# IP Header Compression: A Study of Context Establishment

Cédric Westphal, Rajeev Koodli

*Abstract*—

**The use of bandwidth constrained links in wireless networks necessitates the use of bandwidth saving header compression schemes. In these schemes, a compressor and a decompressor collaborate to encode bulky IP headers into streamlined compressed headers. Intuitively, the gain from compression is the average compressed header length divided by the uncompressed header size. In this paper, we show that this is not true in many situations: this does not account for the cost introduced by the variability of the header sizes during compression context initialization.**

**First, we provide analytical basis for cost of compression context creation and compare the relative costs of establishing contexts with an optimistic approach and with an acknowledgment-based approach. We conclude that the optimistic approach can be used as an upper bound in our analytical model. Second, we evaluate the impact of header compression on traffic patterns in terms of bandwidth allocation, in order to accommodate the burst introduced by compression context initialization. We compute the actual compression ratio, that is the actual number of users of IP header compression that can be multiplexed onto a given link. We show this number is significantly different from the one computed by dividing the link bandwidth by the average rate of a compressed flow. Our results provide a formal basis for context management especially during handovers in mobile networks. For example, our result indicates that there is significant benefit to relocate compression contexts (from one network node to another) rather than to re-establish them each time during handovers.**

*Keywords*— **Header compression, handovers, context transfers, bandwidth allocation.**

## I. Introduction

In a mobile network, such as a wide-area all-IP cellular network, lower link bandwidth requires IP overhead reduction. Header compression is seen as the method to accomplish this reduction [1]. Briefly, Header Compression (HC) functions as follows: the *compressor*, running for instance on a Mobile Node (MN), sends packets with entire IP headers until it gains *sufficient confidence* that the *decompressor*, running for instance on the MN's Access Router, has received the required "Full Context" containing header information. Subsequently, the compressor sends packets with headers as small as possible. This *context initialization* phase forms the basis for both the compressor and the decompressor to progress towards spectrally efficient state using a consistent reference state. The compressor and the decompressor operate in synchronization, either implicitly (optimistic mode) or explicitly (acknowledged mode).

In this paper, we evaluate the cost of compression context establishment in terms of bandwidth. Header compression reduces the IPv6/UDP/RTP header from 84 bytes to 1 byte, and the overall packet size for voice traffic from 114 to 31 bytes. Intuitively, the gain would thus be computed as 31/114 = 27 %, or a 73 % reduction in traffic, or almost four-fold capacity increase for voice traffic. We formally show that this gain is over-estimated since it does not consider the variability of header sizes during context establishment. We propose a model to compute this hidden cost of HC for different channel types. This model provides a formal basis for compression context management, and could be used for Connection Admission Control (CAC) when sessions are first started. For subsequent instantiations during handovers, we suggest relocation of already established compression contexts to avoid the context establishment overhead.

Prior work [2], [11], [18] does not consider formal cost analysis. In Section II, we present and motivate the basic model we use. In Section III, we evaluate the cost of compression context initialization. In Section IV, we evaluate the impact of context initialization phase on some typical bandwidth allocation schemes.

## II. Problem Setting and Basic Model

Our network model is as follows. The Mobile Nodes use IPv6 as the IP layer protocol and use Mobile IPv6 as the mobility management protocol[1] [16], [5], [7]. The access point a MN attaches to is an IPv6 router that supports IPv6/UDP/RTP compression [1]. After attaching to its router, a typical MN initializes header compression contexts for one or more of its IP connections. Subsequently, a MN undergoes handover from a Previous Access Router to a New Access Router, bringing with it packet streams undergoing compression. See Figure 1.

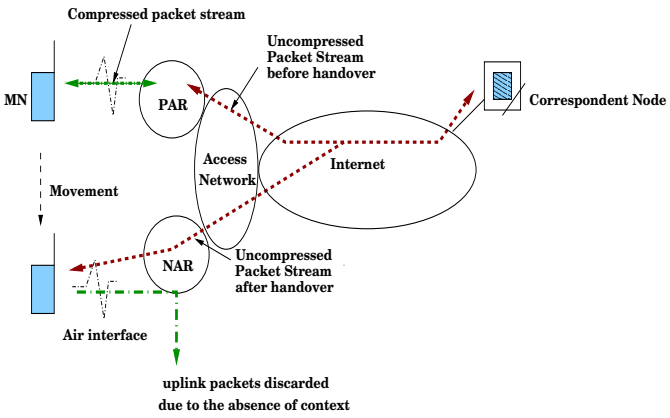


Fig. 1. Mobility Reference Diagram

C. Westphal and R.Koodli are with the Communication System Laboratory, Nokia Research Center, Mountain View, California. E-mail: {cedric.westphal, rajeev.koodli}@nokia.com

[1] we note that the results are applicable to IPv4 and Mobile IPv4 with appropriate header size values.

