Transferring Header Compression Context in Mobile IP Networks

Ha Duong, Arek Dadej and Steven Gordon
Institute for Telecommunications Research, University of South Australia
Mawson Lakes Boulevard, Mawson Lakes, SA 5095, Australia
Email: Ha.Duong@postgrads.unisa.edu.au , Arek.Dadej@unisa.edu.au , Steven.Gordon@unisa.edu.au

Abstract
Header compression (HC) is a useful technique for reducing the load on bandwidth-scarce wireless links. HC normally depends on the establishment and synchronisation of context at the compressor and de-compressor. In Mobile IP networks, it is desirable to transfer this context between access routers to avoid the expensive process of context re-establishment. In this paper we propose a method for avoiding the need for context re-establishment when packets are lost during handovers, as well as means for efficiently transferring the context to new access routers (by piggybacking the context on Mobile IP signalling messages). The analysis of our scheme shows that it can reduce signalling load during handovers, which is beneficial in mobile networks with frequent handovers (e.g. wireless LAN hot spots).

1. INTRODUCTION

Mobile IP networks with wireless access offer users the flexibility of mobility, but performance may be compromised due to the bandwidth limited wireless link. Therefore, it is desirable to utilize this bandwidth efficiently. Header Compression (HC) is considered an effective method to reduce the large header overhead when transmitting voice packets over the wireless link. At the expense of extra processing at the transmitting and receiving nodes, HC can reduce the load on the wireless link by 50-70% [3].

In general, the HC process is initialised by establishing the HC state at the sender's compressor and the receiver's decompressor. This HC state, or HC context, is continuously updated at both compressor and decompressor upon receipt of each new packet. In Mobile IP networks, as packets are lost during handovers, so will be the HC context, and consequently, the Mobile Node (MN) and the new Access Router (AR) have to re-establish the context. As context re-establishment can be a time-consuming process and require additional bandwidth, transferring the HC context from the old AR to the new AR, either via the fixed network or via the MN, is a preferred method over context re-establishment [5]. However, even with HC context transfer, if packets are lost during handover, the compressor and decompressor can become miss-synchronized, forcing a full context re-establishment. This problem illustrates one of the difficulties faced when transferring context that may change dynamically.

Context transfer has emerged as a one of the problems that must be solved to achieve full user mobility in IP networks, with others being routing (essentially solved by Mobile IP), security, quality of service and service context re-establishment. In general, context is the information on the current state of a service provided by an AR. Currently, HC, QoS, security, Authentication, Authorization, and Accounting (AAA) have been identified as candidate services for context transfer [5]. The Koodli and Perkins' work [10] was one of first that mentioned context transfers in mobile networks. The authors, suggesting that context transfer be a part of fast handover signalling, showed the feasibility of network layer fast handover and HC context transfer. However, this approach still requires explicit signalling between ARs and does not take into account the possibility of miss-synchronization between compressor and decompressor because of packet loss during handover period. There are several Internet drafts on context transfer protocols [9][6], suggesting protocol requirements or proposing generic protocols, however these have not been evaluated through analytical, simulation or test bed means. It is also unclear how these protocols would deal with dynamic context such as the HC context.

In this paper, we present a simple approach to deal with the miss-synchronization of the HC context by using three-state HC schemes. We also suggest a method of transferring the context, based on Mobile IP signalling. The rest of the paper is structured as follows. In the next section, we provide an overview of HC schemes and Mobile IP, and describe the miss-synchronization problem in more detail. In sections 3 and 4, we describe our solution and present simulation results to illustrate the effectiveness of the solution. Finally, we give some concluding remarks and comment on areas for intended future work.

2. BACKGROUND

2.1 Overview of Header Compression

A number of header compression schemes have been proposed for existing Internet protocols, such as [1][11] and [7]. In some cases, for example using IPv6/UDP/RTP, the headers can be reduced from 84 bytes to 1 byte, and therefore, can bring up to 73% traffic reduction for a voice packet with 30 bytes payload. The basic principle of the HC schemes is based on the fact that there is a significant amount of redundant information in the headers of subsequent packets. In other words, most of the header field values remain the same over the lifetime of a session. For a non-TCP session, almost all fields are constant, whereas for a TCP session, there are several constant fields while others change in a predictable way. The header fields can be classified based on their behaviour during the session lifetime. For example, the HC scheme in [7] classifies the header fields into:

NOCHANGE (STATIC): the field is not expected to change.

