Compressing the Session Initiation Protocol

STATUS OF THIS MEMO

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

Abstract

This document describes a mechanism to signal that compression is desired for one or more SIP messages. It also states when it is appropriate to send compressed SIP messages to a SIP entity.
Table of Contents

1 Introduction ........................................ 3
2 Overview of operation ............................. 4
3 Sending a Request to a Server ................. 4
4 Sending a Response to a Client ............... 5
5 Error Situations .................................. 5
6 Augmented BNF .................................. 6
7 Example ......................................... 6
8 Security Considerations ....................... 9
9 Acknowledges ..................................... 9
10 Authors' Addresses ............................... 9
11 Normative References ......................... 10
12 Informative References ......................... 10
1 Introduction

A SIP [1] client sending a request to a SIP server typically performs a DNS lookup for the domain name of the server. When NAPTR [3] or SRV [4] records are available for the server, the client can specify the type of service it wants. The service in this context is the transport protocol to be used by SIP (e.g., UDP, TCP or SCTP). A SIP server that support, for instance, three different transport protocols, will have three different DNS entries.

Since it is foreseen that the number of transport protocols supported by a particular application layer protocol is not going to grow dramatically, having a DNS entry per transport seems like a scalable enough solution.

However, sometimes it is necessary to include new layers between the transport protocol and the application layer protocol. Examples of these layers are transport layer security and compression. If DNS was used to discover the availability of these layers for a particular server, the number of DNS entries needed for that server would grow dramatically.

A server that, for example, supported TCP and SCTP as transports, TLS for transport security and Sigcomp for signaling compression, would need the 8 DNS entries listed below:

1. TCP, no security, no compression
2. TCP, no security, SigComp
3. TCP, TLS, no compression
4. TCP, TLS, SigComp
5. SCTP, no security, no compression
6. SCTP, no security, SigComp
7. SCTP, TLS, no compression
8. SCTP, TLS, SigComp

It is clear that this way of using DNS is not scalable. Therefore, an application layer mechanism to express support of signalling compression is needed.

Note that for historical reasons both HTTP and SIP use a different port for TLS on top of TCP than for TCP alone,
although at present this solution is not considered scalable any longer.

A SIP element that supports compression will need to be prepared to receive compressed and uncompressed messages on the same port. It will perform demultiplexing based on the cookie in the topmost bits of every compressed message.

2 Overview of operation

There are two types of SIP messages; SIP requests and SIP responses. Clients send SIP requests to the host part of a URI and servers send responses to the host in the sent-by parameter of the Via header field.

We define two parameters, one for SIP URIs and the other for the Via header field. The format of both parameters is the same, as shown in the examples below:

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

The presence of this parameter in a URI indicates that the request has to be compressed using Sigcomp, as defined in [2]. The presence of comp=sigcomp in a Via header field indicates that the response has to be compressed using Sigcomp.

Note that, as stated in [2], every implementation that supports Sigcomp needs to implement the static dictionary for SIP and SDP.

3 Sending a Request to a Server

A request is sent to the host part of a URI. This URI, referred to as the next-hop URI, is the Request-URI of the request or an entry in the Route header field.

If the next-hop URI contains the parameter comp=sigcomp, the client SHOULD compress the request using Sigcomp as defined in [2]. The next-hop URI can contain such a parameter due to a number of reasons:

- The URI was obtained from a Record-Route header field when the dialog was established.

- The URI was previously registered in a registrar using a REGISTER request.
The URI was obtained from a Contact header field in a 3xx or 485 response to the request.

The URI was obtained out of band or statically configured.

A client MUST NOT send a compressed request to a server if it does not know whether or not the server supports Sigcomp.

If a client does not know whether or not the server supports Sigcomp, but in case the server supported it, it would like to receive compressed responses, this client SHOULD add the parameter comp=sigcomp to the topmost entry of the Via header field. The request, however, as stated above, will not be compressed.

Regardless of whether the request is sent compressed or not, if a client would like to receive subsequent requests within the same dialog compressed, this client SHOULD add the parameter comp=sigcomp to the URI in the Contact header field if it is a user agent client. If the client is a proxy, it SHOULD add the parameter comp=sigcomp to its URI in the Record-Route header field.

4 Sending a Response to a Client

A response is sent to the host in the sent-by parameter of the Via header field. If the topmost Via header field contains the parameter comp=sigcomp, the response SHOULD be compressed. Otherwise, the response MUST NOT be compressed.

A proxy performing Record-Route inspects the Record-Route header field in the response and the Contact header field in the request that triggered this response (see example in Section 7). It looks for the URI of the next upstream (closer to the user agent client) hop in the route set. If this URI contains the parameter comp=sigcomp, the proxy SHOULD add comp=sigcomp to its entry in the Record-Route header field.

The same way, a user agent server SHOULD add comp=sigcomp to the Contact header field of the response if the URI of the next upstream hop in the route set contained the parameter comp=sigcomp.

