**Proceedings of the Fourth IASTED Asian Conference**
**COMMUNICATION SYSTEMS AND NETWORKS**
**April 2–4, 2007, Phuket, Thailand**
**ISBN CD: 978-0-88986-658-4**

# PERFORMANCE ANALYSIS OF SIGCOMP FOR 3G CELLULAR NETWORKS

Woojin Han[3] , Kyung-jun Lee, Hyun-hwa Park[2], Kyung-geun Lee[1], and Juwook Jang
Dept. of Electronic Engineering, Sogang University, Seoul, Korea
[1]Dept. of Information and Communication Engineering, Sejong University, Seoul, Korea
[2]SK Telecom, Seoul, Korea
[3] NEX1FUTURE Co.,Ltd., Mabuk-dong, Giheung-gu, Yongin-City, Gyeonggi-do, Korea
{ghost, jjang} @eeca1.sogang.ac.kr, kglee@sejong.ac.kr[1], hhpark@sktelecom.com[2], wjhan@nex1.co.kr[3]

**ABSTRACT**

This paper demonstrates the performance of signaling compression (SigComp) used in 3G cellular IP Multimedia Subsystems (IMS). To analyze the performance of SigComp for 3G cellular networks, the measurements are performed for different SIP sessions including typical 3GPP scenarios. In this paper, the achievable gain of SigComp is demonstrated through performance results. It is estimated that the one-way RAN delay over the cellular network can be reduced by approximately 75% compared with using uncompressed SIP messages. Also, the size of message related to the performance can be decreased by 86%. Our results show that SigComp is mandatory to maximize the transport efficiency for SIP messages over the radio interface.

**KEY WORDS**

Wireless Networks Standards and Protocols, SigComp (Signaling Compression), 3G wireless networks, SIP (Session Initiation Protocol)

## 1. Introduction

Session Initiation Protocol (SIP) is signaling protocol used for call control in the third generation mobile network beginning from the Third Generation Partnership Project (3GPP) release 5. Because of the convergence of cellular systems to an All-IP network, SIP has been selected as the signaling protocol for multimedia session control. However, the use of SIP for session establishment in narrow bandwidth links such as radio interfaces or low speed serial links may lead to unnecessarily long call set-up times. In order to reduce the call set-up times, SigComp [1] has been developed by the Internet Engineering Task Force (IETF). SigComp provides not only a method to reduce the SIP message sizes, but also decompression functionalities for a wide variety of compression algorithms.

There are a few research that have been accomplished on SigComp and they have just focused on estimating the achievable compression ratios on the prototype implementation [2], [3]. However, in this paper, we demonstrate the performance of SigComp for 3G cellular networks based on the well implemented SigComp module. Our results in the figure 3 below, show that SigComp can reduce the one-way Radio Access Network (RAN) delay from 6.86s to 1.74 maximum. Further, it is mentioned that the compressibility of different SIP message sequences is analyzed.

The rest of this paper is organized as follows. In chapter 2, we briefly describe the SigComp concept as well as Efficient Protocol Independent Compression (EPIC) [4] and how dictionaries and extended operations can be used to improve compression efficiency. In chapter 3, we demonstrate the performance analysis of SigComp based on SIP. In chapter 4, analysis of the SigComp performance in 3G cellular networks such as one-way Radio Access Network (RAN) delay and compressibility of different SIP message sequences is presented. We conclude the paper with chapter 5.

## 2. Signaling Compression (SigComp)

The compression scheme SigComp, seen in Figure 1, which is provided as a layer between SIP and the underlying transport layer protocols, has been designed as a generic compression scheme applicable to a variety of protocols. SigComp has been positioned as an independent component, which can be integrated with any protocol stack.

