

Groupe de travail Réseau
Request for Comments : 3486
 Catégorie : En cours de normalisation

G. Camarillo, Ericsson
 février 2003
 Traduction Claude Brière de L'Isle

Compression du protocole d'initialisation de session (SIP)

Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de 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 (2003). Tous droits réservés

Résumé

Le présent document décrit un mécanisme pour signaler que la compression est désirée pour un ou plusieurs messages du protocole d'initialisation de session (SIP). Il indique aussi quand il est approprié d'envoyer des messages SIP compressés à une entité SIP.

Terminologie

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMETE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Table des Matières

1. Introduction.....	1
2. Vue d'ensemble du fonctionnement.....	2
3. Mises en œuvre de SigComp pour SIP.....	2
4. Envoi d'une demande à un serveur.....	2
4.1 Obtention d'un URI SIP ou SIPS avec comp=sigcomp.....	3
5. Envoi d'une réponse à un client.....	3
6. Double Record-Routing.....	4
7. Situations d'erreur.....	4
8. BNF augmenté.....	4
9. Exemple.....	4
10. Considérations pour la sécurité.....	6
11. Considérations relatives à l'IANA.....	6
12. Remerciements.....	7
13. Références.....	7
14. Adresse de l'auteur.....	7
15. Déclaration complète de droits de reproduction.....	7

1. Introduction

Un client SIP [RFC3261] qui envoie une demande à un serveur SIP effectue normalement une recherche du nom de domaine du serveur dans le DNS. Lorsque des enregistrements NAPTR [RFC3403] ou SRV [RFC2782] sont disponibles pour le serveur, le client peut spécifier le type de service qu'il veut. Le service dans ce contexte est le protocole de transport à utiliser par SIP (par exemple, UDP, TCP ou SCTP). Un serveur SIP qui prend en charge, par exemple, trois protocoles de transport différents, va avoir trois entrées DNS différentes.

Comme il est prévu que le nombre de protocoles de transport pris en charge par un protocole particulier de couche d'application ne doive pas augmenter énormément, avoir une entrée du DNS par transport semble une solution assez adaptable.

Cependant, il est parfois nécessaire d'inclure de nouvelles couches entre le protocole de transport et le protocole de couche application. Des exemples de ces couches sont la sécurité et la compression de la couche transport. Si le DNS était utilisé pour découvrir la disponibilité de ces couches pour un serveur particulier, le nombre d'entrées du DNS nécessaires pour ce serveur augmenterait énormément.

Un serveur qui, par exemple, prend en charge TCP et SCTP comme transports, TLS pour la sécurité du transport et SigComp pour la compression de signalisation, aurait besoin des huit entrées au DNS énumérées ci-dessous :

1. TCP, pas de sécurité, pas de compression
2. TCP, pas de sécurité, SigComp
3. TCP, TLS, pas de compression
4. TCP, TLS, SigComp
5. SCTP, pas de sécurité, pas de compression
6. SCTP, pas de sécurité, SigComp
7. SCTP, TLS, pas de compression
8. SCTP, TLS, SigComp

Il est clair que cette façon d'utiliser le DNS n'est pas adaptable. Il faut donc un mécanisme de couche application pour exprimer la prise en charge de la compression de signalisation.

Noter que pour des raisons historiques, HTTP et SIP utilisent tous deux un accès différent pour TLS par dessus TCP et pour TCP seul, bien qu'à présent, cette solution ne soit plus considérée comme adaptée.

Un élément SIP qui prend en charge la compression va avoir besoin d'être prêt à recevoir des messages compressés et non compressés sur le même accès. Il va effectuer le démultiplexage sur la base du mouchard dans les bits de poids fort de chaque message compressé.

2. Vue d'ensemble du fonctionnement

Il y a deux types de message SIP ; les demandes SIP et les réponses SIP. Les clients envoient des demandes SIP à la partie hôte d'un URI et les serveurs envoient des réponses à la partie hôte dans le paramètre sent-by du champ d'en-tête Via.

On définit deux paramètres, un pour les URI SIP et l'autre pour le champ d'en-tête Via. Le format des deux paramètres est le même, comme le montrent les exemples ci-dessous :

```
sip:alice@atlanta.com;comp=sigcomp  
Via: SIP/2.0/UDP server1.foo.com:5060;branch=z9hG4bK87a7;comp=sigcomp
```

La présence de ce paramètre (comp=sigcomp) dans un URI indique que la demande doit être compressée en utilisant SigComp, comme défini dans la [RFC3320]. La présence de comp=sigcomp dans un champ d'en-tête Via indique que la réponse doit être compressée en utilisant SigComp.

