Wireless Personal Communications (2006) 38: 203–232
DOI: 10.1007/s11277-005-9012-7

Performance Analysis of Header Compression Schemes in Heterogeneous Wireless Multi–Hop Networks

PATRICK SEELING[1], MARTIN REISSLEIN[1], TATIANA K. MADSEN[2], and FRANK H.P. FITZEK[2]

[1]*Patrick Seeling and Martin Reisslein are with the Department of Electrical Engineering, Arizona State University, Goldwater Building MC 5706, Tempe, AZ 85287-5706*
Email: {patrick.seeling,reisslein}@asu.edu, web: http://www.seeling.org, http://www.fulton.asu.edu/∼mre/.
[2]*Tatiana K. Madsen and Frank H.P. Fitzek are with the Department of Communication Technology, University of Aalborg, Denmark*

Abstract. Wireless multi–hop networks are becoming more popular and the demand for multimedia services in these networks rises with the number of their implementations. Header compression schemes that compress the IP/UDP/RTP headers to save bandwidth for multimedia streams were typically evaluated only for individual links, not taking into account the savings that can be achieved using header compression over a complete path. In this paper, we evaluate the performance of three categories of header compression schemes: (*i*) delta coding, (*ii*) framed delta coding, and (*iii*) framed referential coding. We evaluate the performance for these schemes on reliable and unreliable links. We then extend our evaluations to several links constituting a path. As nodes in multi–hop ad-hoc and mesh networks may differ with respect to their capabilities, we assume in our evaluation that (forwarding) nodes may not be able or choose not to perform header compression. We find that the framed referential header compression scheme is the most suitable scheme in case that no or long-delay feedback channels exist. We additionally compare the packet drop savings due to header compression and the combined savings of compression and drops. We again find that the framed referential coding scheme exhibits good performance that can lead to significant header compression and packet drop savings for reasonable bit error rates.

Keywords: Header compression, wireless, multi–hop, link savings, path savings, mesh networks

1. Introduction

Providing network services to users by using multi–hop wireless connections is becoming evermore popular. In multi-hop wireless networks, communication between two nodes is carried out through a number of intermediate nodes. In the last few years, many research activities have been focused on multi-hop ad hoc networks, consisting of mobile nodes whose operation does not require infrastructure support. At the same time, an increasing number of multi-hop wireless deployments and proprietary commercial solutions gave rise to a class of networks known as mesh networks. Mesh networks serve as access networks to send traffic to and from the wireline Internet in a multi–hop fashion. The intermediate nodes will typically play a role of static relays.

Provisioning of high network capacity is a critical point for multi–hop access networks. The bandwidth in such networks is often limited, as nodes within the network carry the burden of forwarding (routing) packets that are not destined for them but to other nodes in the network. As wireless multi–hop networking grows in its usage, the services requested by users are extending and become more sophisticated. With continuously growing device capabilities,

users demand services similar to those accessible on wired and infrastructure-based wireless networks.

A challenging domain in wired and wireless networks alike is multimedia networking. Multimedia data typically has higher requirements towards the network in order to be of use at the receiving client. Multimedia applications comprise several different application scenarios, including audio, video, and gaming. Most of these services are real-time or, with some intermediate buffering, close to real-time. In real-time services such as voice over IP and multimedia streaming applications, the overhead introduced by the various protocol layers can contribute significantly to the overall packet sizes. Moreover, in these end-to-end oriented services the achievable delay and delay jitter are of higher importance than in typical data transmission oriented services. Multimedia applications often use RTP, UDP, and IP as protocols. Each of the protocol layers adds header overhead due to the ISO/OSI layer approach of protocol encapsulations. As compression of the payload is typically performed to an excellent level on the application layer, an additional source for reducing the packet sizes of packets that have to be forwarded by intermediate nodes through a multi-hop network are the different protocol encapsulation overheads. An additional benefit that is derived by compression of the packet sizes in general is the reduction in errors that occur during the transmission, as smaller packets are less error prone.

Several different header compression schemes have been introduced in the past, mostly in the wired domain. The most well-known header compression schemes include the IP header compression scheme (CTCP or VJHC) by Van Jacobson, introduced in 1990 [1] as RFC 1144. In [2], robustness at the cost of less compression was introduced by Perkins and Mutka. The robustness is achieved by introduction of a reference header that is updated after several packets, which results in a frame of dependent packet headers. IP header compression (IPHC) [3] provides a number of extensions to VJHC. The most important extensions are support for UDP, IPv6, and additional TCP features. With the explicit support of UDP come additional features, such as multicast. The Compressed Real Time Protocol (CRTP) scheme presented in RFC 2508 [4] compresses the 40 bytes header of IP to 4 bytes if the UDP checksum is enabled, or to 2 bytes if it is not. This is possible by compressing the RTP/UDP/IP headers together, similar to the VJHC approach. Most of these header compression schemes were designed for wired networks with low bit error rates and thus perform poorly under challenges imposed by wireless links. Recently, Robust Header Compression (ROHC) [5] has been standardized and is currently part of the 3G standard, as well as subject to several efforts to further enhance its usability in several different application scenarios, see, e.g., [6].

As header compression schemes move into the single wireless link domain, it becomes obvious that extending the single-hop based research efforts into the multi–hop domain are needed in order to evaluate the suitability of header compression in mobile ad-hoc and mesh multi–hop networks. For single unreliable links, several studies examined the impact of link errors on header compression and possible remedies, see e.g., [2], [7], [8], [9]. Evaluating the header compression schemes for unreliable multi–hop scenarios has attracted relatively little research interest so far. A first approach was made by the authors of [10], who extended ROHC into the mobile ad-hoc network domain using header-compression aware routing. Also, in a recent Internet draft, the applicability of ROHC in the IEEE 802 protocols has been subject to a call for research [11]. In this contribution, we take a different approach and compare different header compression techniques in general with respect to their performance on single and multiple reliable and error-prone links.

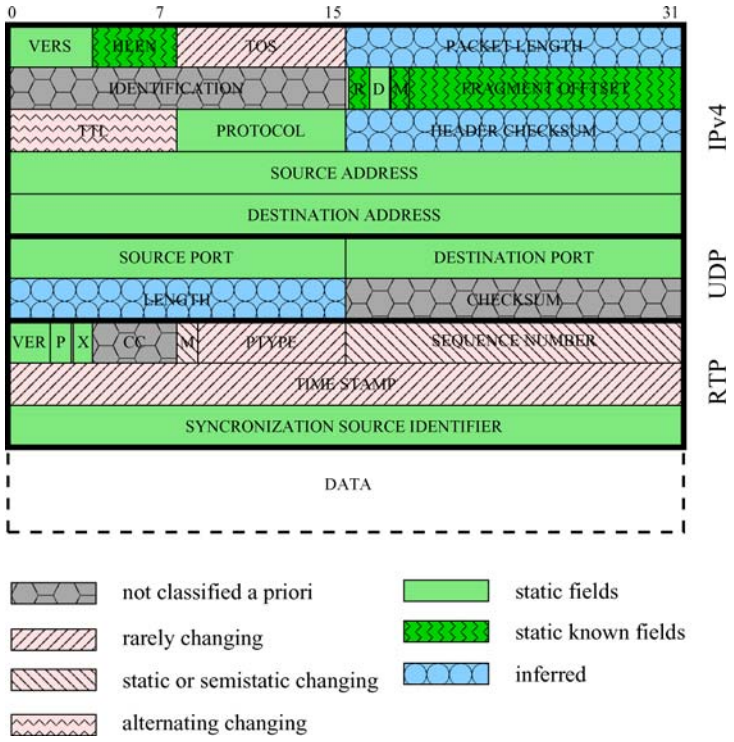

Figure 1. IPv4, UDP, and RTP headers with redundancies.

2. Generic Header Compression System

In this section, we describe our general system under consideration. In particular, we first give an overview of the location of header compression in the protocol stack, before we extend our view from an individual link to a path.

Header compression is typically performed on the headers of the network layer and above. Typically, for multimedia, i.e., real-time based, services, header compression is performed on the IP/UDP/RTP headers. The applied compression typically exploits the redundancy within the IP/UDP/RTP headers of individual packets as well as for consecutive packets that belong to the same stream. We illustrate the different header redundancies of a combined IP/UDP/RTP protocol encapsulated packet in Figure 1. As we can see from the different headers, the redundancy is quite high; it would be even higher if we considered IPv6, as in IPv6 the number of bytes used for static information is larger.

On an individual link, which is typically the area where header compression is performed and for which header compression schemes are evaluated, the packet headers are compressed at the sending node. The receiver has a decompressor that reconstitutes the original header before delivering the packet to the higher protocol layers. As illustrated in Figure 2, the location of the compressor/decompressor is between the link layer and the network layer. On the sender side the compressor removes redundancy from the incoming packet with respect to a reference (base) header. This reference is also known (and maintained) at the receiver and allows the receiver to decompress the incoming compressed packet headers.

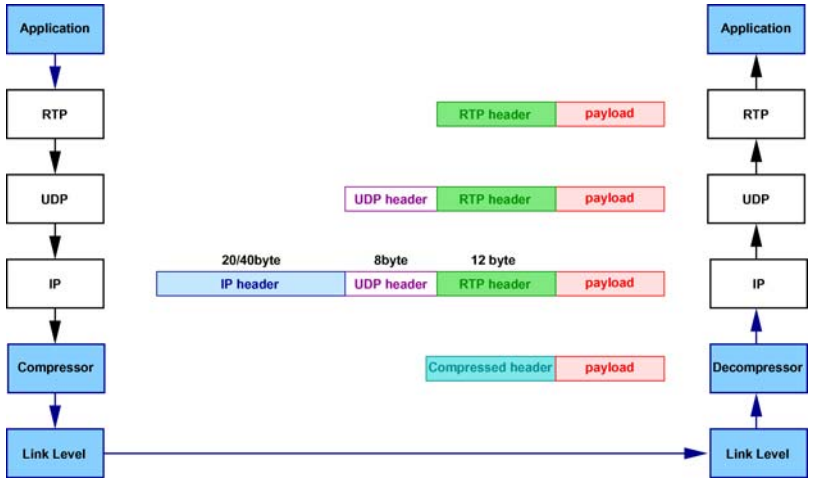

Figure 2. Protocol header encapsulation overhead and location of header compressor/decompressor.

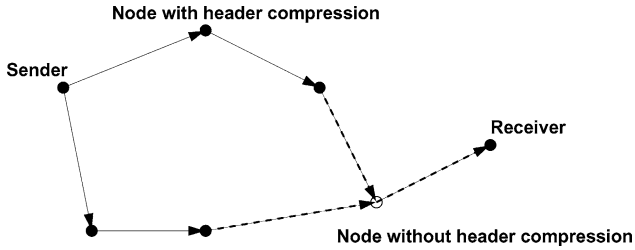

Figure 3. General setup of heterogeneous multi–hop environment with header compression enabled and disabled nodes.