DELTA: the field may change often but usually by a small value (delta) so that the delta can be sent instead of the absolute value of the field.

RANDOM: the field is expected to be random and thus needs to be transmitted on the "as is" basis

INFERRED: the field value can be inferred from other values.

The compression process is initialised with the establishment of the HC context at both compressor and decompressor. The compressor examines packet headers, copies the values of packet header fields to establish the HC context for the new connection, assigns a context identifier (CID) to the established context, and then transmits the full packet (uncompressed), with CID[1], to the decompressor. The CID plays the role of a signal to notify the decompressor about a new HC session. Upon receiving a packet with CID, the decompressor starts to build the HC context using packet header fields. The HC context consists of the STATIC and DELTA fields. Once the HC context is established, packet headers can be compressed by not including STATIC fields, and by using fewer bits to store delta values. Upon receiving a compressed packet header, the decompressor restores the full header by using delta values (if present) and the stored HC context identified by CID.

To increase the probability that the decompressor correctly establishes the HC context, n full header packets (n>1) with CID are initially sent. The decompressor may optionally send an *Acknowledgment* packet to confirm the establishment of the HC context. To further enhance compression reliability, some HC schemes have suggested a multi-state HC context. For instance, the scheme in [1] employees a three-state HC context as illustrated in Figure 1.

In the partly compressed packet, the packet header consists of CID, RANDOM fields, DELTA fields and possibly INFERRED fields. Therefore, the STATIC fields will make up the HC context. In the fully compressed packet, the packet header consists of CID, RANDOM fields, and delta values.

[1] In [13] for example, the CID is transmitted in the Total Length field of the IPv4 packet.

Thus, the STATIC and DELTA fields are stored as the HC context, and the DELTA fields are updated whenever a compressible packet arrives. This three-state scheme enhances reliability by limiting the impact of lost HC context when in the first order (partly compressed) state

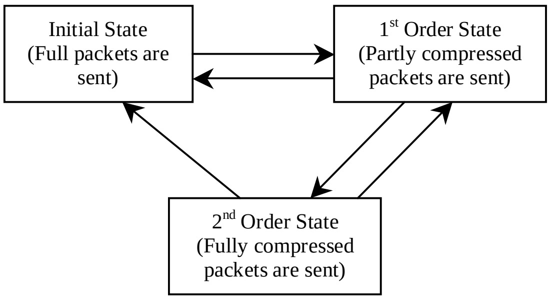

Figure 1. Three state HC context

2.2 Overview of Mobile IP

Mobility in the Internet has been a very active research topic within the IETF community. In the last decade, intense research effort has been spent on the issue, and led to explosive development of mechanisms supporting mobility. The IETF Mobile IP Working Group has developed a solution officially named **IP mobility support** but popularly called **Mobile IP** [2]. The main characteristics of Mobile IP include transparency to applications and transport layer protocols, scalability, and macro mobility. Mobile IP includes two main functions, registration and tunnelling. For our purpose in this paper, we will briefly describe the registration procedure, which is illustrated in Figure 2. Whenever a MN discovers that it is moving into a new subnet, it sends a *Registration Request* message, which includes the MN home address, care of address and Home Agent (HA) address, to the new Foreign Agent (FA). The new FA, which is located at the Access Router, relays the message to the HA after retrieving information necessary for serving the MN in the future. In response to the *Registration Request*, the HA sends a *Registration*