Donc, la présence de comp=sigcomp indique que l'entité SIP identifiée par l'URI ou par le champ d'en-tête Via prend en charge SigComp et veut recevoir des messages compressés. Avoir comp=sigcomp signifie "être d'accord" aussi bien que "prise en charge" permet au receveur d'un message SIP d'influencer la décision d'utiliser ou non SigComp à un moment donné.

3. Mises en œuvre de SigComp pour SIP

Toute mise en œuvre de SIP qui prend en charge SigComp DOIT mettre en œuvre les procédures décrites dans le présent document.

4. Envoi d'une demande à un serveur

Une demande est envoyée à la partie hôte d'un URI. Cet URI, qu'on appelle l'URI de prochain bond, est l'URI de demande de la demande ou une entrée dans le champ d'en-tête Route.

Si l'URI de prochain bond contient le paramètre comp=sigcomp, le client DEVRAIT compresser la demande en utilisant SigComp comme défini dans la [RFC3320].

Si l'URI de prochain bond est un URI SIPS, la demande DEVRAIT être compressée avant d'être passée à la couche TLS.

Un client NE DOIT PAS envoyer une demande compressée à un serveur si il ne sait pas si le serveur prend ou non en charge SigComp.

Indépendamment de la question de savoir si la demande est envoyée compressée ou non, si un client veut recevoir compressées les demandes ultérieures au sein du même dialogue dans la direction UAS->UAC, ce client DEVRAIT ajouter le paramètre `comp=sigcomp` à l'URI dans le champ d'en-tête Contact si il est un client d'agent d'utilisateur. Si le client est un mandataire, il DEVRAIT ajouter le paramètre `comp=sigcomp` à son URI dans le champ d'en-tête Record-Route.

Si un client d'agent d'utilisateur envoie une demande compressée, il DEVRAIT ajouter le paramètre `comp=sigcomp` à l'URI dans le champ d'en-tête Contact. Si un mandataire qui fait des Record-Route envoie une demande compressée, il DEVRAIT ajouter `comp=sigcomp` à son URI dans le champ d'en-tête Record-Route.

Si un client envoie une demande compressée, il DEVRAIT ajouter le paramètre `comp=sigcomp` à l'entrée supérieure du champ d'en-tête Via.

Si un client ne sait pas si le serveur prend ou non en charge SigComp, mais au cas où le serveur le prendrait en charge, il aimerait recevoir des réponses compressées, ce client DEVRAIT ajouter le paramètre `comp=sigcomp` à la première entrée du champ d'en-tête Via. Cependant, comme on l'a noté plus haut, la demande ne sera pas compressée.

4.1 Obtention d'un URI SIP ou SIPS avec `comp=sigcomp`

Pour les demandes au sein d'un dialogue, un URI de prochain bond avec le paramètre `comp=sigcomp` est obtenu d'un champ d'en-tête Record-Route lorsque le dialogue est établi. Un client qui envoie une demande en dehors d'un dialogue peut aussi obtenir des URI SIP avec `comp=sigcomp` dans un champ d'en-tête Contact dans une réponse 3xx ou 485 à la demande.

Cependant, les clients qui établissent une session ne vont normalement pas vouloir attendre que le dialogue soit établi pour commencer à compresser les messages. Un des plus gros gains apportés par SigComp à SIP est la capacité à compresser le INVITE initial d'un dialogue, lorsque l'utilisateur attend que la session soit établie. Donc, les clients ont besoin d'un moyen pour obtenir un URI `comp=sigcomp` de leur mandataire de sortie avant que l'utilisateur décide d'établir une session.

Une solution à ce problème est la configuration manuelle. Cependant, il est parfois nécessaire d'avoir des clients configurés de façon automatique. Malheureusement, les mécanismes actuels de configuration de client SIP (par exemple, en utilisant DHCP [RFC3319]) ne permettent pas de fournir au client les paramètres d'URI. Dans ce cas, le client DEVRAIT envoyer une demande OPTIONS non compressée à son mandataire de sortie. Le mandataire de sortie peut fournir un URI SIP de remplacement avec le paramètre `comp=sigcomp` dans un champ d'en-tête Contact dans une réponse 200 OK à OPTIONS. Le client peut utiliser cet URI pour les demandes suivantes qui sont envoyées par le même mandataire de sortie en utilisant la compression.