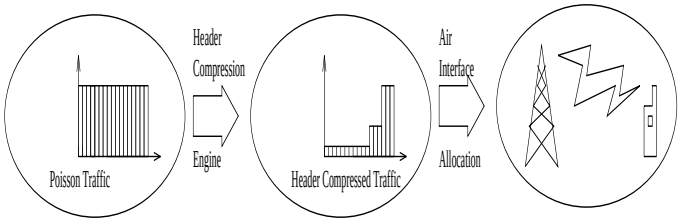


Fig. 2. Traffi c model assumptions

In Figure 2, we illustrate the traffic flow in which connections arriving according to Poisson process generate Full Header packets which are then compressed by a compression engine and sent over a cellular link. As is evident, the connections introduce a burst during initialization.

Figure 3 illustrates a simplified header compression state machine. The traditional state machine according to, for instance, [1], [6], [8], or [13] defines three states. Of these, the First Order and the Second Order states are preferred since fewer header bits are sent in those states. In our model, we consider only two states; an FH state in which all the header bits are sent and a Compressed Header (CH) state in which few header bits are sent. We make this simplification because header bits sent in both Second-Order and First-Order are very small compared to Full Header, and can thus be collapsed into one single state for analysis purposes.

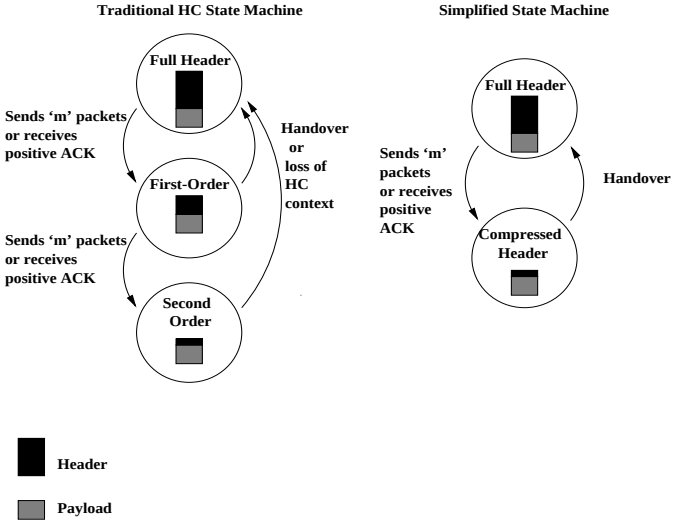


Fig. 3. Header Compression State Machines

We denote by $\lambda$ the call arrival rate, and by $\mu^{-1}$ the mean call duration. The sampling interval is $\tau$. This means that calls are initiated according to a Poisson process with rate $\lambda$ and send a packet every sampling interval $\tau$ (typically every 20 or 30ms) for an exponentially distributed length of time with parameter $\mu$ (see [4], [17] regarding this assumption). We denote by $\beta$ the bandwidth request of a connection, which is related to the payload size by the relationship: $\beta = \frac{\text{payload size}}{\tau}$. We define by $\delta$ the maximum delay allowed on the link that is acceptable to the user. Lastly, we denote by $f$, $c$ and $\pi$ the sizes of the Full Header, the Compressed Header, and of the typical payload respectively

## III. Cost of Context Initialization

In acknowledged mode, it is simple to calculate the number of Full Header packets needed to establish the context. If the link RTT is $X$ ms, and an FH packet is sent every $y$ ms, where $X \geq y$, then at least $\lceil \frac{X}{y} \rceil$ FH packets are sent before the compressor could receive an acknowledgment. In practice, it is typically $\lceil \frac{X}{y} \rceil + z$, where $z$ determines how many successive packets the decompressor has to receive before it could send an acknowledgment.

### A. Probability of Context Establishment

In a more general fashion, assume that the decompressor needs $n$ packets to establish the context. Assume that the channel loses packets with probability $p$, which is a function of the Signal to Noise ratio (SNR) of the channel and the modulation scheme used. Assume also that the compressor sends only $m \geq n$ packets.

The probability that $n$ packets are received when $m$ are sent is expressed as:

$$\begin{aligned} P(n,m) &= P(\text{receive at least n successfully}|\text{m sent}) \\ &= \Sigma_{j=n}^{m} \binom{j}{m} p^{m-j}(1-p)^{j} \end{aligned} \quad (1)$$

Obviously, as the total number $m$ of packets sent increases, the confidence of context establishment also increases. When signal reception gets worse, e.g., near cell boundaries, the compressor needs to send more FH packets in order to gain sufficient confidence. For example, when the packet reception probability is 0.8, sending 6 packets has approximately 98% success of establishing contexts compared to approximately 50% success with sending 3 packets. When the successful packet reception probability is close to one, sending 3 packets should provide sufficient confidence for the compressor. In such a case, the compressor may use the optimistic approach with "higher confidence". Of course, when $m$ increases so much that the inter-arrival time between $m$ packets is longer than $n$ inter-arrival plus one RTT, then there is no value in guessing what the context is when the acknowledgment itself is available.