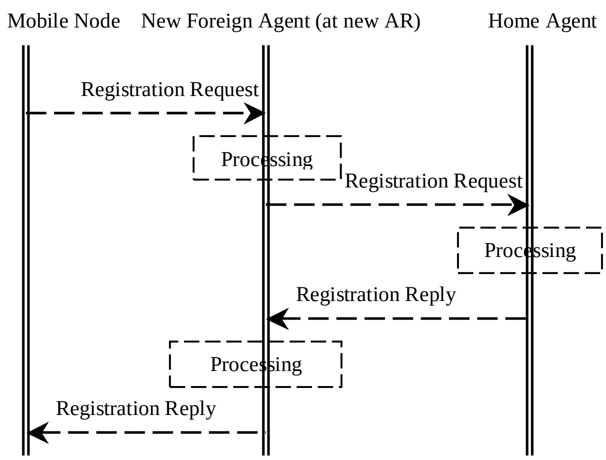

Figure 2. Mobile IP registration process

Reply to the new FA. In turn, the FA sends the *Registration Reply* to the MN to confirm (or reject) the MN registration at the new FA. Following this registration procedure, all communications to and from the MN go via the new FA.

2.3 Problem description

Let us consider a scenario when the MN is running a Voice over IP (VoIP) application and the HC context was established at both MN and AR to allow transmission of compressed voice packets. When the MN performs a Mobile IP handover to the new FA, the HC context must be re-established as the new FA has no knowledge of the previously transmitted packets i.e. the HC context. The context reestablishment, which involves sending *n* packets with full headers, is a time- and bandwidth-consuming process. For example, the cost of the HC context establishment in a shared channel such as Wireless LAN [3] can be 26.4% extra channel occupation by the full packets, and therefore reduction in the effective compression ratio by more than 7%. Therefore, a potential enhancement is to transfer the HC context from the old AR to the new AR to avoid context re-establishment. However, this approach fails if packets are lost during handover. As the HC context depends on previous packets received, any packet loss can lead to miss-synchronisation between compressor and decompressor. A miss-synchronization will force the compressor and decompressor to establish a new HC context, from the beginning (i.e. from the Initiate State in Figure 1). Unfortunately, packet loss has a great potential to take place in the handover process. Even though the buffering techniques can reduce packet loss, there is no guarantee that packet loss can be avoided. Moreover, buffering may increase packet delay, which affects time-sensitive application such as VoIP [12]. Therefore, it is important to be able to perform HC context transfer in mobile IP networks, while avoiding problems of miss-synchronisation (which negate any benefits of context transfer).

3. PROPOSED SOLUTION

In this section, we propose a method for avoiding the miss-synchronisation problem in HC context transfer (Section 3.1), as well as a means for efficiently transferring the context from the old AR to the new AR (Section 3.2).

3.1 Avoiding miss-synchronization problem

Our solution is derived from a simple observation of the three-state HC scheme, described in Section 2.1. Since the HC context in the 2nd order state is very sensitive to packet loss (i.e. a lost packet results in returning to the initial state), the compressor and decompressor should return to the 1st order state whenever there is an indication of upcoming handover (link layer triggers can be used, for example, as an indication). Readers will recall that in the 1st order state the HC context is made up by only STATIC fields; therefore, packet loss during handover time will not affect the HC context. The HC context of the 1st order state is to be transferred by method specified in the Section 3.2. When the handover process is completed, the HC may return to the 2nd order state. The avoidance of miss-synchronization is achieved at the expense of lower compression ratio in the 1st order state. However, in the following, we will show that this is a reasonable trade-off, since there are only small differences between the compression ratios in the 1st order state and the 2nd order state.

Table 1 shows the contribution to overhead reduction of the 1st and 2nd states for different packet headers. The following notations are used

s: The total size of STATIC fields.

d: The total size of DELTA fields.

Δ: The total size of fields carrying small values (delta). Usually, one byte field is enough to carry delta value of a DELTA field.

$\mathbf{R_1}$: Overhead reduction of the 1st state compared with the initial state.

$\mathbf{R_2}$: Overhead reduction of the 2nd state compared with the 1st state.

All values are expressed in bytes.