### 2.1 Architecture

The kernel of SigComp is the Universal Decompressor Virtual Machine (UDVM). The UDVM provides decompression functionalities and it is a virtual machine much like the Java Virtual Machine, but it has been optimized for running decompression algorithms. When the SigComp message is received, the message is placed in the decompression memory. The bytecode and the dictionary, saved at the decompressor entity as states, are loaded into the UDVM and finally, the UDVM starts executing. After a new message is decompressed, the

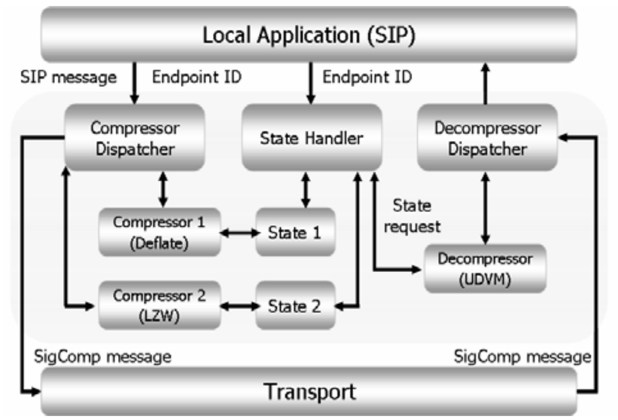


**Figure 1. The structure of SigComp.**

information it contains is used to update the dictionary, which then is saved as a new state.

When the UDVM has been initialized, it can receive additional compressed data from the decompressor dispatcher or state information from the state handler on demand. As the decompression process expires, the UDVM indicates this to the decompressor dispatcher, which provides it with a compartment identifier. This identifier is passed to the state handler for state creation request. The state handler uses the compartment identifier to store the state information. Especially, with the use of SigComp – Extended Operations, information form either sent or received messages can be utilized to update the dictionary. For compression of SIP, adding sent or received messages to the dictionary dramatically improve the compression efficiency.

There are 36 instructions supported by the UDVM, offering various compression algorithms with the minimum possible overhead. The compressor can choose any compression algorithms to generate SigComp messages, and then send byte code for the corresponding decompression algorithm to the UDVM as part of the SigComp message.

### 2.2 Compression algorithm

An interesting point with SigComp is that the compression algorithm is not fixed in the standard; instead this choice is left to the implementer. This means that it is appropriate to assume that SigComp will be implemented using many different compression algorithms. In this paper, the Efficient Protocol Independent Compression (EPIC) [4] is used.

The EPIC scheme is designed to generate ROHC profiles for the compression of new protocol stacks. The scheme includes a number of basic compression techniques (LSB encoding, INFERRED encoding etc.) and a simple language for assigning one or more of these techniques to each field in the stack. In particular EPIC can be used to generate for the compression of signaling messages such as SIP. Since EPIC is pre-programmed with knowledge of how the signaling protocol behaves, the compression ratio obtained is very high and the processing and memory requirements are low.

The drawback with using the standard version of EPIC to compress signaling messages is that it must be programmed with information on how to compress every field in the chosen signaling protocol. This process is straightforward (based on knowledge of how the signaling protocol behaves) but somewhat time-consuming. Fortunately however it is possible to circumvent the programming phase by using the 'learning' version of EPIC.

### 2.3 SIP/SDP static dictionary

When compression is used in SIP/SDP, the compression achieves its maximum rate once a few message exchanges have taken place. This is due to the fact that the first message the compressor sends to the decompressor is only partially compressed, as there is not a previous stored state to compress against. As the goal is to reduce the session set-up time as much as possible, it seems sensible to investigate a mechanism to boost the compression ratio from the first SIP/SDP message.

The SIP/SDP static dictionary [5] is to be used in conjunction with SIP, SDP and SigComp. It makes up a SigComp state that can be referenced in the first SIP/SDP message that compressor sends out.

### 2.4 Extended operations

SigComp extended operations are specified in RFC3321 [6]. They can significantly enhance the compression efficiency compared to per-message compression, which is the mechanism introduced by RFC3320 [1]. SigComp extended operations describe a number of different features that can be used to make the SigComp as efficient as possible. In our SigComp implementation, Dynamic compression, Shared compression and User-Specific Dictionary are used.

