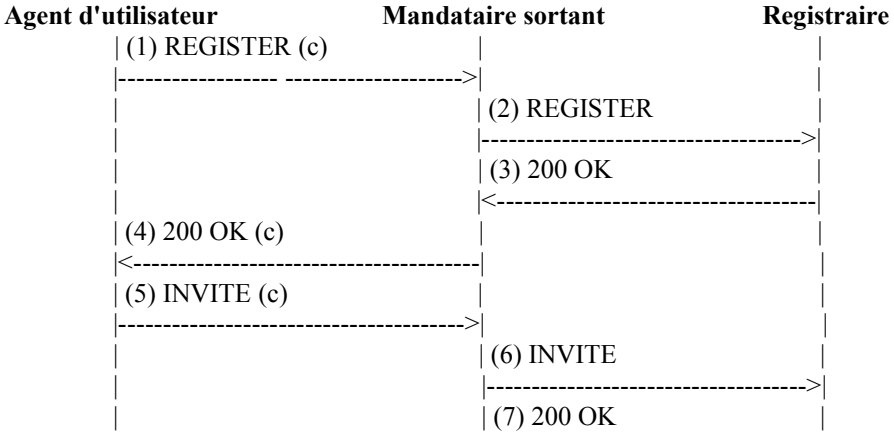
13. Interactions avec la sécurité de la couche Transport (TLS)

Les points d'extrémité qui échangent du trafic SIP sur une connexion TLS [RFC4346] peuvent utiliser la compression fournie par TLS. Deux points d'extrémité qui échangent du trafic SIP/SigComp sur une connexion TLS qui fournit la compression ont d'abord besoin de compresser les messages SIP en utilisant SigComp et ensuite de les passer à la couche TLS, qui va les compresser à nouveau. À la réception des données, l'ordre de traitement est inversé.

Cependant, compresser deux fois les messages de cette façon n'apporte normalement pas de gain significatif. Une fois qu'un message est compressé en utilisant SigComp, TLS n'est généralement pas capable de le compresser plus. Donc, TLS va normalement être seulement capable de compresser le code SigComp envoyé entre le compresseur et le décompresseur. Comme le gain d'avoir le code SigComp compressé devrait être minimal dans la plupart des cas, il n'est PAS RECOMMANDÉ d'utiliser la compression TLS quand la compression SigComp est utilisée.

14. Exemple

La Figure 1 montre un exemple de flux de messages où l'agent d'utilisateur et le mandataire sortant échangent du trafic SIP compressé. Les messages compressés sont marqués d'un (c).



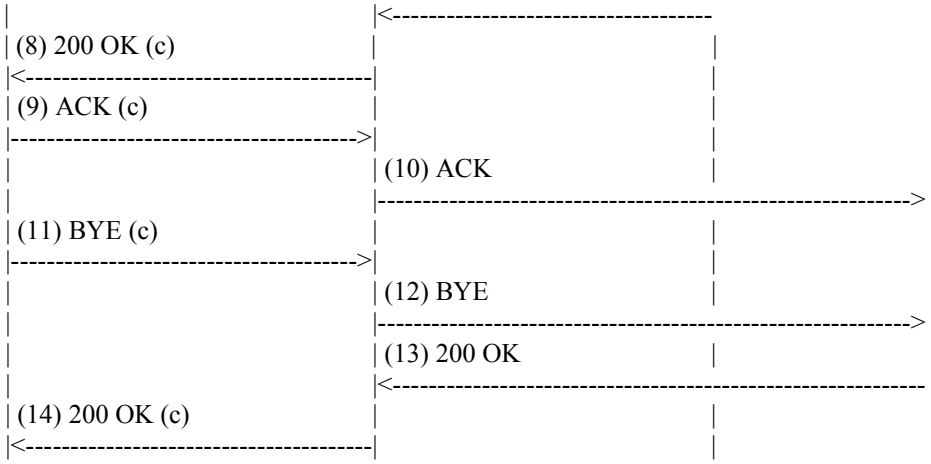


Figure 1 : Exemple de flux de messages

L'agent d'utilisateur dans la Figure 1 est initialement configuré (par exemple, en utilisant le cadre de configuration SIP [RFC6080]) avec l'URI de son mandataire sortant. Cet URI contient l'identifiant SIP/SigComp du mandataire sortant, appelé "Outbound-id", dans un paramètre "sigcomp-id".

Quand l'agent d'utilisateur envoie une demande initiale REGISTER (1) à l'URI du mandataire sortant, l'agent d'utilisateur ouvre un nouveau compartiment pour "Outbound-id". Ce compartiment va être valide pour la durée de l'enregistrement, au moins.

À réception de cette demande REGISTER (1), le mandataire sortant ouvre un nouveau compartiment pour l'identifiant SIP/SigComp qui apparaît dans le paramètre "sigcomp-id" de l'entrée Via supérieure. Cet identifiant, qui est l'identifiant SIP/SigComp de l'agent d'utilisateur, est appelé "UA-id". Le compartiment ouvert par le mandataire sortant va être valide pour la durée de l'enregistrement, au moins. Le mandataire sortant ajoute un champ d'en-tête Path avec son propre URI, qui contient l'identifiant SIP/SigComp "Outbound-id", à la demande REGISTER et la relaie au registraire (2).

Quand le registraire reçoit la demande REGISTER (2), il construit le chemin que les futures demandes entrantes (à l'agent d'utilisateur) vont suivre en utilisant les champs d'en-tête Contact et Path. Les futures demandes entrantes vont traverser le mandataire sortant avant d'atteindre l'agent d'utilisateur.