If the compressor changes state too quickly, then the decompressor sends a complaint message. It takes one RTT for the compressor to ascertain this, and the compressed packets sent and successfully received during this repair process are lost due to the absence of proper context. Thus, the packets lost due to this "jump ahead penalty" are those sent during a time length of one RTT, and they are lost with probability $1 - P(n,m)$. A numerical example in the Appendix illustrates this further.

### B. General cost of context initialization

Using the notation in section II, the bandwidth cost is for the acknowledged context initialization:

$$\text{cost}_{ack} = \left( \lceil \frac{X}{y} \rceil + z \right) f \tag{2}$$

and, following the same steps as in the numerical example, the bandwidth cost for the optimistic context initialization is:

$$\begin{aligned} \text{cost}_{opt} &= \min_{m=1,2,\ldots} \Big[ mfP(z|m) + \\ &\left( mf + \lceil \frac{X}{y} \rceil (c+\pi) + \lceil \frac{X}{y} \rceil (f+z) \right) (1 - P(z|m)) \Big] \\ &= \min_{m=1,2,\ldots} \Big[ mf + \\ &(1 - P(z|m)) \left( \lceil \frac{X}{y} \rceil (f+c+\pi) + zf \right) \Big] \end{aligned} \tag{3}$$

Equation 3 provides a way to compute the optimal value $m$ that minimizes the cost of optimistic context re-establishment.

In figure 4, we plot the cost for optimized context initialization for different values of $m$.

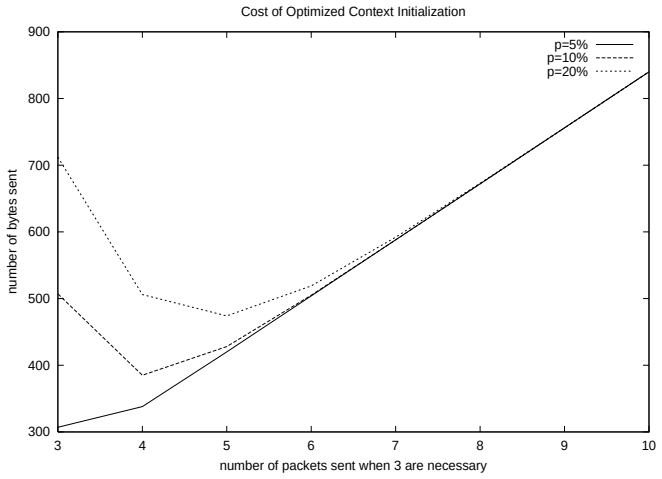


Fig. 4. Cost of optimized context initialization vs. number of packet sent $m$

We use the optimistic context initialization as a deterministic model for the bursty header compression traffic patterns. This *secondary cost* of context initialization is the subject of the next section.

## IV. Header compression burst management

### A. Secondary cost of context initialization

We focus now on the cost of context initialization as it percolates onto the channel allocation. The same framework could be used for CAC study as well. The intuition is as follows: some room has to be reserved in the channel allocation to allow for context initialization. During handover for instance, if the compression context is transferred from the previous router, then the bandwidth can be allocated for a flow with only compressed headers and payload packets. If on the other hand, the context has to be re-initialized, then some bandwidth has to be allocated to accommodate the initialization burst. This is what we denote as the secondary cost of context initialization.

From the previous section, observe that only the initial $n$ packets are sent with Full Header, and the remaining packets are sent with compressed header. Thus, these initial $n$ packets create a burst over the regular traffic. According to our assumption in section II, the connections that generate such bursts arrive according to a Poisson process. Thus, we consider the burst pattern due to FH packets as a peculiar case of M/G/1 queue (see [12]).

It is sufficient to consider only the load being brought by the Full Headers in order to study the burst introduced by arriving connections. We denote by $H$ the extra load introduced by a Full Header: $H = f - c$. This extra load $H$ is carried by the first $n$ packets when establishing the compression context.

We wish to obtain a bound on the maximum delay that this initial burst introduces on subsequent packets, since this delay directly affects probability of packet discarding ("overflow probability") due to buffer overflow or due to real-time constraints (even when no buffer overflow happens). For this, we assume the arrival of a giant packet of size $nH$ with the first packet, instead of the arrival of the first $n$ packets of size $H$. See Figure 5 for an illustration.