In Dynamic compression, compression is done relative to messages sent prior to the current compressed message. This means that the repeatedness of the SIP messages is exploited. In order to use the information from previously sent messages, it is essential for the compressor to gain knowledge about the acceptance of these messages. In case of unreliable transport protocol such as UDP, the SigComp feedback mechanism is used to provide acknowledgements for an endpoint to confirm the reception of the messages. In some cases, e.g. when a reliable transport protocol such as TCP is used, explicit acknowledgements are not necessary.

With Shared compression, compression is done relative to messages received by the end point prior to current compressed message. The compressing endpoint saves the uncompressed version of the SigComp messages as a state. It is efficient to utilize shared compression, because acknowledgements are not necessary. Instead of acknowledging a state, endpoint informs to another endpoint that it has saved the uncompressed version of message by setting a special bit on the SigComp header. The concept of the user-specific dictionary is based on the observation that for protocols such as SIP, a given user/device combination will produce some messages including fields that are always populated with the same data. For example, user-specific information such as the user's URL, name, and e-mail address will likely not change frequently, and will appear regularly in SIP signaling exchange involving a specific user. When user-specific dictionary is used, compressor includes the user-specific dictionary to the initial message which is heading for the remote decompressor. This enhances the compression efficiency.

## 3. Performance Analysis

In this paper, a sequence of SIP messages from a call hold flow taken from [7] is studied. There are 13 messages exchanged between UE and the proxy. The total message size is 7155 bytes.

The SIP messages are compressed using Static, Dynamic, Shared and Mixed compression. In every compression mode, SIP/SDP static dictionary and user-specific dictionary are applied to obtain high compression efficiency. The compression efficiency is influenced by the Decompression Memory Size (DMS) and the amount of information stored in the dictionary. The DMS is 4K bytes and State Memory Size (SMS) is also 4K bytes.

For all the results shown in this section, we consider realistic overhead such as header overhead for each compressed message. When feedback mechanism is used for extended operations, an extra 10~15 bytes overhead is added per message. However, the information about decompression algorithm piggybacked on the first message is not included. It is assumed that both UE and the proxy have already known about decompression algorithm i.e. EPIC.

Four compression modes are studied. Static compression, where the messages are compressed using only the SIP/SDP static dictionary. In case of using Dynamic compression, previously sent messages are used as dictionary for compression. Shared compression, where the messages are compressed using previously received messages. The final case is Mixed compression, where both previously sent and received messages are used as dictionary for compression.

Table 1 shows the compressibility of each compression mode for the entire SIP message sequence. The compression ratio for the entire sequences is 30.52% for Static compression, 19.69 and 15.79 for Dynamic compression and Shared compression respectively. In case of Mixed compression, the compression ratio is 14.13%.

Figure 2 shows the message size. From the figure, it can be seen that the compression size of the first message appears to be same. This is explained by the fact that here, all compression schemes use exactly same dictionary i.e. SIP/SDP static dictionary and user-specific dictionary.

| Compression modes \ Ratio, Size | Our experiment | |
|---|---|---|
| | Ratio (%) | Compressed Size (byte) |
| Static Compression | 30.52 | 2184 |
| Dynamic Compression | 19.69 | 1409 |
| Shared Compression | 15.79 | 1130 |
| Mixed Compression | 14.13 | 1011 |