Header compression schemes have been evaluated typically for individual links, as they require that sender and receiver share the same compression scheme (and in practice often also the same implementation thereof). It is expected, however, that in multi-hop mesh and ad–hoc networks nodes are not homogeneous with respect to their capabilities. In heterogeneous wireless environments, the intermediate nodes along a path from the sender to the receiver can greatly vary with respect to their offered services to the overall network. In particular, wireless nodes may disable features such as header compression, enter into sleep states, or not be able to perform advanced services such as header compression at all. We illustrate an exemplary combination of links that form a path from the sender to the receiver with nodes that perform header compression and nodes that do not in Figure 3. As illustrated, we consider an environment where each node may or may not have header compression enabled. We assume throughout this paper that the individual nodes share a single header compression scheme.

We consider an end-to-end delivery scenario where the sender streams data to the receiver over multiple intermediate nodes. If a node on the path from the sender to the receiver is unable to perform header compression, all links to and from that node are considered to be with uncompressed headers. The nodes before this node without header compression have thus to decompress the header and transmit the packet uncompressed. The node following to the node without header compression then has to re-compress the header.

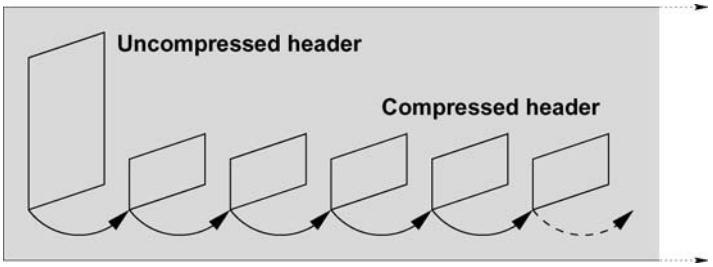

Figure 4. Delta coding header compression scheme, where following an uncompressed header delta coded compressed headers of the same size are sent.

3. Header Compression Schemes

In this section, we review typical header compression schemes. We perform our review based on our previous results with an actual implementation of robust header compression. For other header compression schemes, we assume a similar compression of the packet headers.

We assume for the simple case of a delta coding header compression scheme among consecutive packet headers that the initial header is sent uncompressed, and consecutive IP/UDP/RTP headers are compressed to 6 byte each, as illustrated in Figure 4. This assumption is corroborated by our earlier observations for robust header compression (ROHC) in [12], where we typically found average header sizes around 6 byte. Similar results were obtained in [13] for ROHC in the u-mode (without feedback channel, unidirectional link). The delta coding header compression scheme requires the decompressor to request an update from the compressor whenever decompression errors occur. To do so, TCP retransmissions can be used as introduced in [1]. The downfall of this approach is that TCP retransmissions only occur after a certain number of packets were dropped by the decompressor. The second approach would require a feedback channel from the decompressor to the compressor. Feedback packets, however, may not be received in case of highly bursty or error-prone links. For these types of links it is thus advisable to use an automatic update of the reference header at fixed or dynamic, e.g., with respect to the link state, intervals to ensure that the compressors and decompressors on each hop remain synchronized. (We note that we only consider fixed update intervals in this paper. Additionally, link state-based approaches can be used.) This automatic update approach results in frames of dependent compressed headers, whereby multiple compressed headers are send after an uncompressed header. We illustrate the framed delta coding header compression scheme in Figure 5. As illustrated, the initial header is sent uncompressed and consecutive headers in each frame are delta coded to the preceding headers. For our numerical analysis,

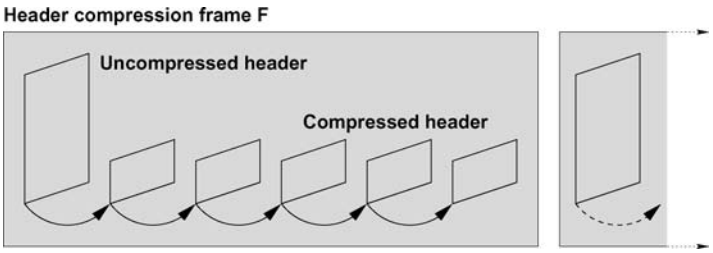

Figure 5. Framed delta coding header compression scheme, where after an uncompressed header compressed headers of the same size are sent until the end of the frame.

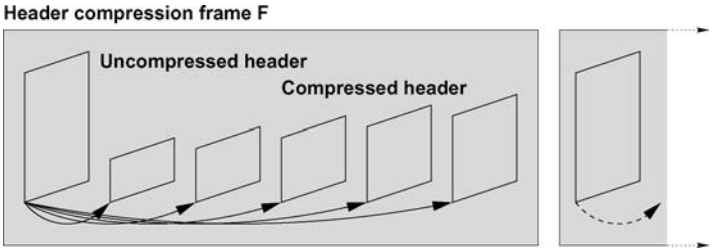

Figure 6. Framed referential coding header compression scheme, where after an uncompressed header compressed headers of increasing size are sent until the end of the frame.

we assume that the delta compressed headers are each 6 byte long. Further increasing the robustness, referential coding to the initial header within a given frame can be used to minimize the impact of individual errors as illustrated in Figure 6. For this framed referential coding header compression scheme, we assume that the initial delta coded header has a size of 6 byte and each subsequently encoded header results in an overhead of one additional byte.

4. Header Compression Performance for Links

Let C_i denote the size of the compressed header for packet i sent from the sender (compressor) to the receiver (decompressor). In case that an initialization header is sent, either at the beginning of a stream of packets or at the beginning of a frame as outlined in Section 3, the size of C_i equals the size of the uncompressed header denoted by U. As we consider compression of IP/UDP/RTP headers, we let $U = 60$ byte for IPv6. For delta coded headers, we assume a size of $C_i = 6$ byte, whereas we assume that referential coded headers incur an overhead of 1 byte with increasing distance from the initial reference header.

4.1. Header Compression Performance on Reliable Link

The header compression savings S can be calculated depending on the compression scheme used. In general, the header compression savings can be calculated as

$$S = \frac{F \cdot U - \sum_{i=1}^{F} C_i}{F \cdot U} = 1 - \frac{\sum_{i=1}^{F} C_i}{F \cdot U} = 1 - \frac{U + \sum_{i=2}^{F} C_i}{F \cdot U}, \tag{1}$$

where F denotes the length of the frame used in the compression scheme under consideration. In case that no framing is used, the initial uncompressed header becomes negligible in the long run and the compression can be calculated as

$$S_{\Delta} = 1 - \frac{C}{U}. \tag{2}$$

For delta-based coding within frames as illustrated in Figure 5, the compression can be simplified to

$$S_{F\Delta} = \frac{F-1}{F} \cdot \left(1 - \frac{C}{U}\right). \tag{3}$$

We illustrate the header compression savings as a function of the frame length F in Figure 7. We observe that the header compression savings are always larger for the frame using delta

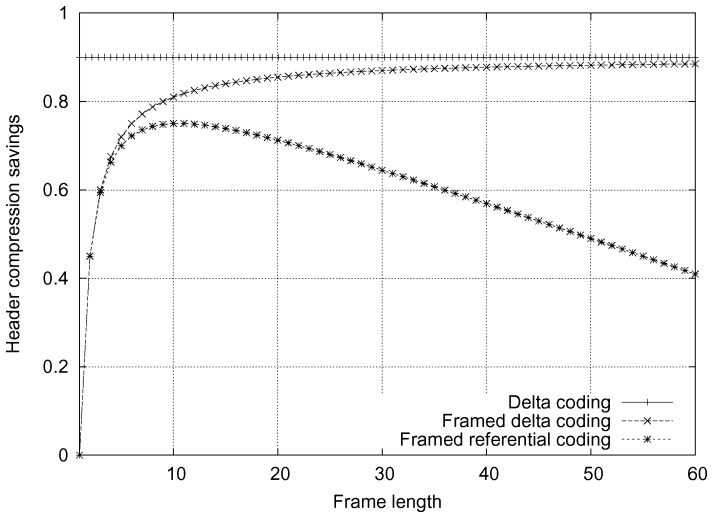

Figure 7. Header compression savings S for different compression schemes as function of the frame length F for a reliable link.

coding, which approach the maximum obtainable header compression savings of 0.9. The frame with referential coded headers achieves maximum header compression savings of 0.75 for frame lengths $F = 10$ and $F = 11$. After reaching the maximum obtainable compression savings for these two frame lengths, the header compression savings drop and become zero for a frame length of $F = 110$ packets.

4.2. Header Compression Performance on Unreliable Link

For a general evaluation, let us assume that we can model each link or hop using uncorrelated bit errors. Uncorrelated bit errors is a useful approximation if a heavy interleaving is applied. For UMTS channels this behavior was found in [14], [15]. For other channel types, this still provides a useful overview of the behavior of different header compression schemes. In particular the Gilbert/Elliot channel model could be approximated by using the steady-state behavior to allow a more general view of the impacts of channel errors and to derive a mean bit error rate [16].

We denote the probability of a bit sent and received successfully as P^g_{bit}. The corresponding probability of an error within x byte (as we use a byte-wise view throughout this paper) is thus calculated as

$$P^e(x) = 1 - (P^g_{bit})^{8 \cdot x} = 1 - P^g(x). \tag{4}$$

The probability of a whole packet consisting of X_i bytes payload and CH_i bytes header information being incorrectly transmitted is subsequently given as

$$P^e(X_i + CH_i) = 1 - (P^g_{bit})^{8 \cdot (X_i + CH_i)}. \tag{5}$$

For the transmission of real-time data, however, it is mostly desirable to deliver packets even in case of payload errors and to leave the error handling and concealment mechanisms to the application layer. This can be achieved, e.g., by using UDP-lite instead of UDP: the checksum is covering only the header. We therefore derive the long-term probability of only the (compressed) headers being transmitted erroneously and causing a loss of synchronization between

compressor and decompressor, i.e., we only consider $P^e(C_i)$. For performance evaluation, we consider the header compression savings and the savings for the number of dropped packets due to header compression errors in each compression scheme under the influence of link errors.

1) Delta Coding Only: For the delta coding scheme, we assume that the compressor has to send a packet with an uncompressed header to the decompressor whenever a compressed header suffers from a transmission error. The decompressor has to have a means of informing the compressor either directly or indirectly of the loss of synchronization, e.g., for bidirectional links, send a notification packet to the compressor. This closely resembles header compression schemes such as robust header compression or IP header compression. We furthermore assume that when a compressed header is hit by a transmission error, the decompressor has to discard several packets until the compressor is notified and can send a packet with an uncompressed header to resynchronize the decompressor. We denote the number of these dropped packets by d. Let the random variable γ denote the number of compressed headers that can be sent before a header suffers from erroneous transmission. This expectation can then be calculated as

$$E[\gamma] = \left\lfloor \frac{1}{P^e(C)} \right\rfloor. \tag{6}$$

In other words, we expect that every $E[\gamma]$ packets, a header decompression error will occur causing the decompressor to lose synchronization until the compressor can send an uncompressed header to restore synchronization.