La [RFC3261] ne définit pas comment un mandataire devrait répondre à une demande OPTIONS adressée à lui-même. Elle décrit seulement comment les serveurs répondent aux OPTIONS adressés à un usager particulier. Le paragraphe 11.2 de la RFC3261 dit : "Les champs d'en-tête Contact PEUVENT être présents dans une réponse 200 (OK) et avoir la même sémantique qu'une réponse 3xx. C'est-à-dire qu'ils peuvent faire la liste d'un ensemble de noms de remplacement et des méthodes pour atteindre l'utilisateur."

On étend ce comportement aux serveurs mandataires qui répondent aux OPTIONS qui leur sont adressés. Il PEUVENT faire la liste d'un ensemble d'URI de remplacement pour contacter le mandataire.

Noter que recevoir des demandes entrantes (même des INVITE initiales) compressées n'est pas un problème, car les agents d'utilisateur peuvent enregistrer (avec REGISTER) un URI SIP avec `comp=sigcomp` chez leur registraire. Toutes les demandes entrantes pour l'utilisateur seront envoyées à cet URI SIP en utilisant la compression.

5. Envoi d'une réponse à un client

Une réponse est envoyée à l'hôte dans le paramètre sent-by du champ d'en-tête Via. Si le premier champ d'en-tête Via contient le paramètre `comp=sigcomp`, la réponse DEVRAIT être compressée. Autrement, la réponse NE DOIT PAS être compressée.

Afin d'éviter une compression asymétrique (c'est-à-dire, deux entités SIP échangent des demandes compressées dans une direction et des demandes non compressées dans l'autre direction) les mandataires doivent réécrire leurs entrées Record-Route dans les réponses. Un mandataire qui effectue un Record-Route inspecte le champ d'en-tête Record-Route dans la réponse et le champ d'en-tête Contact dans la demande qui a déclenché cette réponse (voir l'exemple de la Section 9). Il cherche l'URI du prochain bond vers l'amont (plus proche du client d'agent d'utilisateur) sur le chemin. Si cet URI contient le paramètre `comp=sigcomp`, le mandataire DEVRAIT ajouter `comp=sigcomp` à son entrée dans le champ d'en-

tête Record-Route. Si cet URI ne contient pas le paramètre `comp=sigcomp`, le mandataire DEVRAIT retirer `comp=sigcomp` (s'il est présent) de son entrée dans le champ d'en-tête Record-Route.

De la même façon, un serveur d'agent d'utilisateur DEVRAIT ajouter `comp=sigcomp` au champ d'en-tête Contact de la réponse si l'URI du prochain bond vers l'amont dans le chemin contient le paramètre `comp=sigcomp`.

6. Double Record-Routing

Bien que les mandataires ajoutent habituellement zéro ou une entrée Record-Route à une demande particulière, certains mandataires en ajoutent deux pour éviter de réécrire le Record-Route. Un exemple typique de double Record-Routing est celui d'un mandataire SIP qui agit comme pare-feu entre deux réseaux. Selon le réseau d'où provient la demande, elle sera reçue sur une interface différente par le mandataire. Le mandataire ajoute une entrée Record-Route pour une interface et une seconde pour l'autre interface. De cette façon, le mandataire n'a pas besoin de réécrire le champ d'en-tête Record-Route dans la réponse.

Les mandataires qui reçoivent des messages compressés d'un côté du dialogue (par exemple, de l'amont) et des messages non compressés de l'autre côté (par exemple, de l'aval) PEUVENT utiliser le mécanisme décrit ci-dessus.

Si un mandataire détecte que le mandataire de prochain bond pour une demande est ce mandataire lui-même et que la demande ne sera pas envoyée sur le réseau, ce mandataire PEUT choisir de ne pas compresser la demande même si l'URI contient le paramètre `comp=sigcomp`.

7. Situations d'erreur

Si une demande SIP compressée arrive à un serveur SIP qui ne comprend pas SigComp, le serveur n'aura aucun moyen d'indiquer l'erreur au client. Le message sera impossible à analyser, et il n'y aura pas de champ d'en-tête Via pour indiquer une adresse où envoyer une réponse d'erreur.

Si un client SIP envoie une demande compressée et si la transaction du client arrive à expiration sans avoir reçu de réponse, le client DEVRAIT réessayer la même demande sans utiliser la compression. Si la demande compressée était envoyée sur une connexion TCP, le client DEVRAIT fermer cette connexion et en ouvrir une nouvelle pour envoyer la demande non compressée. Autrement, le serveur ne serait pas capable de détecter le début du nouveau message.