Le registraire construit aussi le chemin que les futures demandes sortantes (provenant de l'agent d'utilisateur) vont suivre et les place dans un champ d'en-tête Service-Route dans une réponse 200 (OK) (3). Les futures demandes sortantes vont toujours traverser le mandataire sortant. Le registraire s'est assuré que le mandataire sortant qui effectue la compression traite les demandes entrantes et sortantes.

Quand le mandataire sortant reçoit une réponse 200 (OK) (3), il inspecte l'entrée Via supérieure. L'identifiant SIP/SigComp "UA-id" de cette entrée correspond à celui de compartiment créé antérieurement. Donc, le mandataire sortant utilise ce compartiment pour le compresser et le relayer à l'agent d'utilisateur.

À réception de la réponse 200 (OK) (4), l'agent d'utilisateur mémorise le champ d'en-tête Service-Route afin de l'utiliser pour envoyer les futures demandes sortantes. Le champ d'en-tête Service-Route contient l'URI du mandataire sortant, qui contient l'identifiant SIP/SigComp "Outbound-id".

Plus tard, l'agent d'utilisateur a besoin d'envoyer une demande INVITE (5). Conformément au champ d'en-tête Service-Route reçu précédemment, l'agent d'utilisateur envoie la demande INVITE (5) à l'URI du mandataire sortant.

Comme cet identifiant SIP/SigComp "Outbound-id" d'URI correspond à celui du compartiment créé auparavant, ce compartiment est utilisé pour compresser la demande INVITE.

À réception de la demande INVITE (5), le mandataire sortant enregistre les chemins et relaie la demande INVITE (6) vers l'avant. Le mandataire sortant enregistre les chemins pour s'assurer que tous les messages SIP relatifs à ce nouveau dialogue sont acheminés à travers le mandataire sortant.

Finalement, le dialogue se termine par une transaction BYE (11) qui traverse aussi le mandataire sortant.

15. Considérations sur la sécurité

Les mêmes considérations de sécurité que décrites dans la [RFC3320] s'appliquent au présent document. Noter que garder des états SigComp plus longtemps que la durée d'un dialogue SIP ne devrait pas faire peser de nouveaux risques pour la sécurité parce que la création de l'état a été permise en premier.

16. Considérations relatives à l'IANA

L'IANA a enregistré le paramètre de champ d'en-tête Via "sigcomp-id", qui est défini au paragraphe 9.1, dans le sous registre Paramètres de champ d'en-tête et valeurs de paramètres au sein du registre des paramètres SIP :

Champ d'en-tête	Nom de paramètre prédéfini	Valeurs	Référence
Via	sigcomp-id	aucune	[RFC5049]

L'IANA a enregistré le paramètre d'URI SIP "sigcomp-id", qui est défini au paragraphe 9.1, dans le sous registre Paramètres d'URI SIP/SIPS au sein du registre des paramètres SIP :

Nom de paramètre	Valeurs prédéfinies	Référence
sigcomp-id	aucune	[RFC5049]

17. Remerciements

Les auteurs tiennent à remercier les personnes suivantes de leurs commentaires et suggestions : Jan Christoffersson, Joerg Ott, Mark West, Pekka Pessi, Robert Sugar, Jonathan Rosenberg, Robert Sparks, Juergen Schoenwaelder, et Tuukka Karvonen. Abigail Surtees et Adam Roach ont effectué une relecture serrée de ce document.

18. Références

18.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2141] R. Moats, "Syntaxe des URN", mai 1997. *(Obsolète, voir* RFC8141)

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760)

[RFC3308] P. Calhoun et autres, "Extensions de services différenciés du protocole de tunnelage de couche deux (L2TP)", novembre 2002. *(P.S.)*

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

[RFC3327] D. Willis, B. Hoeneisen, "Champ d'en-tête d'extension du protocole d'initialisation de session (SIP) pour enregistrer des contacts non adjacents", décembre 2002. *(P.S.)*

[RFC3485] M. Garcia-Martin et autres,"Dictionnaire statique du protocole d'initialisation de session (SIP) et du protocole de description de session (SDP) pour la compression de signaux (SigComp)", février 2003. (*MàJ par* RFC4896) *(P.S.)*

[RFC3486] G. Camarillo, "Compression du protocole d'initialisation de session (SIP)", février 2003. (*MàJ par* RFC5049) *(P.S.)*

[RFC4077] A.B. Roach, "Mécanisme d'accusé de réception négatif pour la compression de signalisation", mai 2005. *(P.S.)*

[RFC4122] P. Leach et autres, "Espace de noms d'URN d'identifiant univoque universel (UUID)", juillet 2005. *(P.S.)*

[RFC4234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", octobre 2005. *(Remplace* RFC2234, *remplacée par* RFC5234)

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4896] A. Surtees et autres, "Corrections et précisions à la compression de signalisation (SigComp)", juin 2007. (*MàJ* RFC3320, RFC3321, RFC3485) *(P.S.)*

18.2 Références pour information

[RFC5626] C. Jennings, R. Mahy, F. Audet, éd., "Gestion des connexions initiées par le client dans le protocole d'initialisation de session (SIP)", octobre 2009. *(MàJ* RFC3261, RFC3327) *(P.S.)*

[RFC6080] D. Petrie, S. Channabasappa, éd.. "Cadre pour la livraison de profil d'agent d'utilisateur du protocole d'initialisation de session", mars 2011. *(P.S.)*

Adresse des auteurs

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28334
Germany
mél : cabo@tzi.org

Zhigang Liu
Nokia Research Center
955 Page Mill Road
Palo Alto, CA 94304
USA
mél : zhigang.c.liu@nokia.com

Richard Price
EADS Defence and Security Systems Limited
Meadows Road
Queensway Meadows
Newport, Gwent NP19 4SS
mél : richard.price@eads.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland
mél : Gonzalo.Camarillo@ericsson.com

Déclaration complète de droits de reproduction

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 le 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.