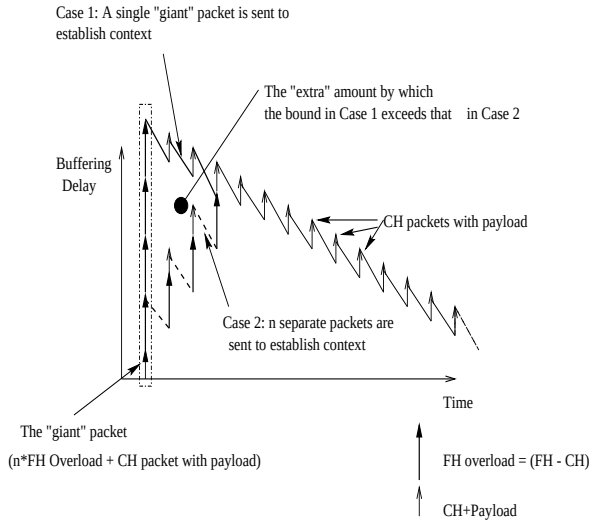


Fig. 5. Maximum delay brought by the initial FH packets

From Figure 5, it is clear that the assumption of the arrival of a giant packet with size $nH$ yields a strict upper bound, since it does not take into account the reduction of the load seen by subsequent packets due to the inter-arrival spread. Yet, if the allocation is such that the initial bulk subsides only slowly, then the delay pattern for the packets $n+1, n+2, \ldots$, is very similar to the case where there are $n$ separate arrivals with size $H$ each. The maximum delay however, is incurred for packet $n+1$.

Only a given number of concurrent requests for context es-

tablishment can be served at the same time. Denote by $N_{max}$ the upper bound on the number of handovers. This is an $M/D/1/N_{max}$ system. If we denote by

$$a_k = \frac{e^{-\lambda nH}(\lambda nH)^k}{k!} \quad (4)$$

the probability that $k$ new connections request context establishment during the normalized processing time of one context establishment, then the distribution of the $M/D/1/N_{max}$ system solves the following triangular system of equations:

$$\begin{aligned} p_0 &= p_0 a_0 + p_1 a_0 \\ p_1 &= p_0 a_1 + p_1 a_1 + p_2 a_0 \\ p_j &= p_0 a_j + p_1 a_j + p_2 a_{j-1} + \ldots + p_{j+1} a_0 \\ \Sigma_{j=0}^{N_{max}} p_j &= 1 \end{aligned} \quad (5)$$

The probability that the number of requests due to handover exceeds the maximum number of concurrent requests to establish compression context, denoted by $P_overflow$ is, due tot he PASTA property:

$$P_{overflow} = p_{N_{max}} \quad (6)$$

This probability will be used in the next section to define some bandwidth allocation schemes.

### *B. Solutions to the FH Burst Management Problem*

We now consider solutions to managing the burst introduced by compression context establishment. We investigate three types of resource allocations, namely Constant Bit Rate (CBR), a channel with shared allocation of the header bursts and reserved bandwidth for the payload, and a totally shared channel allocation.

We define some performance measures. The *call dropping probability* is the probability that a call is refused access to the resource due to the inability to allocate some bandwidth to this call. This readily applies to the CBR channel. In the shared channel case, there is no call dropping in a strict sense. However, if too many concurrent connections consume the resource, they all suffer. The connections "drop" themselves when quality degradation, measured by the packet loss due to congestion or by the packet delay, is such that the user cannot continue with the call.

The *maximum channel utilization* is the ratio of maximum payload throughput over the overall throughput capacity of the channel.

#### B.1 Constant Bit Rate allocation

We define $\alpha$ to be the bandwidth over-allocation co-efficient for a constant bit rate allocation. The channel resource manager overprovisions the channel by the coefficient $\alpha$, i.e., for the request of bandwidth $\beta$, it allocates $\beta(1+\alpha)$. This allocation is constant throughout the life of the connection: it is Constant Bit Rate.

For $\alpha = 0$, that is when the channel allocated has the minimum bandwidth required to accept the flow, then the buffering delay is $nH/\beta$. The other extreme, where $\alpha$ is such that $\beta(1+\alpha)$ is greater than the payload plus Full Header rate, gives a zero delay (without considering the jitter).

Thus the maximum added delay to the packets of the connection with allocation coefficient $\alpha$ follows:

$$n(\frac{H}{\beta(1+\alpha)} - \tau) \text{ which is less than } \frac{nH}{\beta(1+\alpha)} \quad (7)$$

Recall that we defined $\tau$ in section II to be the interarrival time between packets of the same flow. Recall as well that $\delta$ is the maximum acceptable delay of this system. Equation 7 gives the delay for the $(n+1)$st packet. The maximum channel utilization in this scenario is $\frac{1}{1+\alpha}$.