The header compression efficiency with this scheme is therefore reduced by the number of erroneous compressed headers for which we have to send uncompressed headers. Assuming that the decompressor drops d packets in a row after not being able to decompress a header, the compression efficiency is actually reduced as the compression of these packets is of no use at the decompressor. We therefore calculate the expected header compression savings for the delta coding scheme as

$$\begin{aligned} E[S_\Delta] &= 1 - \frac{U + \sum_{i=2}^{E[\gamma]-1} C_i + U + \sum_{i=E[\gamma]+1}^{E[\gamma]+d} U}{(E[\gamma]+d)\cdot U} \\ &= 1 - \frac{(E[\gamma]-2)\cdot C + (d+2)\cdot U}{(E[\gamma]+d)\cdot U} \end{aligned} \tag{7}$$

We illustrate the expected header compression savings as function of the number of dropped packets d for several bit error rates in Figure 8. We observe that for very low bit error rates the expected header savings are close to the reliable link values illustrated previously in Figure 7. As the bit error rate increases, the expected header compression savings drop to 0. The rate of decline in the expected savings is additionally influenced by the number of packets that are dropped until the decompressor can resynchronize to the compressor. The more packets are lost during the re-synchronization phase, the less expected header compression savings can be obtained. For low bit error rates, we observe that the expected header compression savings drop linearly with the number of dropped packets d.

The expected header compression savings do not take the impact of the unreliable link on the loss of packets due to an employed header compression scheme into account. We therefore calculate the expected packet drop savings due to an employed header compression scheme in

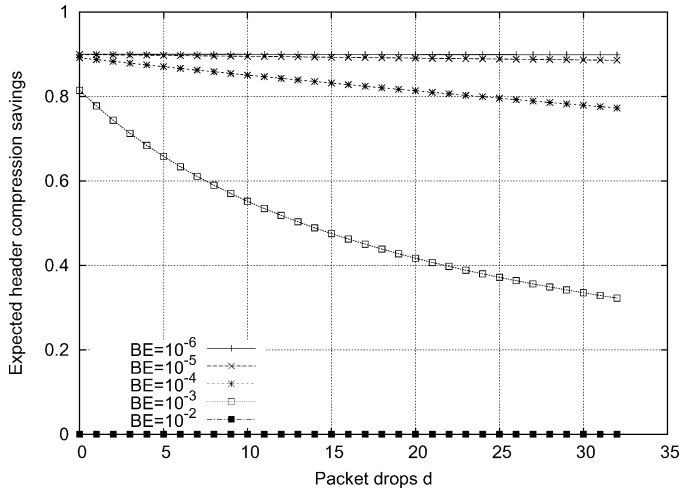

Figure 8. Expected header compression savings for delta coding $E[S_\Delta]$ with different numbers of packet drops after erroneous header d and different bit error rates BE.

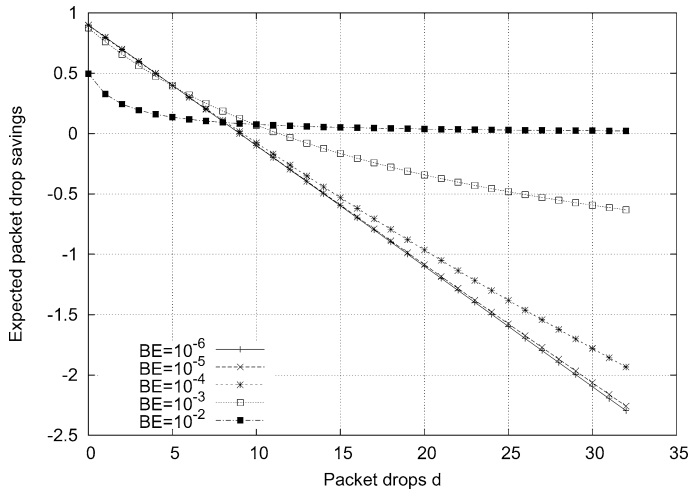

Figure 9. Expected packet drop savings due to header compression for delta coding $E[D_\Delta]$ with different numbers of packet drops after erroneous header d and different bit error rates BE.

addition to the header compression savings. We derive the expected savings in packet drops for the delta coding header compression scheme as

$$E[D_\Delta] = 1 - \frac{d+1}{(E[\gamma]+d)\cdot P^e(U)}. \tag{8}$$

We illustrate the percentage of expected saved dropped packets as second header compression performance metric in Figure 9 for the delta coding scheme. We observe that for low bit error rates the expected packet drop savings due to header compression sharply decline with the number of packets needed for re-synchronization of the decompressor. Only for higher bit error rates the expected packet loss savings decline less than linear. The reason for this behavior is that the uncompressed headers on links with higher bit error probabilities are highly error prone and thus the losses due to the smaller header sizes are always less than for the

compressed full size header. In most applicable bit error regions, however, we observe that savings in terms of packet losses are consumed by the re-synchronization packet drops if it takes nine or more packets to update the decompressor. In summary, we note that if only few packets have to be dropped by the decompressor, this scheme achieves high expected header savings. For updates that require a longer time to reach the decompressor, this scheme rapidly drops in performance.

2) Framed Coding: Frame-based coding in header compression schemes allows a compressor to frequently update the decompressor automatically and does not require a feedback channel or indirect notification of the compressor in case of header errors. In framed coding schemes, the compressor only sends (uncompressed) reference headers at the beginning of each frame. In the following we look closer at the performance of header compression schemes with delta-coded frames and referential-coded frames.

a) Framed Delta Coding: In case the initial reference header of a frame is lost, the whole frame is assumed to be discarded by the decompressor. For framed delta coding, we calculate the header compression savings with respect to the frame length F for $F \geq 2$ and the position of the erroneous header within the frame h as

$$S_{F\Delta}(h) = \begin{cases} 1 - \frac{C}{U} & h \geq 2, \\ 0 & h = 1. \end{cases} \tag{9}$$

and note that for the initial header $S_{F\Delta}\ (h = 1) = 0$. We calculate the probability of header h within the frame of F headers being transmitted correctly as

$$P^g_{F\Delta}(h) = \begin{cases} P^g(U) \cdot P^g(C)^{h-1}, & h \geq 2, \\ P^g(U), & h = 1. \end{cases} \tag{10}$$

The expected header savings for the framed delta coding scheme are subsequently calculated as

$$\begin{aligned} E[S_{F\Delta}] &= \frac{1}{F} \cdot \sum_{h=2}^{F} p^g_{F\Delta}(h) \cdot S_{F\Delta}(h) \\ &= \frac{1}{F} \cdot \sum_{h=2}^{F} p^g(U) \cdot P^g(C)^{h-1} \cdot \left(1 - \frac{C}{U}\right) \\ &= \frac{P^g(U)}{F} \cdot \left(1 - \frac{C}{U}\right) \cdot \sum_{h=2}^{F} p^g(C)^{h-1}, \end{aligned} \tag{11}$$

whereby we note that the expected header savings for the first header can be neglected as $S_{F\Delta}(1) = 0$.

We illustrate the expected header compression savings for different frame lengths and bit error rates in Figure 10. We observe that the expected header compression savings approach the theoretical upper bound as the length of the frame F increases for low bit error rates. We furthermore note that the expected savings rise with the frame length and decline after a maximum is reached at a frame length of $F = 7$ frames for a bit error rate of 10^{-3}. With the dependency between consecutive headers, the expected header compression savings here indicate that it would be beneficial to increase the frame length as the bit error rate declines, as for very large frames, we would reach the upper bound of simple delta coding. This, however neglects the effect of the packet drops that occur due to the inter-header dependencies. To determine the impact of this header compression scheme on the packet drops due to headers

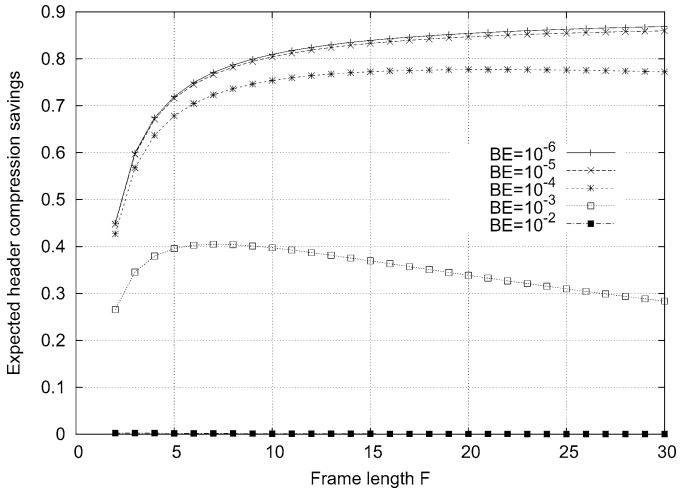

Figure 10. Expected header compression savings for framed delta coding header compression scheme $E[S_{F\Delta}]$ with different frame lengths F and different bit error rates BE.