Packet header	s	d	Δ	Contribution to $\mathbf{R_1}$ (**s**)	Contribution to $\mathbf{R_2}$ (**d**-Δ)
IPv4	14	2	1	14	1
IPv6	38	0	0	38	0
TCP	4.5*	12	2	4*	10
UDP	4	0	0	4	0
RTP	4.5*	6	2	4*	4

** The compressed header size is an integer number of bytes*

Table 1. Contributions (in bytes) to overhead reduction.

We can see from Table 1, that for all protocols (except TCP) the compression ratio in the 1st state is only slightly less than in the 2nd state. For example, using IPv4, the header is reduced by 14 bytes when in the 1st state, and only by a further 1 byte when moving from the 1st state to the 2nd state. When using TCP, although the 1st state brings a reasonable overhead reduction, operating in the 2nd state significantly enhances the compression ratio. However, for VoIP applications UDP is the preferred transport protocol.

If we assume that the CID requires 1 byte in the IP header field of a compressed packet, then Table 2 shows the overhead reduction for a VoIP application that uses RTP over UDP over IP (v4 and v6). $\mathbf{R_1}$ indicates the reduction when operating in the 1st state while $\mathbf{R_{overall}}$ indicates the reduction when in the 2nd state. Intuitively, we can see that the small reduction in compression ratio when reverting to the 1st state during the handover process is an adequate trade-off for avoiding miss-synchronisation (and hence full context re-establishment).

Packet header	R_1	$R_{overall}$
IPv4/UDP/RTP	21	26
IPv6/UDP/RTP	45	49

Table 2. Overhead reduction (bytes) in three-states HC scheme.

3.2 Proposed method of transferring HC context

Assuming that miss-synchronisation can be avoided, for HC context transfer to be beneficial we require a means for transferring the context from the old AR to the new AR with minimum overhead. We propose that the MN informs the new AR of the context during the handover procedure. This is based on the observation that the HC context of the compressor (or decompressor) at the old AR is the same as the HC context of the de-compressor (or compressor) at the MN. In other words, the MN has a copy of the HC context held at the old AR; when the handover process begins, the MN will transfer the HC context to the new AR. For instance, the MN can append the context data to the *Registration Request* message in the Mobile IP protocol as a *Registration Extension* (which we will call *Context Transfer Extension*). At the new AR, the *Context Transfer Extension* will be detached from the *Registration Request* message and the contents of the *Context Transfer Extension* can be stored as the HC context of the link between the new AR and the MN. The rest of the *Registration Request* message will be processed according to the normal Mobile IP procedures. This method does not require any new signalling messages between network entities such as the MN, old AR and new AR. It also guarantees that the transferred HC context is available at the new AR before the MN and the new AR begin to exchange packets within the HC session. The *Context Transfer Extension* containing context data of the 1st order state will be of a few tens of bytes in size; therefore it does not increase very much the *Registration Request* message size.

4. SIMULATION RESULTS

We have used the OPNET simulation tool (http://www.mil3.com) to validate the proposed solution and to compare its performance with the case of full HC context re-establishment at the new AR.

Figure 3 illustrates the network topology that has been used in the simulation. The border routers R_F, R_H and R_C connect the foreign domain, home subnet and corresponding subnet to the Internet. The HA and FAs reside in ARs equipped with IEEE 802.11 access point functionality. The Mobile IP functionality has been implemented in the HA, FAs and MN using the simulation model developed by T. Park [8]. The acknowledgement-based three-state HC scheme was implemented at the HA, FAs and MN. The HC context establishment requires the compressor to send n full header packets (for the results presented in this paper n = 3) and to receive an Ack from the decompressor. For simplicity, we only model the IPv4/UDP protocol stack to carry voice packets. To obtain the constant data rate, we disabled silence compression of the voice encoder (i.e. voice packets are still generated during the silence period). The constant data rate makes simulation observation and analysis easier. VoIP application uses G.729 as the encoder scheme, with frame size equivalent to 10 ms and coding rate of 8kbps.