For a voice application, with an acceptable end-to-end delay of 150 ms, typical values of the parameters would be: $\delta = 30$ ms, $\beta = 10$ kbit/s, $\tau = 20$ ms. If $n = 3$ packets are needed to establish the CH compression state, with initial headers of $H = 84\ bytes = 672\ bits$, then to satisfy the maximum delay constraint, we need:

$$\begin{aligned} n(\frac{H}{\beta(1+\alpha)} - \tau) &\leq \quad \delta \\ \text{or } \alpha \geq \frac{H - 30.10^{-3}\beta}{30.10^{-3}\beta} &= \frac{672 - 300}{300} = 1.24 \end{aligned}$$

In this case, a request for 10 kbit/s actually gets 22.4 kbit/s. The *effective* compression of the header is 1.24 times the payload, namely 31 bytes for 24 bytes payloads. Even though CH size is 1 byte, handling the bursts within the delay constraints entails an effective size of 31 bytes. The variability in the header size costs 30 bytes per packets here. Even though packets are actually compressed a four-fold factor from 109 bytes down to 25, half of this benefit is lost into an effective compression from 109 to 56, a two-fold factor.

#### B.2 Shared Signaling channel

Assume that the resource manager allocates a shared channel dedicated to establish compression state. This channel, shared across different flows and users, is used for accommodating the bursts due to FH packets while a separate dedicated channel is assigned for each flow to handle the steadier payload. It thus helps to visualize payload traffic separately from the headers that generate bursts. Note that payload traffic includes a small CH header. The total bandwidth $\beta_T$ is divided between $\alpha_p\beta_T$ for the payload, and $(1-\alpha_p)\beta_T$ for the extra header traffic, where $0 \leq 1-\alpha_p \leq 1$ is the shared channel allocation coefficient. And, $\alpha_p$ is the maximum payload utilization.

In the payload channel, the maximum number of concurrent connections all requesting bandwidth at rate $\beta$ is:

$$N_{max} = \frac{\alpha_p \beta_T}{\beta} \quad (8)$$

Given $N_{max}$ as a function of $\alpha_p$, we can now identify the extra-header traffic. From section IV-A, the arrival process of the header bursts is an $M/D/1/N_{max}$ queuing system, with Poisson arrival rate $\lambda$ and call duration $\mu^{-1}$.

We define $N_h$ to be the maximum number of 'giant' headers in the $M/D/1/N_{max}$ system such that, for a fixed probability $\epsilon$,

$$P(\text{number of requests in the system } \leq N_h) \geq 1 - \epsilon. \quad (9)$$

Since $N_{max}$ depends on the value of $\alpha_p$, so does $N_h$. Now, to minimize the delay with probability $\epsilon$, we need $N_h$ to satisfy:

$$\frac{N_h nH}{(1-\alpha_p)\beta_T} \leq \delta \quad (10)$$

so $\alpha_p$ solves the equation, where we explicitly write the dependency of $N_h$ over $\alpha_p$ as:

$$\begin{aligned} &\text{Find } \alpha_p^* = \max\{\alpha_p \in (0,1)\} \text{ s.t.} \\ &nHN_h(\alpha_p) + \alpha_p\beta_T\delta \leq \delta\beta_T \end{aligned} \quad (11)$$

The maximum number of concurrent connections is thus:

$$N_{max} = \frac{\alpha_p^*\beta_T}{\beta} \quad (12)$$

As before, when $\delta = 30ms, n = 3, H = 672bits, \beta = 10kbits/s$, a channel of size $1Mbits/s$ and an arrival rate of $200^{-1}$ calls per second, we compute the value $\alpha_p^*$ using the steps described above.

Figure 6 shows the behavior of the quantity $nHN_h(\alpha_p) - (1-\alpha_p)\beta_T\delta$. The maximum value satisfying the acceptable delay threshold $\delta$ is that value for which this increasing quantity is zero. The $y$-axis represents the normalized delay, which is delay minus $\delta$. A negative value means it is within the acceptable $\delta$ delay threshold. The payload utilization is the ratio $\alpha_p$ allocated to the payload. The value of $N_h$ is such that the probability to exceed the delay is less than 1% (see equations 11 and 12).

We can see that when increasing the payload ratio (i.e., greater value of $\alpha_p$), the delay also increases due to congestion involving the header bursts in the shared allocation channel. The graph indicates a tradeoff between better payload utilization at the expense of a longer delay (to establish compression contexts) for the packets.

In Figure 6, the optimal value of payload utilization is about 95%. In comparison, the corresponding value for CBR channel of the previous section is $\frac{1}{2.24} = 44.6$ %.