**Table 1. The compressibility of Static, Dynamic, Shared, and Mixed compression in our experiment.**

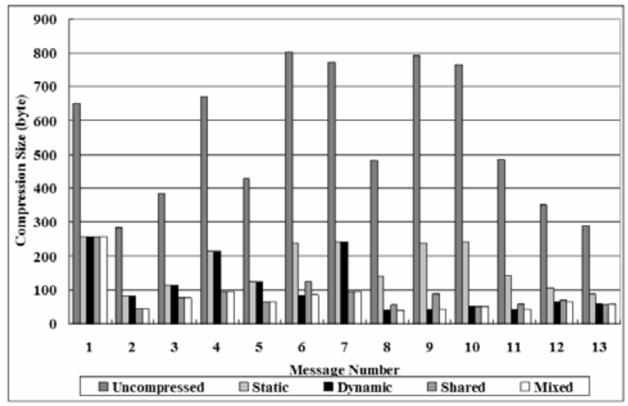


**Figure 2. Message sizes for SIP call-hold flow sequence, subjected to Static, Dynamic, Shared, and Mixed compression.**

For the second and third message, Shared compression and Mixed compression is more efficient than other compression schemes. Because of saving uncompressed SigComp message, these schemes are able to use shared state i.e. uncompressed version of message 1.

For the fourth and later messages, two endpoints successfully use of explicit endpoint initiated acknowledgements. Consequently, Dynamic compression also performs dramatically better than Static compression. Due to the fact that Dynamic compression is invoking, Mixed compression is most efficient. It means that the more information we have, the higher compression efficiency will be.

## 4. SigComp for 3G cellular networks

In this chapter, analysis of the SigComp performance in 3G cellular networks such as one-way Radio Access Network (RAN) delay and compressibility of different SIP message sequences are presented

### 4.1 One-way RAN Delay

The cellular systems must support a sufficient number of users to make them economically feasible. For example, a

WCDMA system can provide maximum bit rates up to 2Mbps in ideal conditions, but that means one single user would consume all radio resources in the cell [8].

According to [3], it is reasonable to assume that a bit rate of the order of 9.6 or 12.2 kbps will be allocated for each user. Therefore, it is important to study the effect of using SigComp in aspect of reducing one-way RAN delay. SigComp can only affect the one-way RAN delay; the core network delay, bearer establishment and the overhead added by lower protocol layers will not be affected.

The one-way RAN delay for each message can be calculated as follows:

$$One\_way\_RAN\_delay = \frac{Size\_of\_message[bits]}{link\_speed[bits/s]} + \frac{RTT}{2}$$

The one-way RAN delay for a SIP session establishment using the SIP call-hold messages is depicted in Figure 3.

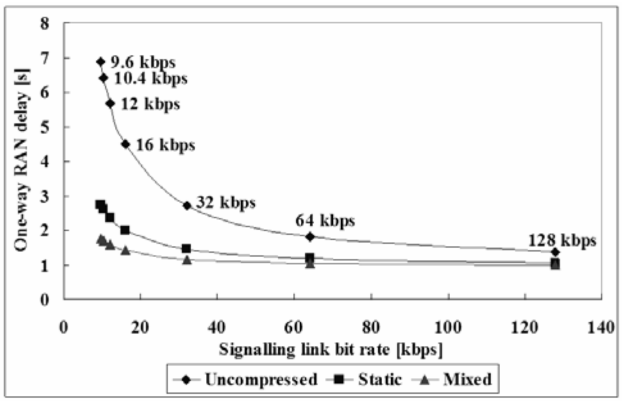


**Figure 3. The one-way RAN delay for a SIP call-hold flow sequence, subjected to Static and Mixed compression.**

From Figure 3, we can observe that the one-way RAN delay for a SIP call-hold session is 6.86 seconds if the bit rate of the signaling link is 9.6kbps and SigComp is not used. To reduce the one-way RAN delay, we applied to Static compression and Mixed compression. In case of Static compression, one-way RAN delay is decreased to 2.75 seconds. As expected, Mixed compression performs better than Static compression. When we use mixed compression, it is decreased to 1.74 seconds i.e. one-way RAN delay is fallen off by 75 %, a decrease of 5.12 seconds compared with using uncompressed SIP messages. And we also see that the improvement SigComp offers is the greatest when bandwidth of the signaling link is low. In other words, higher channel bit rates reduce the need of SigComp, since the gain is not as high in this case.