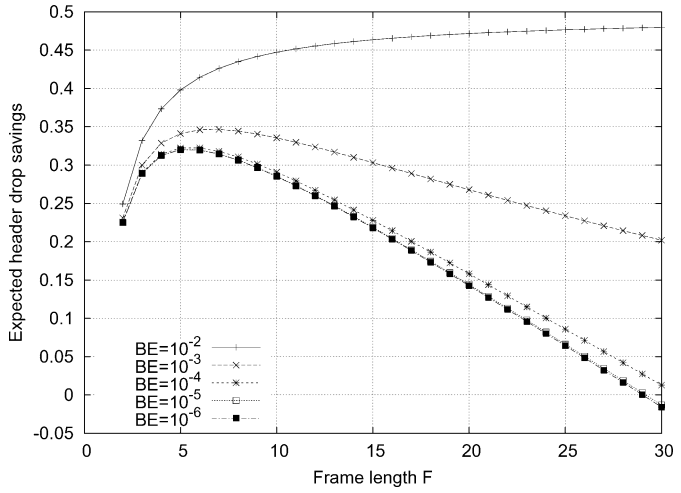

Figure 11. Expected packet drop savings for framed delta coding header compression scheme $E[D_{F\Delta}]$ with different frame lengths F and different bit error rates BE.

that cannot be decompressed, we first determine the probabilities of headers not reaching the receiver's decompressor correctly as

$$P^e_{F\Delta}(h) = \begin{cases} 1 - P^g(U) \cdot p^g(C)^{h-1}, & h \geq 2, \\ 1 - P^g(U), & h = 1. \end{cases} \tag{12}$$

The expected packet drop savings due to header compression are then calculated as

$$E[D_{F\Delta}] = 1 - \frac{1}{F} \cdot \frac{F \cdot P^e(U) + \sum_{h=2}^{F} P^e_{F\Delta}(h) \cdot (F - h + 1)}{F \cdot P^e(U)}. \tag{13}$$

We illustrate the corresponding packet drop savings for the framed delta header compression scheme in Figure 11. We observe that the savings over the uncompressed case for framed delta coding exhibit a peak for all bit error rates. For a bit error rate of 10^{-3}, the peak is located at

$F = 7$ headers in a frame. As the bit error rate decreases, the peak shifts from $F = 6$ headers in a frame for 10^{-4} to $F = 5$ headers in a frame for bit error rates of 10^{-5} and below. This is explained by the lower bit error rates resulting in less errors in the uncompressed headers. The longer the frame length F, on the other hand, the higher the losses due to the intra-frame dependencies of headers as given by the delta coding approach. Overall, however, we observe that the framed delta header compression scheme results in expected packet drop savings of approximately one third compared to the uncompressed case.

b) Framed Referential Coding: For determination of the framed referential coding header compression scheme, we use the approach outlined for the framed delta coding scheme. We note that for the framed referential scheme, however, we cannot simplify over the individual headers within a given frame, but have to consider them individually due to the assumed compression overhead. We also note that since each header does only rely on the initial (uncompressed) reference header of a frame, the savings for each header are given independently from the other compressed headers within the frame (in opposite to the framed delta coding approach, where the decoding of a header relies on the successful reception of all previous headers in a given frame). We calculate the header savings for the individual headers in the referential coded frame as

$$S_{FR} = \begin{cases} 1 - \frac{C_h}{U}, & h \geq 2, \\ 0, & h = 2 \end{cases} \tag{14}$$

The probability for compressed header h within the frame of being delivered correctly is given as

$$P^g_{FR}(h) = \begin{cases} P^g(U) \cdot P^g(C_h), & h \geq 2, \\ P^g(U), & h = 1. \end{cases} \tag{15}$$

We can now determine the expected header compression savings for framed referential coding schemes as

$$\begin{aligned} E[S_{FR}] &= \frac{1}{F} \cdot \sum_{h=2}^{F} p^g_{FR}(h) \cdot S_{FR}(h) \\ &= \frac{P^g(U)}{F} \cdot \sum_{h=2}^{F} P^g(C_h) \cdot \left(1 - \frac{C_h}{U}\right), \end{aligned} \tag{16}$$

which is derived by noting again that the savings for the initial header are $S_{FR}(1) = 0$. We illustrate the expected header compression savings for different frame lengths and bit error rates in Figure 12. We observe that as for the reliable case, the framed referential coding exhibits a strong peak in the header compression savings as function of the frame length. Closer observation shows that for the different bit error rates this peak is independent of the bit error rates at a frame length of $F = 10$ headers or packets per frame given the 1 byte increase for C_h with $h \geq 3$ (in case of $h = 2$ the simple delta coding size of $C_2 = 6$ byte applies). As for the case of framed delta coding, we note that the high bit error rates yield low compression savings, whereas for lower bit error rates, the compression savings are very close to another and the maximum header compression savings that were obtainable on a reliable link.

Similar to the other header compression schemes, we compare the number of dropped headers for framed referential coding with the number of dropped headers in the uncompressed

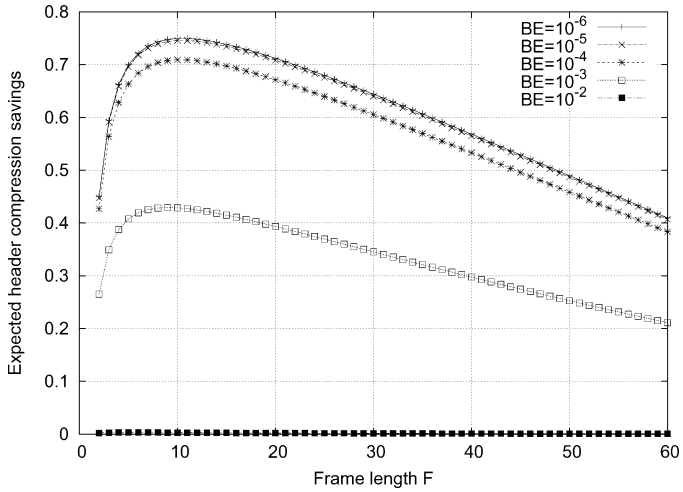

Figure 12. Expected header compression savings for framed referential coding header compression scheme $E[S_{FR}]$ with different frame lengths F and different bit error rates BE.

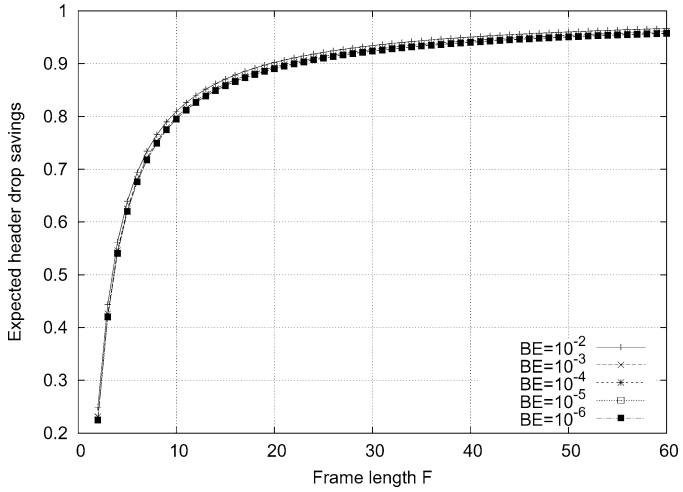

Figure 13. Expected packet drop savings for framed referential coding header compression scheme $E[D_{FR}]$ with different frame lengths F and different bit error rates BE.

scheme. We calculate the error probability for header h as

$$P^e_{F\Delta}(h) = \begin{cases} 1 - P^g(U) \cdot P^g(C_h), & h \geq 2 \\ 1 - P^g(u), & h = 1. \end{cases} \tag{17}$$

The expected packet drop savings are calculated as

$$E[D_{FR}] = 1 - \frac{1}{F-1} \cdot \frac{F \cdot 1 - P^e(U) + \sum_{h=2}^{f} 1 - P^g(U) \cdot P^g(C_h)}{F \cdot P^e(U)}. \tag{18}$$

We illustrate the expected packed drop savings in Figure 13. We observe that the expected dropped packet savings for the framed referential coding scheme approach 1 for longer frame lengths. Similar to the expected dropped packet savings for the framed delta coding compression scheme, the framed referential coding scheme always results in an overhead of dropped packets due to the reference header which, if erroneous, results in loss of the whole frame.

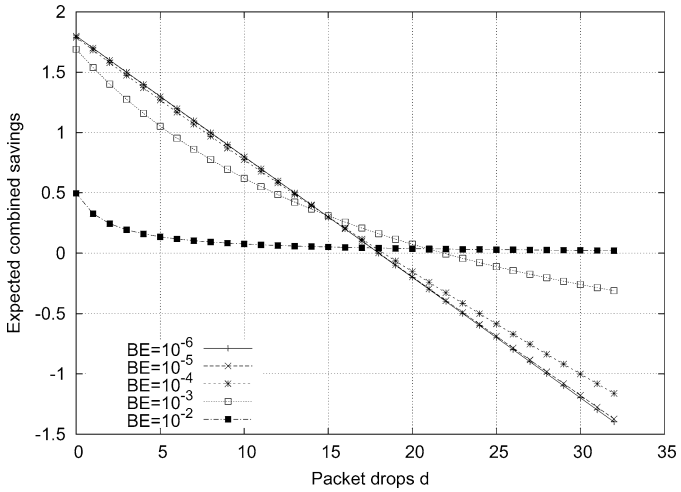

Figure 14. Expected combined savings for delta coding header compression scheme with different packet drops d and different bit error rates BE.

As additional result, we observe that therefore the expected packet drop overheads stay close to each other for different bit error rates. The framed referential coding header compression scheme thus clearly outperforms the framed delta coding header compression scheme.

4.3. Expected Combined Savings

In this section, we compare all three header compression schemes with respect to their combined performance. In particular, we consider the expected combined savings due to header compression and packet drop savings as $E[S] + E[D]$. This is motivated by header compression savings being rendered useless if too many packets are dropped, as especially for real-time data, the user experience would be very negative.

We illustrate the expected combined savings in Figures 14, 15, and 16 for the delta coding, framed delta coding, and the framed referential coding header compression schemes, respectively. We observe that the expected combined savings decline for the delta coding scheme until they become negative between $d = 18$ and $d = 22$ packets needed for re-synchronization of the decompressor, depending on the bit error rate. For extremely high bit error rates, the savings remain positive, albeit close to 0. For the framed delta coding header compression approach, the expected combined savings rise, reach a maximum, and decline afterwards. The maximum is located at a frame length of $F = 7$ headers for a high bit error rate of 10^{-3}. For bit error rates of 10^{-4} and below, the peak in the expected combined savings is attained at $F = 9$ headers constituting a frame. We additionally observe that the expected combined savings attained for the lower bit error rates are above 100%. The expected combined savings for the framed referential header compression scheme exhibit a similar behavior of rising, attaining a maximum, and declining afterwards. We observe that the level of savings remains above the framed delta coding scheme for all examined frame lengths. For a high bit error rate of 10^{-3}, the peak is attained at a frame length of $F = 21$ headers. For lower bit error rates, the framed referential approach attains the maximum savings, independent of the bit error rate, at a frame length of $F = 19$ headers. We furthermore observe that the expected combined savings for bit error rates of 10^{-3} and below are more than 125%.

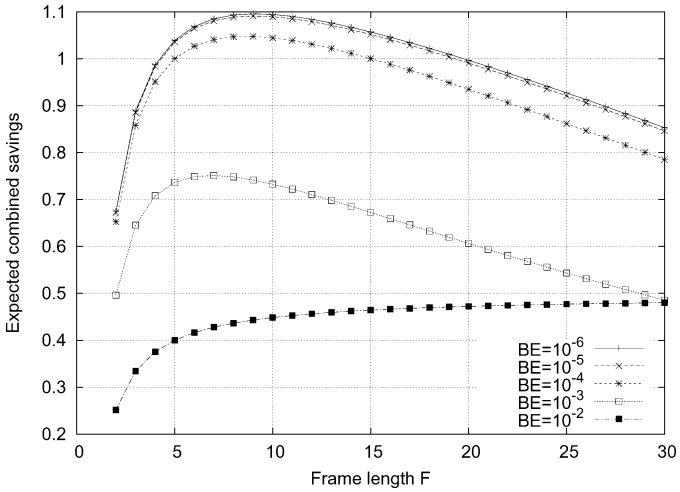

Figure 15. Expected combined savings for framed delta coding header compression scheme with different frame lengths *F* and different bit error rates *BE*.