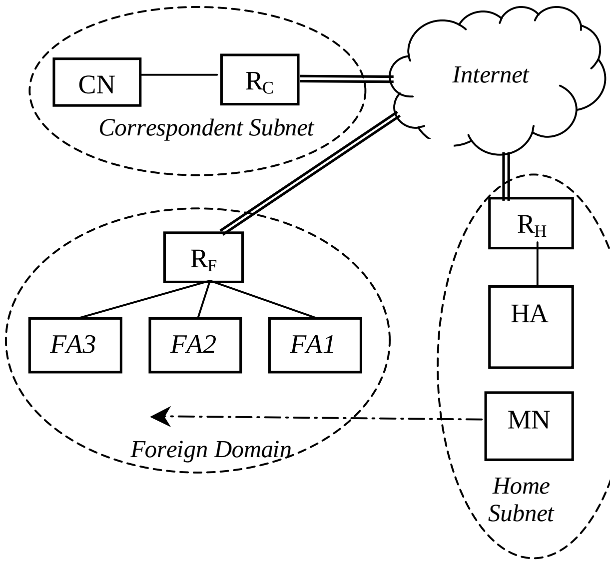

Figure 3. Network topology used in the OPNET simulation environment

The following scenario has been used in the simulation: the MN connects with its HA at time t = 0sec, and starts a VoIP session with the Corresponding Node (CN) at time t = 15sec. While the VoIP session is ongoing, 2.3, the packet loss during the handover causes a context miss-match between the compressor and decompressor, forcing context re-establishment.

We also investigated how our HC context transfer protocol performs during the handover periods. It is worth to mention that since VoIP is a two-way interactive application, packets of the flow (CN→ MN) are compressed at FA or HA and decompressed at MN, meanwhile MN acts as compressor and FA (or HA) acts as decompressor for packets of the flow (MN→ CN). Figure 4 shows a “snapshot” of packet size versus time at the MN compressor during handover periods with FA1, FA2 and FA3. In the case of the handover without context transfer, after sending a Mobile IP Registration Request (Point 1 in Figure 4), the MN has to send full header packets (Point 2 in Figure 4) to establish a new HC context for the flow (MN→ CN). An *Acknowledgment* (Point 3 in Figure 4) of

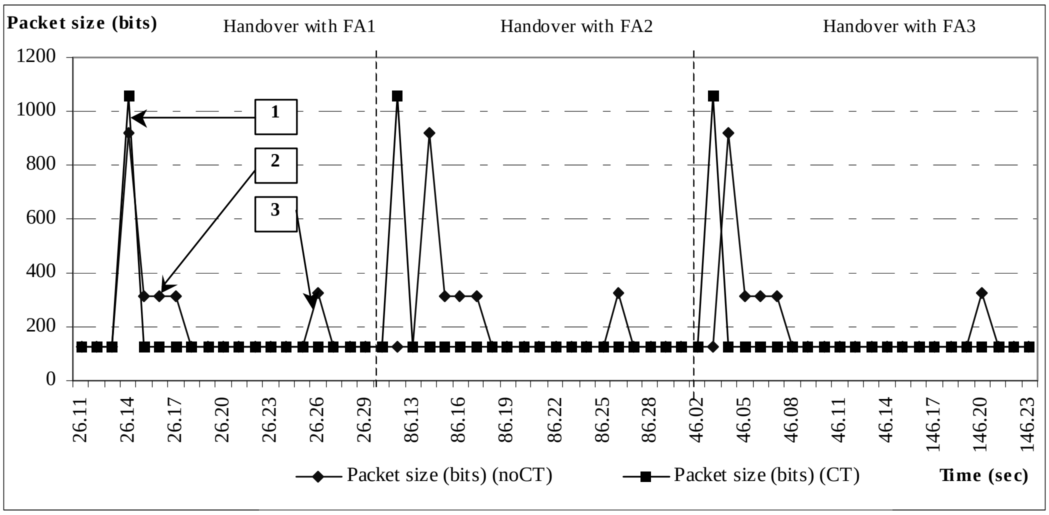

Figure 4. **Packet size at MN compressor in two cases: with context transfer (CT) and without CT**