In Figure 7 we plot another graph when the value of $N_h$ is such that the probability to exceed the delay is less than 0.1% (see equations 11 and 12). The optimum value of payload utilization in this case is 67.5 %. More *guard* bandwidth has to be provisioned so as to avoid collisions with probability $\epsilon$.

Finally, Figure 8 plots the case when the mean delay needs to be less than the delay threshold. The optimal payload utilization in this case is about 96 %.

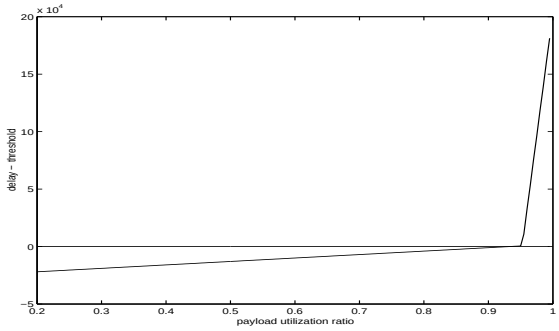


Fig. 6. Delay vs. payload utilization, $\epsilon = 1\%$

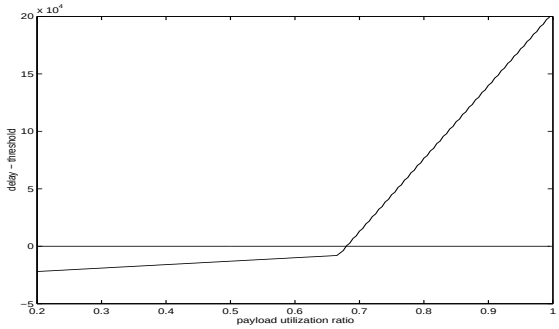


Fig. 7. Delay vs. payload utilization, $\epsilon = .1$ %

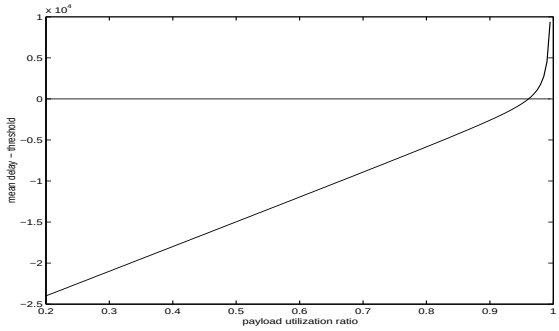


Fig. 8. Mean Delay vs. payload utilization

Dividing the wireless link in a payload plane and a bursty traffic plane yields an increased efficiency over the CBR allocation scheme. Yet, all calls in such a scheme offer some guaranteed quality of service, since the delay is bounded and the payload allocation is CBR, which offers flow isolation.

The cost of context establishment is now $(1 - \alpha_p^*)\beta_T$. Considering again Figure 7, $(1 - \alpha_p^* = 32.5$ %. For the 25 bytes payload size, it amounts to an effective 12 bytes header. The *effective* compression is not from 84 to 1, but from 84 to 12 in this case.

### B.3 Shared Channel

We focus now on a completely shared channel, with no distinction between the header bursts and the payload traffic. In this situation, corresponding to a Wireless LAN for instance, there

is no allocation of the resource to the users, and the users simply get the available bandwidth. We evaluate the performance of this allocation using ns-2 [15] simulations.

We consider a wireless link of capacity 300 kbits/s, and users requesting connections for calls of bandwidth $\beta = 24$ bytes per 20 ms plus the headers: 1 byte for second order, 4 bytes to first order and 84 bytes for the full header.

In Figure 9, we consider the packet loss probability. We set a threshold of 1% for packet loss beyond which the call is dropped due to degradation of the quality of service.

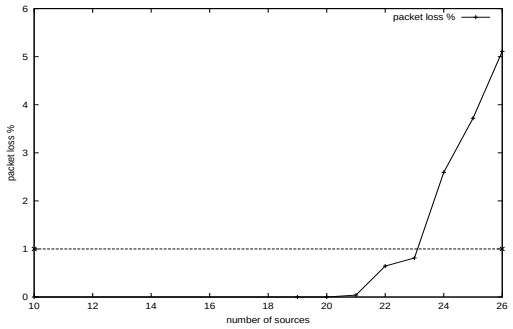


Fig. 9. Packet loss vs. number of concurrent sources

From Figure 9, we see that the maximum number of concurrent sources using HC which satisfies the packet drop requirements in this system is 23, and that the payload utilization for this value is:

$$\frac{23 * 9.6\,kbits}{300\,kbits} = 0.736 \tag{13}$$

A similar value can be obtained for the $\delta$ delay requirement. Under a different scenario in section IV-B.2 we found a range between 67.5 % and 95 % for different values of the probability $\epsilon$. However, the shared signaling offers some guarantees on the quality of service during a call by protecting one flow from the other once the HC is established.