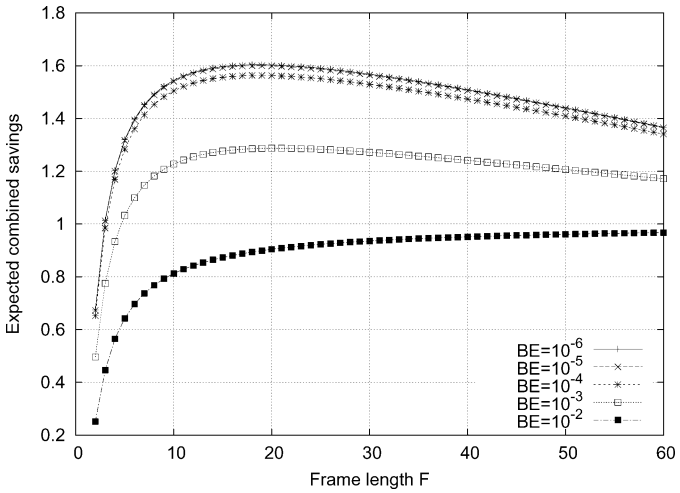

Figure 16. Expected combined savings for framed referential coding header compression scheme with different frame lengths *F* and different bit error rates *BE*.

Depending on the availability of a feedback channel and the bit error rate on the link, the header compression scheme can be selected. Our results indicate that the delta coding header compression scheme should be favored for little re-synchronization times (small number of dropped packets *d*). For the case of unidirectional links, or if feedback incurs long delays (and thus a large number of packet drops *d* at the decompressor), framed referential coding should be used.

5. Header Compression Performance for Paths

In this section, we extend our previous results obtained for single links to several consecutive links connecting a sending node to a receiving node. Intermediate nodes along such path can

either have header compression enabled or may not be able to perform header compression at all, as illustrated in Figure 3.

5.1. Reliable Links

To derive the expected path header compression savings for a given path, let us assume that we have N nodes along the path which constitute $L = N - 1$ links. Each node n has a probability of having header compression enabled P_n^c or disabled $P_n^u = 1 - P_n^c$, which we assume is uniformly distributed among the nodes. We furthermore assume without loss of generality that the nodes all utilize the same header compression scheme and that each link has the same bit error rate. (We note that in a similar manner, nodes that are able to use different header compression schemes, but are not able or willing to transcode between them can be regarded. In addition, we also note that the case of of applied header compression during a fraction of the time a path exists, e.g., due to applied power saving schemes, can be regarded in a similar manner.) We denote the probabilities of header compression being performed or not on a particular link l along the path to the receiving node with P_l^c and P_l^u, respectively. We derive

$$P_l^c = P_n^c \cdot P_n^c = (P_n^c)^2 \tag{19}$$

as probability of performed header compression on a link and

$$P_l^u = 1 - P_l^c = 1 - (P_n^c)^2 \tag{20}$$

as probability of sending headers without compression over a link. Let S_l denote the header savings achieved on a link using any header compression scheme outlined in Section 4. The expected header savings on a path consisting of L links are calculated as

$$\begin{aligned} E[S(L)] &= \frac{1}{L} \cdot \sum_{l=1}^{L} P_l^c \cdot S_l \\ &= (P_n^c)^2 \cdot S_l. \end{aligned} \tag{21}$$

We illustrate the resulting expected header compression savings in Figure 17 for a fixed frame length of $F = 10$, which corresponds to the highest header compression savings for the referential header compression scheme in case of a single reliable link. We observe that in case of reliable links along the path, the expected header compression savings only depend on the probability (or fraction) of nodes that are assumed to have compression enabled. In addition, we observe that only a minor fraction of nodes performs header compression, the expected path header compression savings may not justify the additional efforts, i.e., computational overhead and energy consumption, for nodes that would be able and willing to perform header compression. With respect to these findings, it is necessary to investigate the incurred overhead, especially with respect to power consumptions, to derive a threshold. We now extend our evaluation to unreliable links.

5.2. Unreliable Links

To extend the results obtained for unreliable single links to several links L constituting a path, we need to consider the probabilities of headers being transmitted uncompressed over links that do not support header compression along the path of from the sender to the receiver. We incorporate the probability of losses on these links without header compression into our

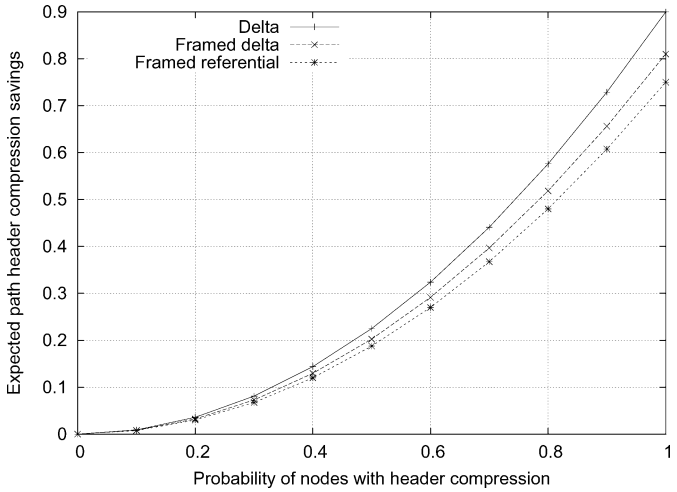

Figure 17. Expected path header compression savings for reliable links with frame lengths $F = 10$ for delta, framed delta, and referential coding header compression schemes.

calculations and provide an end-to-end view on the achievable expected savings for the different header compression schemes.

1) Delta Coding: For delta coding, we first calculate the number of headers that we expect to be sent consecutively from the sender to the receiver until the first decompression error occurs at the receiver's decompressor. Let the random variable ξ denote the number of compressed headers that can be sent over L links until a header suffers from a transmission error. To this end, we use the random variable ξ similar to the random variable γ introduced in Section 3 for single links. Assuming that on each link either header compression is enabled or disabled, we calculate the expectation as

$$E[\xi] = \left\lfloor \frac{1}{1 - \left[P_l^u \cdot P^g(U) + P_l^c \cdot P^g(C)\right]^L} \right\rfloor. \tag{22}$$

With the thus determined expected number of headers that can be sent until an error occurs, we derive the expected path header compression savings as

$$\begin{aligned} E[S_\Delta(L)] &= P_l^c \cdot \left\{ 1 - \frac{U + \sum_{i=2}^{E[\xi]-1} C_i + U + \sum_{i=E[\xi]+1}^{E[\xi]+d} U}{(E[\xi] + d) \cdot U} \right\} \\ &= P_l^c \cdot \left\{ 1 - \frac{(E[\xi] - 2) \cdot C + (d + 2) \cdot U}{(E[\xi] + d) \cdot U} \right\}. \end{aligned} \tag{23}$$

We illustrate the expected path header compression savings for fixed packet drops $d = 4$ and different path lengths in Figures 18 and 19 for bit error rates of 10^{-4} and 10^{-3}, respectively. We observe that the header compression savings for a lower bit error rate are all positive. For higher bit error rates the savings become negative as the number of links increases. Only for a high probability P_n^c of nodes having header compression enabled, the savings become positive for more than one link. In contrast, for a lower bit error rate, the savings are positive even for the largest number of links $L = 15$. In both scenarios, however, the probability of header compression enabled nodes increases the header compression savings similar to the reliable

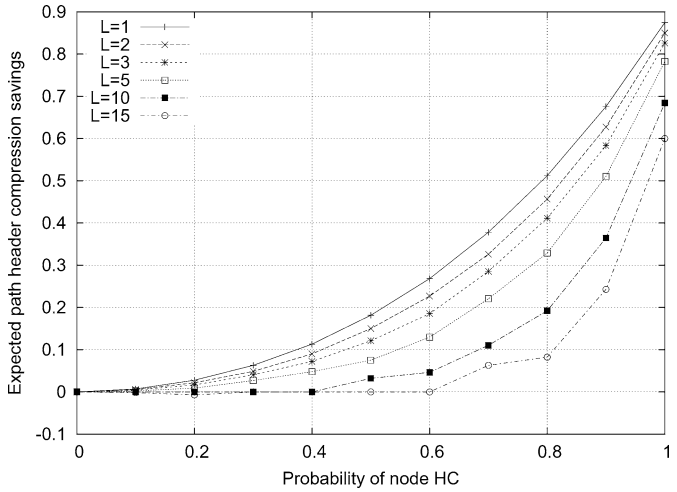

Figure 18. Expected path header compression savings for delta coding header compression scheme with $d = 4$ packet drops and different path lengths L for bit error rate of 10^{-4}.

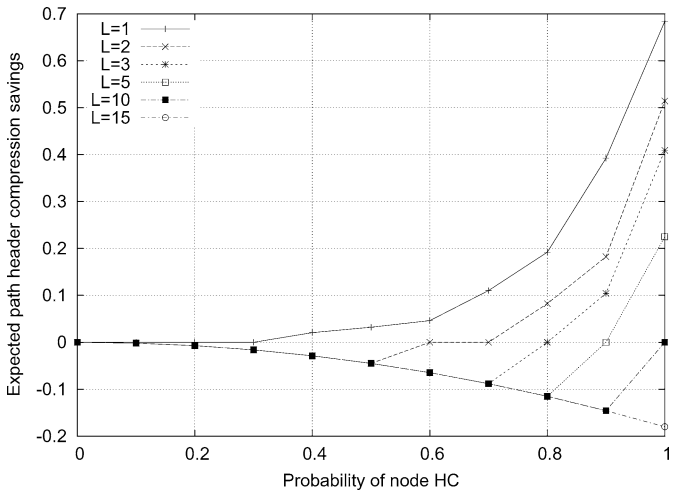

Figure 19. Expected path header compression savings for delta coding header compression scheme with $d = 4$ packet drops and different path lengths L for bit error rate of 10^{-3}.

case. For the case of $L = 5$, we observe that even if all nodes have header compression enabled, the savings decline from approximately 0.8 for a bit error rate of 10^{-4} to 0.2 for a bit error rate of 10^{-3}.