8. BNF augmenté

Cette section donne la forme Backus-Naur augmenté (BNF) des deux paramètres décrits ci-dessus.

Le paramètre Compression d'URI est un "uri-parameter", comme défini par l'ABNF SIP (paragraphe 25.1 de la [RFC3261]):

```
compression-param = "comp=" ("sigcomp" / other-compression)
other-compression = jeton
```

Le paramètre Via compression est un "via-extension", comme défini par l'ABNF SIP (paragraphe 25.1 de la [RFC3261]):

```
via-compression = "comp" EQUAL ("sigcomp" / other-compression)
other-compression = jeton
```

9. Exemple

L'exemple suivant illustre l'utilisation des paramètres définis ci-dessus. Le flux d'appels de la Figure 1 montre une prise de contact INVITE-200 OK-ACK entre un UAC et un UAS à travers deux mandataires. Le mandataire P1 ne fait pas de Record-Route mais le mandataire P2 le fait. Les deux mandataires prennent en charge la compression, mais ils ne l'utilisent pas par défaut.

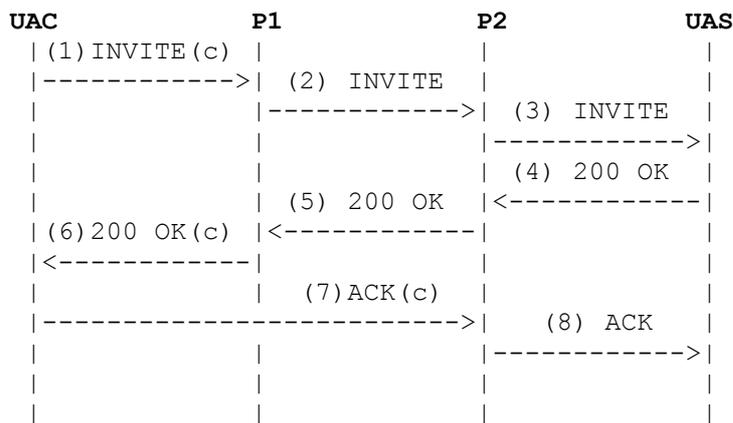


Figure 1 : Transaction INVITE à travers deux mandataires

Les messages (1), (6) et (7) sont compressés (c).

On donne une description partielle des messages impliqués dans le flux d'appels ci-dessous. Seules certaines parties de chaque message sont montrées, à savoir les champs d'en-tête Nom de la méthode, URI de demande, Via, Route, Record-Route et Contact. On n'a pas utilisé un format correct pour ces champs d'en-tête. On s'est plutôt concentré sur le contenu des champs d'en-tête et sur la présence (ou l'absence) du paramètre "comp=sigcomp".

(1) INVITE UAS

Via: UAC;comp=sigcomp
Route: P1;comp=sigcomp
Contact: UAC;comp=sigcomp

P1 est le mandataire de sortie de l'UAC, et il prend en charge SigComp. L'UAC est configuré pour envoyer du trafic compressé à P1, et donc, il compresse le INVITE (1). De plus, l'UAC veut recevoir les demandes et réponses futures compressées pour ce dialogue. Donc, il ajoute le paramètre comp=Sigcomp aux champs d'en-tête Via et Contact.

(2) INVITE UAS

Via: P1
Via: UAC;comp=sigcomp
Route: P2
Contact: UAC;comp=sigcomp

P1 transmet le INVITE (2) à P2. P1 n'utilise pas la compression par défaut, de sorte qu'il envoie le INVITE non compressé à P2.

(3) INVITE UAS

Via: P2
Via: P1
Via: UAC;comp=sigcomp
Record-Route: P2
Contact: UAC;comp=sigcomp

P2 transmet l'INVITE (3) à l'UAS. P2 prend en charge la compression, mais il ne l'utilise pas par défaut. Donc, il envoie l'INVITE non compressé. P2 souhaite rester dans le chemin de signalisation et il fait donc des Record-Route.

(4) 200 OK

Via: P2
Via: P1
Via: UAC;comp=sigcomp
Record-Route: P2
Contact: UAS

L'UAS génère une réponse 200 OK et l'envoie à l'hôte dans le premier Via, qui est P2.