context establishment for the flow (CN→ MN) is also sent. On the other hand, in the case of the handover with context transfer, the MN sends Mobile IP Registration Request with *Context Transfer Extension* (Point 1 in Figure 4), and then the MN can send compressed header packets to the new FA. MN also does not need to send *Acknowledgment* for the flow (CN→MN) as the new FA, with HC transferred context, can start sending compressed header packets immediately. The same behaviour is observed for the handovers with FA2 and FA3 (the times when handovers occur differ slightly because each scenario is run as independent simulation). In all handover periods, based on the above packet size behaviour, we can observe that the MN without context transfer generates more traffic load than the MN with context transfer. The reduction in load brought in by the context transfer depends on factors such as number of full header packets sent to establish HC context, whether or not Acknowledgements are used, the ratio of packet payload and packet header, and the round trip time (RTT) between compressor and decompressor. For the scenario assumed in simulations, we estimate that approximately a 20% load reduction on the WLAN is achieved during the handover period. This reduction is quite significant if we consider that the MN and AR usually require more bandwidth for Mobile IP signalling during handover. Although the handover period may be relatively short (e.g. a few hundred milliseconds) the load reduction becomes a significant factor in "hot spots" where WLANs are serving large numbers of MNs encountering frequent handovers.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a simple solution to the problem of miss-synchronization which occurs due to packet losses during handover, when transferring header compression context in Mobile IP networks. In addition, we have illustrated how to transfer the HC context from an old access router to a new access router using existing Mobile IP signalling messages. Preliminary simulation results indicated that load reduction of 20% could be achieved during the handover period by transferring the HC context. The current work can be extended by applying our approach to a cellular mobile network link. It is expected that the HC context transfer in a cellular network can bring even higher benefits, due to the higher cost of HC establishment in the cellular link The main reason here is that as the cellular link has much longer RTT, typically 100-200ms, there are more in-flight full header packets during HC establishment period. Also, during handover period more in-flight compressed packets may need to be discarded at the decompressor if miss-synchronization happens. This makes using the proposed HC context transfer more beneficial.

REFERENCES

[1] C. Bormann (editor), "Robust Header Compression (ROHC)," Request for Comments 3095, IETF, July 2001,

[2] C. Perkins (editor), "IP Mobility Support for IP v4," Request for Comments 3320, IETF, January 2002.

[3] C. Westphal and R. Koodli, "IP Header Compression: A Study of Context Establishment," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC2003)*, New Orleans, LA USA, 2003.

[4] Context Transfer, Handoff Candidate Discovery, and Dormant Mode Host Alerting (Seamoby) Working Group, IETF, Http:// www.ietf.org/html.charters/seamoby-charter.html

[5] J. Kempf et. al., "Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network, " Internet Draft of IETF, draft-ietf-seamoby-

context-transfer-problem-stat-04.txt (work in progress), December 2001.

[6] J. Loughney et. al., "Context Transfer Protocol," Internet Draft of IETF, draft-ietf-seamoby-ctp-00.txt (work in progress), October 2002.

[7] M. Degermark, B. Nordgen , and S. Pink, "IP Header Compression," Request for Comments 2507, IETF, February 1999.

[8] Park T. and Dadej A., "OPNET Simulation Modelling and Analysis of Enhanced Mobile IP," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC2003)*, New Orleans, LA USA, 2003.

[9] R. Koodli R. and C. E. Perkins , "A Context Transfer Protocol for Seamless Mobility," Internet Draft of IETF, draft-koodlt-seamoby-ct-04.txt (work in progress), August 2002.

[10] R. Koodli R. and C. E. Perkins, "Fast Handovers and Context Transfers in Mobile Networks," in *ACM Computer Communication Review*, vol. 31, number 5, 2001.

[11] S. Casner and V. Jacobson , "Compressing IP/UDP/TCP Headers for Low-Speed Serial Links," Request for Comments 2508, IETF, February 1999,

[12] T. Park and A. Dadej , "Adaptive Handover Control in IP-based Mobility Networks," in *Proceedings of 1st Workshop on the Internet , Telecommunications and Signal Processing (WITSP 2002)*, Wollongong, Australia, 2002.

[13] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links," Request for Comments 1144, IETF, February 1999.