We derive the expected path packet drop savings for the delta coding header compression scheme as

$$E[D_{\Delta}(L)] = P_l^c \cdot \left\{ 1 - \frac{d+1}{\left(E[\xi] + d\right) \cdot \left(1 - [P^g(U)]^L\right)} \right\} \tag{24}$$

We illustrate the expected path packet drop savings for packet drops $d = 4$ and different path lengths in Figures 20 and 21 for bit error rates of 10^{-4} and 10^{-3}, respectively. We observe that the expected path packet drop savings remain negative for both different bit error rates and for nearly all different node header compression probabilities. We additionally observe that for

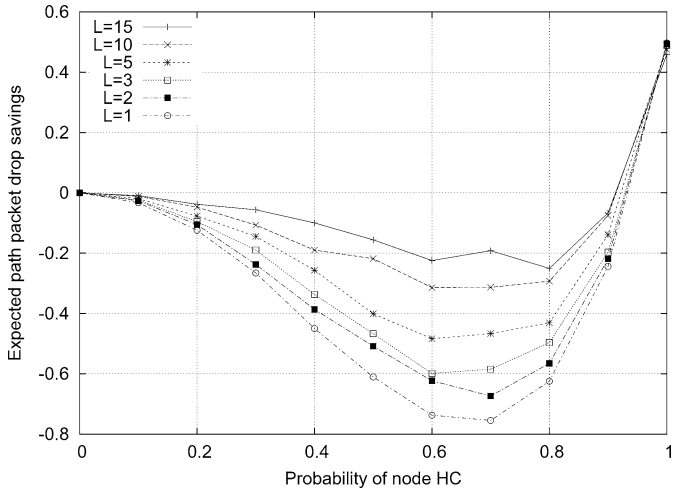

Figure 20. Expected path packet drop savings for delta coding header compression scheme with $d = 4$ packet drops and different path lengths L for bit error rate of 10^{-4}.

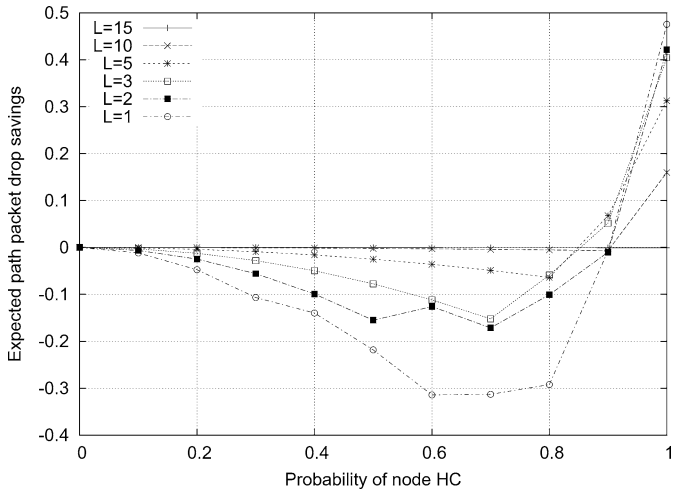

Figure 21. Expected path packet drop savings for delta coding header compression scheme with $d = 4$ packet drops and different path lengths L for bit error rate of 10^{-3}.

both illustrated bit error rates, it seems most undesirable to have only a fraction of nodes to perform header compression, as the minima are attained for one half to three quarters of the nodes having header compression enabled.

2) Framed Delta Coding: For framed delta coding, we had the probability of the hth header within a frame being erroneous given as in Equation (12) for $h \geq 2$. For links on which no header compression is performed, the uncompressed headers are sent instead of the compressed headers. The probability of the uncompressed header C_h to traverse such link without an error is given as

$$P^g(C_h, U) = [P^g(U)]^h, \tag{25}$$

noting that the header h requires all previous headers in the frame to be successfully transmitted over this link. Assuming that we need to traverse all L links connecting the N nodes, the

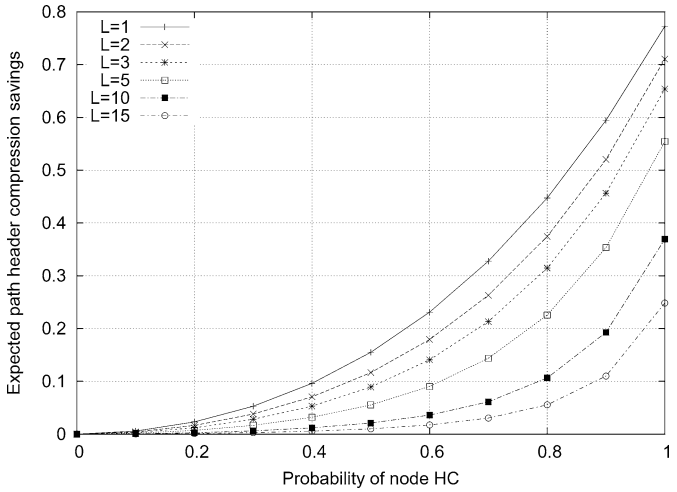

Figure 22. Expected path header compression savings for framed delta coding header compression scheme with frame length $F = 15$ headers and different path lengths L for bit error rate of 10^{-4}.

probability of the hth header to reach the destination can be denoted as

$$P^g_{F\Delta}(h \geq 2, L) = [P^u_l \cdot P^g(C_h, U) + P^C_l \cdot P^g_{F\Delta}(h)]^L. \tag{26}$$

Using the probability of header h being transmitted correctly over a header compression enabled link

$$P^g_{F\Delta}(h \geq 2) = P^g(U) \cdot [P^g(C)]^{h-1}, \tag{27}$$

we finally calculate the probability of header h being delivered correctly to the destination as

$$P^g_{F\Delta}(h \geq 2, L) = \left\{P^u_l \cdot [P^g(U)]^h + P^c_l \cdot P^g(U) \cdot [P^g(C)]^{h-1}\right\}^L. \tag{28}$$

The expected header savings for the whole path are subsequently calculated as

$$\begin{aligned} E[S_{F\Delta}(L)] &= P^c_l \cdot \frac{1}{F} \cdot \sum_{h=2}^{F} P^g_{F\Delta}(h, L) \cdot \left(1 - \frac{C}{U}\right) \\ &= \frac{P^c_l}{F} \cdot \left(1 - \frac{C}{U}\right) \cdot \sum_{h=2}^{F} P^g_{F\Delta}(h, L). \end{aligned} \tag{29}$$

We illustrate the expected path header compression savings in Figures 22 and 23 as function of the node probability to have header compression enabled and for two bit error rates of 10^{-4} and 10^{-3}, with frame lengths $F = 15$ and $F = 7$, respectively. We observe that the expected path header compression savings decline as the path length L increases. We furthermore observe that the expected path header compression savings as function of the node header compression probability exhibit a characteristics observed for the delta coding header compression scheme. We additinally note that for high bit error rates, the expected path savings for $L = 5$ links forming a path are already below 5%, even in case all nodes perform header compression. In case of lower bit error rates, the expected path savings even for $L = 15$ links still remain above 20% in case all nodes use header compression.

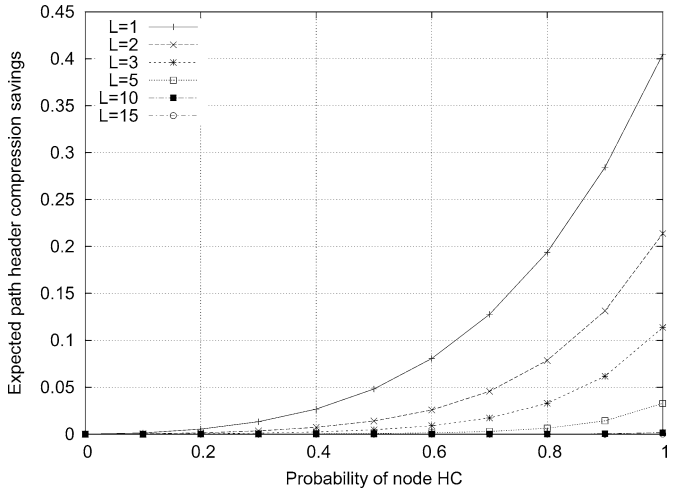

Figure 23. Expected path header compression savings for framed delta coding header compression scheme with frame length $F = 7$ headers and different path lengths L for bit error rate of 10^{-3}.

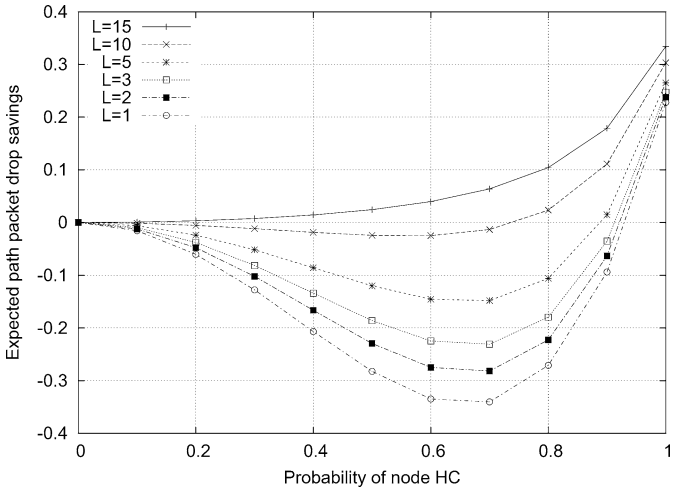

Figure 24. Expected path packet drop savings for framed delta coding header compression scheme with frame length $F = 15$ headers and different path lengths L for bit error rate of 10^{-4}.

For the expected packet drop savings, we derive

$$E[D_{F\Delta}(L)] = P_l^c \cdot \left\{ 1 - \frac{1}{F} \cdot \frac{F.\left(1 - [P^g(U)]^L\right) + \sum_{h=2}^{F} P_{F\Delta}^e(h, L) \cdot (F - h + 1)}{F \cdot P^e(U)} \right\}. \quad (30)$$

with $P_{F\Delta}^e(h \geq 2, L) = 1 - P_{F\Delta}^g(h \geq 2, L)$. We illustrate the expected path packet drop savings in Figures 24and 25 as function of the node probability to have header compression enabled and for two bit error rates of 10^{-4} and 10^{-3}, with frame lengths $F = 15$ and $F = 7$, respectively. We observe that for low bit error rates, we derive a fanout between 40% and 90% of the nodes using header compression, similar to the observations of delta coding only. We observe that for more than a single link, the savings become negative, attain a minimum, and rise close to the value obtained for the single link, whereby the savings increase with the number of links.