The cost of context initiation is thus 26.4 % for a connection dropped over 1% of packet loss. The effective HC in this case can be computed as 9 bytes for a 24 bytes payload size. Again, the compression benefit of 1 byte is not achieved.

## V. Conclusion

We presented in this document an analysis of relative benefits of optimistic and acknowledgment-based HC context establishment procedures. We used this analysis to model the traffic pattern created by the HC context re-initialization and to study the impact of context initialization on channel allocation or CAC decision. We considered here typical voice over IP streams, and it would be of interest to consider other applications that will use HC.

We have shown in multiple scenarios that the cost of establishing the context reduced significantly the header compression benefits. In the three scenarios we looked at, even though the compressed header size was 1 byte, the effective compressed header size was between 9 and 31 bytes. The four-fold improvement expected from compressing a packet of overall size 109 bytes down to 26 turns out to be a two-fold or three-fold improvement depending on the situation. Needless to say, this is still a significant improvement.

A context transfer scheme could transfer the compression context from the previous router to the new router when the mobile node roams. By avoiding the context establishment, such a context transfer would thus be extremely valuable, both in terms of bandwidth saved, and in tighter channel allocation or CAC decisions. For instance, assume that $\lambda = \lambda_i + \lambda_c$ where $\lambda_i$ is the rate of calls that need to have the context initiated, and $\lambda_c$ are the calls for which the context is available. And, consider the scenario in section IV-B.1. The maximum number of concurrent connections is given by:

$$N_{max} = \frac{\beta_T}{\frac{\lambda_i}{\lambda}\beta(1+\alpha) + \frac{\lambda_c}{\lambda}\beta} = \frac{\beta_T}{(1+\frac{\lambda_i}{\lambda}\alpha)\beta} \tag{14}$$

The cost factor $\alpha$ is thus reduced by $\lambda_i/\lambda$. We find the savings due to context transfer particularly appealing, and wish to conduct further analysis on the topic.

## Acknowledgments

The authors would like to thank S. Balandin for his help with the simulations.

## References

[1] C. Bormann, editor. *Robust Header Compression*, Request For Comments, RFC 3095. Internet Engineering Task Force, July 2001.
[2] B. Buchanan *Evaluation of TCP/IP Header compression in ATM Networks* Masters Thesis, University of Iowa, Iowa City, 1994
[3] *Cooperative Association for Internet Data Analysis* http: www.caida.org
[4] E. Chlebus, W. Ludwin *Is Handoff Traffi c Really Poissonian?* ICUPC'95, Tokyo, Japan, Nov. 6-10, 1995, pp.348-53.
[5] P. Bhagwat, C. Perkins, and S. Tripathi *Network Layer Mobility: An Architecture and Survey* IEEE Communications Magazine, Vol. 34, No. 6, June 1996, pp. 54-64.
[6] M. Degermark, M. Engan, B. Nordgren, S. Pink *Low-Loss TCP/IP Header Compression for Wireless Networks* Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking (MOBICOM'96), 1996, pp.1-14.
[7] C. Huitema *IPv6: The New Internet Protocol* Prentice-Hall Inc., Oct. 1997.
[8] V. Jacobson, R. Braden, D. Borman. *Compressing TCP/IP headers for low-speed serial links* IETF, Network working group, RFC 1144, Feb. 1990
[9] J. Kempf, Editor Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network draft-ietf-seamoby-context-transfer-problem-stat-04.txt, IETF draft, work in progress
[10] R. Koodli and C. E. Perkins A Context Transfer Protocol for Seamless Mobility draft-koodli-seamoby-ct-03.txt, IETF draft, work in progress
[11] Z. Kostic, X. Qiu, and Li Fung Chang *Impact of TCP/IP header compression on the performance of a cellular system* Wireless Communications and Networking Conference, 2000. WCNC. Vol.1 pp. 281-6
[12] L. Kleinrock, *Queueing Systems, vol.1 and vol.2*, Wiley, New York, 1975.
[13] K. Le, C. Clanton, Z. Liu, and H. Zheng *Effi cient and robust header compression for real-time services* Wireless Communications and Networking Conference, 2000. WCNC. Vol.2 pp. 924-8
[14] S. McCreary and K.C. Claffy, *Trends in Wide Area IP Traffi c Patterns: A View from Ames Internet Exchange.* ITC Specialist Seminar on IP Traffi c Measurement, Modeling, and Management, Monterey, California, September 14, 2000.
[15] Network Simulator *ns-2* University of California, Berkeley, CA. 1997. Available via http://www.isi.edu/nsnam/ns/

[16] C. Perkins *Mobile Networking Through Mobile IP - Tutorial* IEEE Internet Computing, Vol. 2, No. 1, January/February 1998.
[17] H. Zeng, I. Chlamtac *Handoff Traffi c Distribution In Cellular Networks* Wireless Communications and Networking Conference, 1999, WCNC. vol. 1, pp. 413 -417, 1999.
[18] C. Westphal *A User-based Frequency-dependent IP Header Compression Architecture* To appear in IEEE Globecom 2002, Taiwan, 2002.