### 4.2 Compressibility of SIP message sequences

The object of the measurements presented in this chapter is to study the compressibility of different SIP message sequences. Because SigComp is a new feature, it is important to estimate the potential compression gain of utilizing different SIP message sequences. The four SIP signaling sequences are examined; basic voice call [9], basic video call [9], PoC session [10] and 3GPP video call [9].

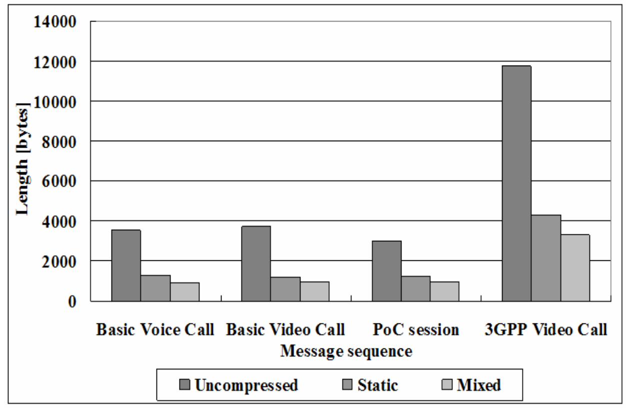


**Figure 4. The compressibility of different sequences : Basic Voice Call, Basic Video Call, PoC session, and 3GPP Video Call.**

From the Figure 4, it can be seen that four SIP message sequences are different in compressibility, because the contents of SIP messages vary according to the SIP message sequence. The results show that the signaling flow having the largest combined message sizes, 3GPP video call organized fourteen SIP messages, benefits the most from SigComp. SIP messages in 3GPP video call results in a good compression ratio for the entire sequence, because the compressibility of the last messages is very high. In case of using the signaling flow made up of six messages is rather low.

## 5. Conclusion

In this paper, we examine the performance of SigComp for 3G cellular networks through measurements based on the SigComp implementation. It is shown that the use of SigComp schemes such as extended operations can be used to significantly improve the compression efficiency and reduce the one-way RAN delay. We also show that the longest SIP message sequences achieve the best compression ratios because their last messages can be compressed very efficiently.

We would like to investigate into this issue further in the future. This paper focused on SigComp performance on the core network side. Another topic will be the performance of SigComp in the 3G mobile terminals supporting SIP.

## Acknowledgements


The authors would like to thank Graham Williamson and Mark West of Roke Manor Research for helping the simulation and providing helpful comments.

## References


[1] R. Price, C. Bormann, J. Christoffersson, etc., Signaling Compression (SigComp), *IETF RFC3320*, January 2003

[2] Haipeng Jin, AC Mahendran, Using SigComp to compress SIP/SDP messages, *IEEE International Conference on Communications*, 2005, 3107-3111.

[3] M. Nordberg, H. Hannu, etc., Improving SigComp performance through extended operations, *58th IEEE Conference Vehicular Technolagy*, 2003, 3425-3428.

[4] R. Price, R. Hancock, S.Mccann, etc., Enhanced TCP/IP Compression for ROHC <draft-price-rohc-epic-02.txt>, January 2002

[5] M. Garcia-Martin, C. Bormann, J. Ott, etc., The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp), *IETF RFC3485*, February 2003

[6] H. Hannu, J. Christoffersson, S. Forsgren, etc., Signaling Compression (SigComp) – Extended Operations, *IETF RFC3321*, January 2003

[7] A. Johnston, R. Sparks, C. Cunningham, etc., Session Initiation Protocol Service Examples draft-ietf-sipping-service-example-10, March 2006

[8] H. Hannu, Signaling Compression (SigComp) Requirements & Assumptions, *IETF RFC3322*, January 2003

[9] 3GPP, *Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)*, TS 24.228 v5.15.0, September 2006

[10] Open Mobile Alliance 2005, *OMA PoC Control Plane Candidate Version 1.0*, March 2005