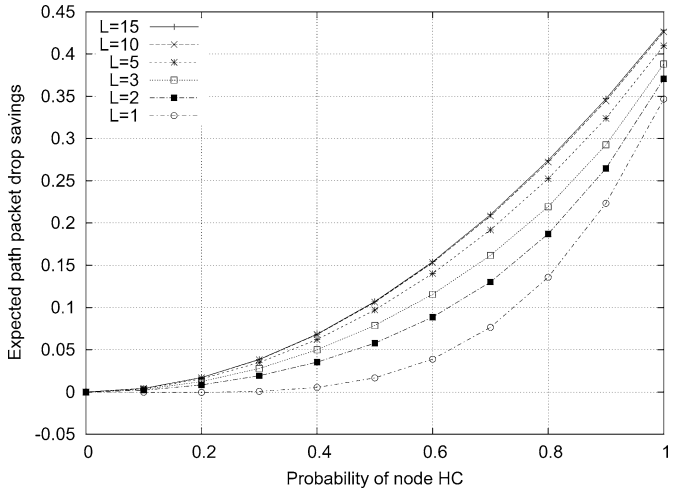

Figure 25. Expected path packet drop savings for framed delta coding header compression scheme with frame length $F = 7$ headers and different path lengths L for bit error rate of 10^{-3}.

We note that in case of higher bit error probabilities, the expected savings remain positive except for a minuscule drop below 0 and are only increasing.

3) Framed Referential Coding: For the framed referential header compression scheme, the probability of an individual compressed header h within a frame being transmitted successfully is expressed as

$$P^g_{FR}(h \geq 2) = P^g(U) \cdot P^g(C_h). \tag{31}$$

Using this probability, we derive the probability for this header to reach the destination node as

$$P^g_{FR}(h \geq 2, L) = \left\{P^u_l \cdot [P^g(U)]^2 + P^c_l \cdot P^g_{FR}(h)\right\}^L \cdot \tag{32}$$

Similar to the previous considerations for the individual links, we derive the expected header compression savings along a path with N nodes as

$$E[S_{FR}(L)] = P^c_l \cdot \frac{1}{F} \cdot \sum_{h=2}^{F} P^g_{FR}(h, L) \cdot \left(1 - \frac{C_h}{U}\right) \cdot \tag{33}$$

We note again that the savings for the initial (uncompressed) header $h = 1$ are 0.

We illustrate the expected path header compression savings in Figures 26 and 27 as function of the node probability to have header compression enabled and for two bit error rates of 10^{-4} and 10^{-3}, both with a frame length of $F = 10$. We observe that the expected path header compression savings for framed referential coding exhibit mostly the same characteristics that were observed for the framed delta coding scheme. We furthermore note that the level of savings is higher throughout the different bit error rates and node header compression probabilities in comparison to the framed delta coding scheme.

For the expected packet drop savings due to the header compression scheme, we first calculate the probability of transmission errors for header h along the path as

$$P^e_{FR}(h \geq 2, L) = 1 - \left\{P^u_l \cdot [P^g(U)]^2 + P^c_l \cdot P^g_{FR}(h)\right\}^L \cdot \tag{34}$$

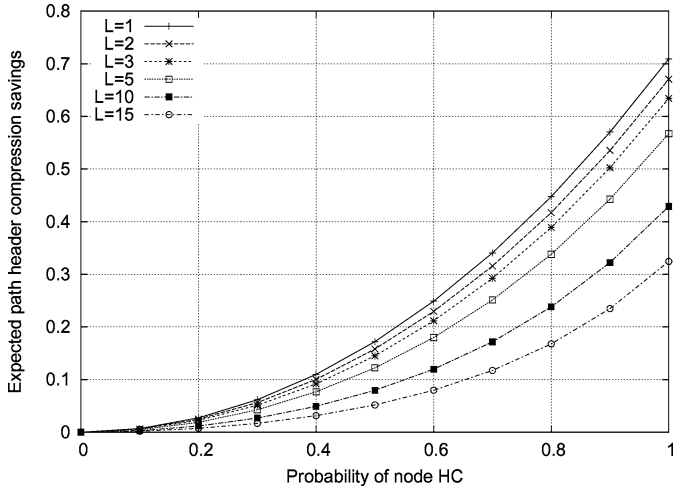

Figure 26. Expected path header compression savings for framed referential coding header compression scheme with frame length $F = 10$ headers and different path lengths L for bit error rate of 10^{-4}.

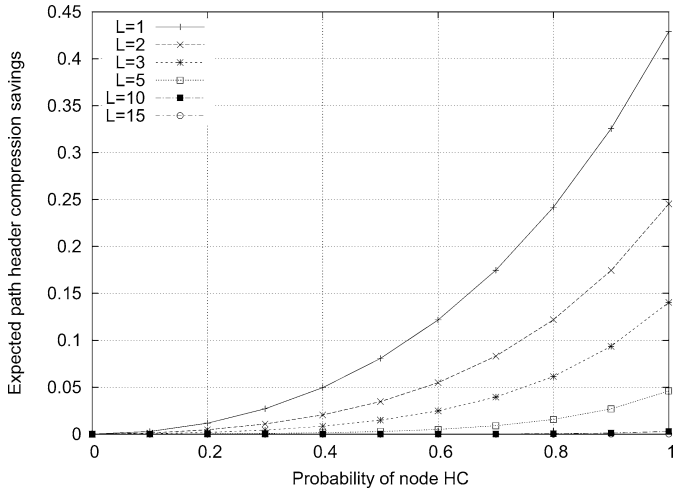

Figure 27. Expected path header compression savings for framed referential coding header compression scheme with frame length $F = 10$ headers and different path lengths L for bit error rate of 10^{-3}.

The expected packet drop savings are calculated only with respect to the fraction of the links having header compression enabled as

$$E[D_{FR}(L)] = P_l^c \left\{ 1 - \frac{1}{F} \cdot \frac{F \cdot \left(1 - [P^g(U)]^L\right) + \sum_{h=2}^{F} P_{FR}^e(h, L)}{F \cdot \left(1 - [P^g(U)]^L\right)} \right\} \tag{35}$$

We illustrate the expected path packet drop savings in Figures 28 and 29 as function of the node probability to have header compression enabled and for two bit error rates of 10^{-4} and 10^{-3}, both with a frame length of $F = 10$ headers. We observe that in stark contrast to the two other header compression schemes, the framed referential coding scheme does not result in a significant fanout for either node header compression probabilities or number of links traversed. This result is similar to the result obtained for the evaluation on individual links. We

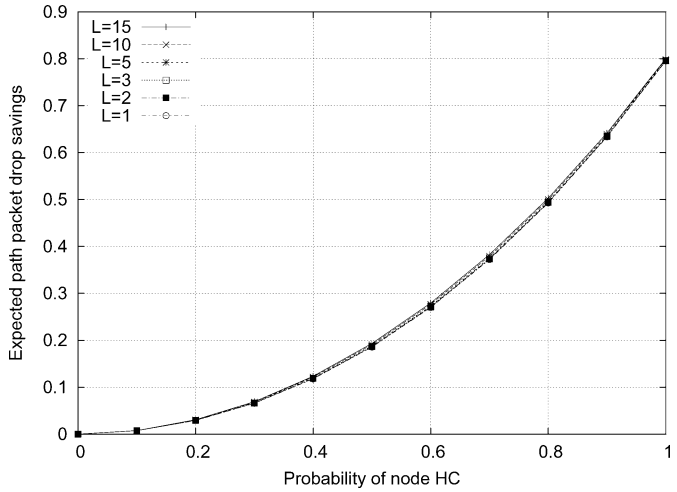

Figure 28. Expected path packet drop savings for framed referential coding header compression scheme with frame length $F = 10$ headers and different path lengths L for bit error rate of 10^{-4}.

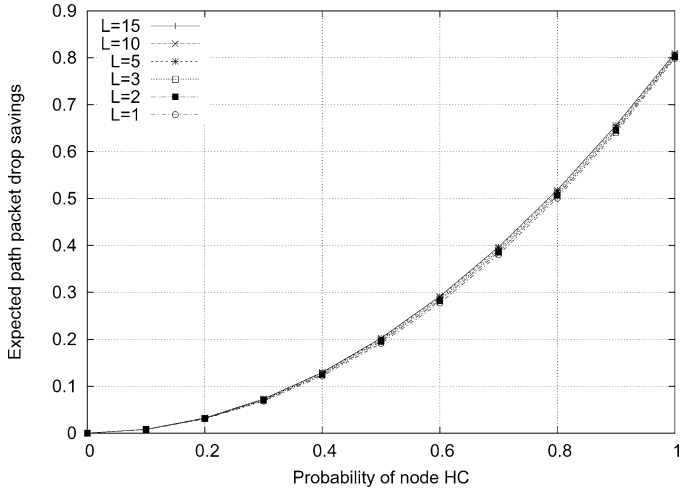

Figure 29. Expected path packet drop savings for framed referential coding header compression scheme with frame length $F = 10$ headers and different path lengths L for bit error rate of 10^{-3}.

additionally note that the level of savings can be approximated by the link's header compression probabilities and the results obtained for an individual link.

5.3. Expected Combined Path Savings

Similar to the evaluation for links, we now look at the combined savings in terms of header compression savings and in terms of savings for dropped packets due to header compression errors. We illustrate the expected combined savings for paths in Figures 30, 31, and 32 for the delta coding, framed delta coding, and the framed referential coding header compression schemes, respectively, for a medium bit error rate of 10^{-4} for which we examined the different schemes in Section 5.2. We observe that for delta coding and framed delta coding schemes, the

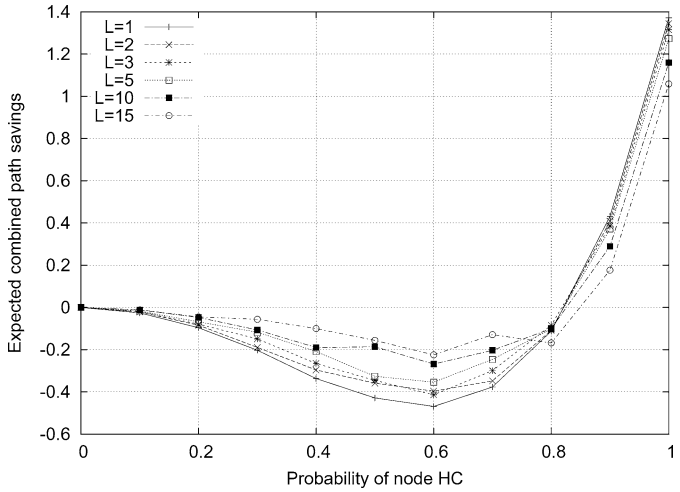

Figure 30. Expected combined path savings for delta coding header compression scheme with packet drops $d = 4$ and bit error rates 10^{-4}.