5 Error Situations

If a compressed SIP request arrives to a SIP server that does not understand SigComp, the server will not have any means to indicate the error to the client. The message will be impossible to parse, and there will be no Via header field indicating an address to send the error response.
If a SIP client sends a compressed request and the client transaction times out without having received any response, the client SHOULD retry the same request without using compression. If the compressed request was sent over a TCP connection, the client SHOULD close that connection and open a new one to send the uncompressed request. Otherwise the server would not be able to detect the beginning of the new message.

6 Augmented BNF

This section provides the augmented Backus-Naur Form (BNF) of both parameters described above.

The compression URI parameter is a "uri-parameter", as defined by the SIP ABNF (Section 25.1 of [1]):

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

The Via compression parameter is a "via-extension", as defined by the SIP ABNF (Section 25.1 of [1]):

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

7 Example

The following example illustrates the use of the parameters defined above. The call flow of Figure 1 shows an INVITE-200 OK-ACK handshake between a UAC and a UAS through two proxies. Proxy P1 does not Record-Route but proxy P2 does. Both proxies support compression, but they do not use it by default.

<table>
<thead>
<tr>
<th>UAC</th>
<th>P1</th>
<th>P2</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) INVITE(c)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------&gt;</td>
<td>(2) INVITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------&gt;</td>
<td>(3) INVITE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>--------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(4) 200 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(5) 200 OK</td>
<td>&lt;--------------</td>
</tr>
<tr>
<td>(6) 200 OK(c)</td>
<td>&lt;--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--------------</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 1: INVITE transaction through two proxies

Messages (1), (6) and (7) are compressed (c).

We provide a partial description of the messages involved in this call flow below. Only some parts of each message are shown, namely the Method name, the Request-URI and the Via, Route, Record-Route and Contact header fields. We have not used a correct format for these header fields. We have rather focus on the contents of the header fields and on the presence (or absence) of the "comp=sigcomp" parameter.

(1) INVITE UAS
   Via: UAC;comp=sigcomp
   Route:P1;comp=sigcomp
   Contact: UAC;comp=sigcomp

P1 is the outbound proxy of the UAC, and it supports Sigcomp. The UAC is configured to send compressed traffic to P1, and therefore, it compresses the INVITE (1). In addition, the UAC wants to receive future requests and responses for this dialog compressed. Therefore, it adds the comp=Sigcomp parameter to the Via and to the Contact header fields.

(2) INVITE UAS
   Via: P1
   Via: UAC;comp=sigcomp
   Route:P2
   Contact: UAC;comp=sigcomp

P1 forwards the INVITE (2) to P2. P1 does not use compression by default, so it sends the INVITE uncompressed to P2.

(3) INVITE UAS
   Via: P2
P2 forwards the INVITE (3) to the UAS. P2 supports compression, but it does not use it by default. Therefore, it sends the INVITE uncompressed. P2 wishes to remain in the signalling path and therefore it Record-Routes.

(4) 200 OK
   Via: P2
   Via: P1
   Via: UAC;comp=sigcomp
   Record-Route:P2
   Contact: UAS

The UAS generates a 200 OK response and sends it to the host in the topmost Via, which is P2.

(5) 200 OK
   Via: P1
   Via: UAC;comp=sigcomp
   Record-Route:P2;comp=sigcomp
   Contact: UAS

P2 receives the 200 OK response. P2 Record-Routed, so it inspects the Route set for this dialog. For requests from the UAS towards the UAC (the opposite direction than the first INVITE), the next hop will be the Contact header field of the INVITE, because P1 did not Record-Route. That Contact identified the UAC:

   Contact: UAC;comp=sigcomp

Since the UAC wants to receive compressed requests (Contact of the INVITE), P2 assumes that the UAC would also like to send compressed requests (Record-Route of the 200 OK). Therefore, P2 modifies its entry in the Record-Route header field of the 200 OK (5). In the INVITE (3), P2 did not used the comp=sigcomp parameter. Now it adds it in the 200 OK (5). This will allow the UAC sending compressed requests within this dialog.
(6) 200 OK
   Via: UAC;comp=sigcomp
   Record-Route:P2;comp=sigcomp
   Contact: UAS

P1 sends the 200 OK (6) compressed to the UAC because the Via header
field contained the comp=Sigcomp parameter.

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

The UAC sends the ACK (7) compressed directly to P2 (P1 did not
Record-Route).

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

P2 sends the ACK (8) uncompressed to the UAS.

8 Security Considerations

A SIP entity receiving a compressed message has to decompress it and
to parse it. This requires slightly more processing power than only
parsing a message. This implies that a denial of service attack using
compressed messages would be slightly worse than an attack with
uncompressed messages.

An attacker inserting the parameter comp=sigcomp in a SIP message
could make a SIP entity send compressed messages to another SIP
tentity that did not support Sigcomp. Appropriate integrity mechanisms
should be used to avoid this attack.

9 Acknowledges

Jonathan Rosenberg and Miguel Angel Garcia provided valuable comments
on this memo.

10 Authors' Addresses
Normative References


Informative References


Full Copyright Statement

Copyright (c) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.
This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.