(5) 200 OK

Via: P1

Via: UAC;comp=sigcomp
 Record-Route: P2;comp=sigcomp
 Contact: UAS

P2 reçoit la réponse 200 OK. P2 a enregistré les chemins (Record-Route) donc il inspecte l'ensemble des chemins pour ce dialogue. Pour les demandes provenant de l'UAS vers l'UAC (la direction opposée à celle du premier INVITE) le prochain bond sera le champ d'en-tête Contact de l'INVITE, parce que P1 n'enregistre pas les chemins. Ce Contact identifiait l'UAC:

Contact: UAC;comp=sigcomp

Comme l'UAC veut recevoir des demandes compressées (Contact de l'INVITE) P2 suppose que l'UAC aimerait aussi envoyer des demandes compressées (Record-Route de la 200 OK). Donc, P2 modifie son entrée dans le champ d'en-tête Record-Route de la réponse 200 OK (5). Dans l'INVITE (3), P2 n'a pas utilisé le paramètre comp=sigcomp. Il l'ajoute maintenant dans la 200 OK (5). Cela va permettre à l'UAC d'envoyer des demandes compressées au sein de ce dialogue.

(6) 200 OK
 Via: UAC;comp=sigcomp
 Record-Route: P2;comp=sigcomp
 Contact: UAS

P1 envoie la 200 OK (6) compressée à l'UAC parce que le champ d'en-tête Via contenait le paramètre comp=sigcomp.

(7) ACK UAS
 Via: UAC;comp=sigcomp
 Route: P2;comp=sigcomp
 Contact: UAC;comp=sigcomp

L'UAC envoie le ACK (7) compressé directement à P2 (P1 n'a pas fait de Record-Route).

(8) ACK UAS
 Via: P2
 Via: UAC;comp=sigcomp
 Contact: UAC;comp=sigcomp

P2 envoie le ACK (8) non compressé à l'UAS.

10. Considérations pour la sécurité

Une entité SIP qui reçoit un message compressé doit le décompresser et l'analyser. Cela exige un petit peu plus de puissance de traitement que de seulement analyser un message. Cela implique qu'une attaque de déni de service utilisant des messages compressés serait légèrement pire qu'une attaque avec des messages non compressés.

Un attaquant qui insère le paramètre comp=sigcomp dans un message SIP pourrait faire qu'une entité SIP envoie des messages compressés à une autre entité SIP qui ne prend pas en charge SigComp. Des mécanismes d'intégrité appropriés devraient être utilisés pour éviter cette attaque.

11. Considérations relatives à l'IANA

Le présent document définit le paramètre d'URI "comp" et l'extension via. De nouvelles valeurs pour "comp" seront enregistrées par l'IANA à <http://www.iana.org/assignments/sip-parameters> lorsque de nouveaux schémas de compression de signalisation seront publiés dans des RFC en cours de normalisation. La section Considérations relatives à l'IANA de la RFC DOIVENT inclure les informations suivantes, qui apparaissent dans le registre de l'IANA avec le numéro de RFC de la publication.

- o Nom du schéma de compression.
- o Valeur du jeton à utiliser. Le jeton PEUT être d'une longueur quelconque, mais DEVRAIT ne pas être de plus de dix caractères.

La seule entrée dans le registre est pour l'instant :

Schéma de compression	Jeton	Référence
Signaling Compression	sigcomp	RFC3486

12. Remerciements

Allison Mankin, Jonathan Rosenberg et Miguel Angel Garcia-Martin ont fourni des commentaires précieux sur le présent mémoire.

13. Références

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "RR DNS pour la spécification de la [localisation des services](#) (DNS SRV)", février 2000.
- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (*Mise à jour par [RFC3265](#), [RFC3853](#), [RFC4320](#), [RFC4916](#), [RFC5393](#), [RFC6665](#)*)
- [RFC3319] H. Schulzrinne, B. Volz, "[Options du protocole de configuration dynamique d'hôte](#) (DHCPv6) pour serveurs de protocole d'initialisation de session (SIP)", juillet 2003. (*P.S.*)
- [RFC3320] R. Price, et autres, "[Compression de signalisation](#) (SigComp)", janvier 2003. (*MàJ par [RFC4896](#)*) (*P.S.*)
- [RFC3403] M. Mealling, "Système de découverte dynamique de délégation ([DDDS](#)) [Partie III : base de données du système](#) de noms de domaines (DNS)", octobre 2002. (*P.S.*)

14. Adresse de l'auteur

Gonzalo Camarillo
Ericsson
Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland
mél : Gonzalo.Camarillo@ericsson.com

15. 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 y 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.

Remerciement

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