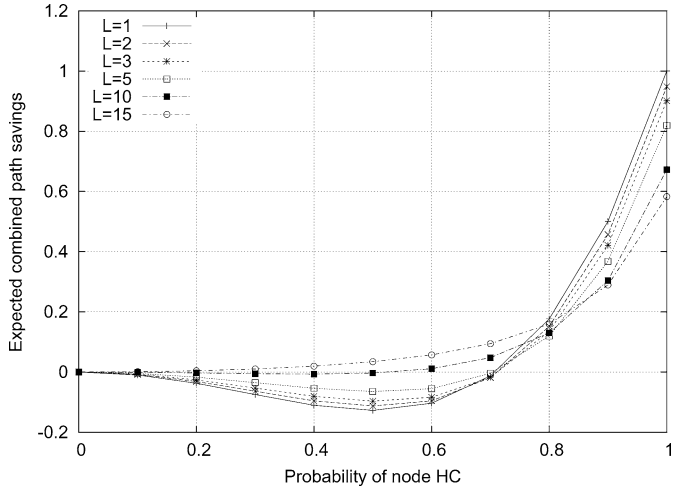

Figure 31. Expected combined path savings for framed delta coding header compression scheme with frame length $F = 7$ and bit error 10^{-4}.

combined savings for a medium node probability of having header compression enabled results in negative expected combined path savings. For the framed referential header compression scheme, the expected savings are throughout positive for all node probabilities and path lengths. The referential approach results in the highest obtainable savings of all the three evaluated schemes and thus seems superior. For the two delta coded schemes, having only a fraction of the nodes perform header compression greatly reduces the gains of compression, as intermediate links would require uncompressed sending, which, due to the interdependence of the headers, in turn leads to significant losses. Only in case of more than 90% of the nodes using header compression for the delta coding scheme and more than 80% for the framed delta coding scheme these schemes result in positive combined savings. In summary, the referential coding results in a more stable coding performance, which, for the changing nature of wireless ad-hoc and multi-hop networks is more suitable to accommodate the network topology and node capability changes.

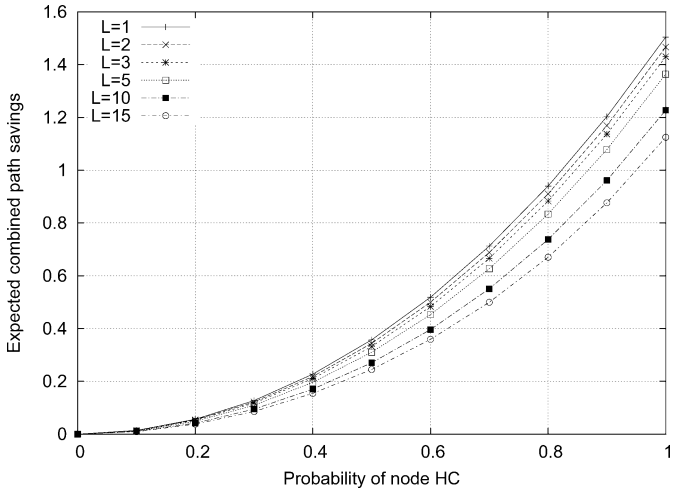

Figure 32. Expected combined path savings for framed referential coding header compression scheme with frame length $F = 10$ and bit error rate 10^{-4}.

6. Conclusion

We introduced and evaluated three different approaches to header compression in general for reliable and unreliable links and paths. For reliable links, the header compression savings for (*i*) delta coding, (*ii*) framed delta coding, and (*iii*) framed referential coding can be calculated from the size of the compressed and uncompressed headers. For paths, the compression savings can be adjusted taking into account the node probability of having header compression enabled.

For highly unreliable links, such as to be found in wireless networks, the header compression savings greatly depend on the number of packets needed for re-synchronization of decompressor and compressor for delta coding. For frame-based coding the number of headers constituting a frame is the most important parameter for expected compression savings. For framed referential coding, we find that the highest savings are obtained independent of the bit error rate for frame lengths of $F = 10$ headers per frame. For the savings in packet drops due to applied header compression, we find that the framed delta scheme both exhibits a peak in the savings, whereas the framed referential scheme results in increasing savings with the frame length, as the probability of loosing only the first header declines. We furthermore find that the obtainable savings for delta coding strongly depend on the number of packets lost until re-synchronization and that in case of more than 18 dropped packets, the expected drop savings diminish in most cases.

For highly unreliable paths, we find that the header compression savings decline if only a low to medium fraction of the nodes supports header compression. Given the higher probability of transmission errors on links without header compression, especially delta coding introduces a high level of losses due to dependencies. For the packet drop savings, we observe that the schemes employing delta coding exhibit a maximum overhead (negative savings) if a major fraction of the nodes has header compression enabled. Furthermore, we observe that in case of more than 80–90% of the nodes perform header compression, the savings become positive again. For framed referential header compression schemes, we do not observe that behavior for drop savings and derive only positive savings increasing with the nodes that perform header compression.

We define a combined savings metric that combines the header compression savings and the savings in packet drops due to header compression schemes. We find that in most cases, the framed referential coding scheme results in the highest combined savings, if it takes more than just several packets for the delta coded scheme to re-synchronize compressor and decompressor.

Overall, these results give a first understanding in the suitability of header compression schemes in the multi-hop domains of wireless networking. Further research directions include the simulation of different traffic patterns and contents with the resulting savings over paths, as well as determining possible cooperation issues that arise from the decline in savings with the probability or number of nodes having header compression enabled.

Acknowledgements

We would like to thank Prof. V. R. Syrotiuk from the Computer Science and Engineering Department, Arizona State University. This work was supported in part by the National Science Foundation under Grant No. ANI-0136774.

References

1. V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links, Request for Comments 1144," February 1990.
2. S.J. Perkins and M. W. Mutka, "Dependency Removal for Transport Protocol Header Compression over Noisy Channels," in *Proc. of IEEE International Conference on Communications (ICC)*, vol. 2, Montreal, Canada, June 1997, pp. 1025–1029.
3. M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression, Request for Comments 2507" February 1999.
4. S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, Request for Comments 2508," Feb. 1999.
5. C. Bormann, C. Burmeister, M. Degermark, M. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, "RObust Header Compression: ROHC: Framework and four profiles: RTP, UDP, ESP, and uncompressed," Request for Comments 3095, Tech. Rep., July 2001.
6. F. Fitzek, S. Hendrata, P. Seeling, and M. Reisslein, *Wireless Internet*, ser. Electrical Engineering & Applied Signal Processing. CRC Press, 2004, ch. Header Compresion Schemes for Wireless Internet Access.
7. A. Calversas, M. Arnau, and J. Paradells, "An Imprvement of TCP/IP Header Compression Algorithm for Wirelss Links," in *Proc. of Third World Multiconference on Systemics, Cybernetics and Informatics (SCI'99) and the Fifth International Conference on Information Systems Analysi and Synthesis (ISAS'99)*, vol. 4, Orlando, FL, July/August 1999, pp. 39–46.
8. ——, "A controlled Overhead for TCP/IP Header Compression Algorithm over Wireless Links," in *Proc. of the 11th International Confernce on Wireless Communications (Wireless'99)*, Calgary, Canda, 1999.
9. M. Degeermark, H. Hannu, L. Jonsson, and K. Svanbro, "Evaluation of CRTP performance over cellular radio links," *IEEE Personal Communications*, vol. 7, no. 4, pp. 20–25, 2000.
10. R. Sridharan, R. Sridhar, and S. Mishra, "A robust header compression technique for wirless ad hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 23–24, 2003.
11. C. Bormann, "Rohe over 802," Internet Draft (work in progress), Feb. 2005. [Online]. Available: http://www.ietf.org/internet-drafts/draft-bormann-rohc-over-802-01.txt
12. F. Fitzek, S. Rein, P. Seeling, and M. Reisslein, "Robust header compresion (rohc) performance for multimedia transmission over 3g/4g wireless networks," *Wireless Personal Communications*, vol. 32, no. 1, pp. 23–41, Jan. 2005.

13. H. Wang, J. Li, and P. Hong, "Performance analysis of rohc u-mode in wireless links," *IEE Proceedings of Communications*, vol. 151, no. 6, pp. 549–551, Dec. 2004.
14. B. Wang, H. Schwefel, K. Chua, R. Kutka, and C. Schmidt, "On implementation and improvement of robust header compression in umts," in *Proc. of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Lisbon, Portugal, Sept. 2002, pp. 1151–1155.
15. M. Rossi, A. Philippini, and M. Zorzi, "Link error characteristics of dedicated (dch) and common (cch) umts channels," Universita di Ferrara, FUTURE Group, Ferrara, Italy, Tech. Rep., July 2003.
16. A. Willig, "A new class of packet- and bit-level models for wireless channels," in *Proc. of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, vol. 5, Lisbon, Portugal, Sept. 2002, pp. 2434–2440.

Patrick Seeling is a Faculty Research Associate in the Department of Electrical Engineering at Arizona State University (ASU), Tempe. He received the Dipl.-Ing. degree in Industrial Engineering and Management (specializing in electrical engineering) from the Technical University of Berlin (TUB), Germany, in 2002. He received his Ph.D. in electrical engineering from Arizona State University, Arizona, in 2005. His research interests are in the area of multimedia communications in wired and wireless networks and engineering education. He is a member of the IEEE and the ACM.

Martin Reisslein is an Associate Professor in the Department of Electrical Engineering at Arizona State University (ASU), Tempe. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994–1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin and lecturer at the Technical University

Berlin. From October 2000 through August 2005 he was an Assistant Professor at ASU. He is editor-in-chief of the *IEEE Communications Surveys and Tutorials* and has served on the Technical Program Committees of *IEEE Infocom, IEEE Globecom*, and the *IEEE International Symposium on Computer and Communications*. He has organized sessions at the *IEEE Computer Communications Workshop (CCW)*. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG-4 and H.263 encoded video, at http://trace.eas.asu.edu. He is co-recipient of the Best Paper Award of the *SPIE Photonics East 2000 – Terabit Optical Networking* conference. His research interests are in the areas of Internet Quality of Service, video traffic characterization, wireless networking, optical networking, and engineering education.

Tatiana K. Madsen has received her M.Sc. and Ph.D. degrees in Mathematics from Moscow State University, Russia in 1997 and 2000, respectively. In 2001 she joined Dept. of Communication Technology, Aalborg University, Denmark where she is currently an Assistant Professor. Her research interests lie within the areas of wireless networking with the focus on IP header compression techniques and mathematical modeling of wireless protocols behavior.

Frank Fitzek is an Associate Professor in the Department of Communication Technology, Unversity of Aalborg, Denmark heading the Future Vision gorup. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology – Rheinish-Westflische Technische Hochschule (RWTH) – Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical Univeristy Berlin, Germany in 2002 for quality of service support in wireless CDMA networks. As a visiting student at the Arizona State University he conducted research in the field of video services over wireless networks.

He co-founded the start-up company acticom GmbH in Berlin in 1999. In 2002 he was Adjunct Professor at the University of Ferrara, Italy giving lectures on wireless communications and conducting research on multi-hop networks. In 2005 he won the YRP award for the work on MIMO MDC. His current research interests are in the areas of 4G wireless communication networks and cooperative networking. Dr. Fitzek serves on the Editorial Board of the IEEE Communications Surveys and Tutorials.