## APPENDIX

### I. COST OF RE-INITIALIZATION: A NUMERICAL EXAMPLE

Consider the case when $n = 3, m = 5$. See Figure 10 for the illustration. For a packet error $p$ of 10%, the decompressor can update the state after 5 packets with 99% confidence from equation 1. Assume that the RTT is 120 ms and the packet interarrival time is 20 ms. If the compressor waits for an acknowledgment to change compression state, then the earliest time it can do so is after the third packet is acknowledged, which is after 160 ms. By this time, it could have sent 9 packets. If the compressor does not wait for an acknowledgment to change compression state, then with a probability $P(3, 5) = 0.99$, it can change compression state after 80 ms, which is 4 packets earlier.

For a packet with 84 bytes in FH, 1 byte in CH, and 30 in the payload, the gain over waiting for the 3 packets to be acknowledged is at least 83 bytes times 4 packets, which is 332 bytes in 99% of the cases. In the remaining 1% of the time when the compressor changes state too early, it notices that it is in an erroneous state when it receives the acknowledgments. This would be no later than the 5th packet acknowledgment, at time 200 ms. Thus, the 6 CH packets sent between 80 ms and 200 ms are lost, wasting 6 * 31 bytes of bandwidth, and the compression context needs to be re-established. Assuming the compressor does not take a chance a second time and safely waits for the acknowledgment, then the overall loss due to optimistic context establishment is 6 CH packets (payload plus compressed header) and the 5 Full Headers of the initial attempt, which could have been sent as CH Headers had the compressor waited for the acknowledgments in the first place. Note that the payload part of the initial Full Header packets is not wasted, since those packets could be forwarded normally. It is only the Full Headers that are wasted since they fail to initialize the context. The total bandwidth cost in this case is thus 6*31 + 5*83 = 601 bytes. Figure 10 explains the different scenarios.

So, as far as bandwidth utilization is concerned, the gain is on the average:

$$332 \times 0.99 - 601 \times 0.01 = 322.67 \text{ bytes} \qquad (15)$$

per each packet stream. This is not insignificant whenever the air interface channel is allocated so as to give a throughput close to the payload throughput, as we will see below. However, it is also important to keep in mind that the cellular boundaries typically exhibit higher BER and thus the optimistic approach is susceptible to higher occurrences of "jump ahead" penalty.

The cost of the Acknowledged context initialization is of 9 * 83 = 747 bytes. Thus, in the case of a handover, we have:

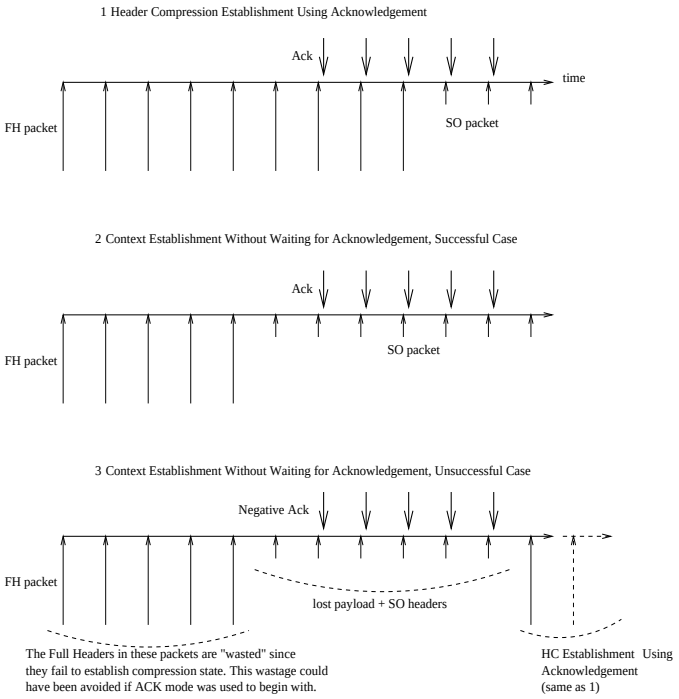


Fig. 10. Acknowledged vs. UnAcknowledged Header Compression Establishment

- if the HC context is transferred using a context transfer protocol between successive access routers, there is no overhead: the mobile node transmits all the time with CH headers.
- if the acknowledged initialization is used, then it consumes an extra 747 bytes.
- if the optimistic initialization is used, then it consumes an extra 424 bytes (or 323 less than the acknowledged initialization).