TCP/IP Header Compression: Proposal and Performance Investigation on a WCDMA Air Interface

M. Rossi, A. Giovanardi, M. Zorzi, G. Mazzini
DIF, University of Ferrara
E-mail: {mrossi, agiovanardi, mzorzi, gmazzini}@ing.unife.it

Abstract— **Two TCP/IP header compression schemes for wireless networks are proposed and investigated on a WCDMA air interface. These novel schemes are compared with the classic ones by means of a TCP/IP New Reno simulator taking as input WCDMA channel traces. These channel traces are derived by using a simulator built according to the UMTS standard requirements. RLC level operates in transparent mode, while overhead is added to simulate the presence of a MAC level. Some new performance metrics are introduced to evaluate the HC schemes performance.**

I. Introduction

The wide diffusion of Internet and wireless networks leads to the integration of these two technologies to provide wide-area wireless Internet access.

Regarding Internet, TCP/IP [1] is the most important protocol stack integrating a wide range of different physical networks. As far as wireless networks are concerned, 3G cellular systems, like UMTS, based on a Wideband CDMA (WCDMA) air interface [2] [3], are an interesting solution to integrate voice, data and video. These trends suggest to investigate the TCP/IP performance on a WCDMA wireless channel.

Let us first note that, compared with fixed and wired networks, wireless networks have higher bit-error rate (BER) and offer less available bandwidth. Furthermore, host mobility also results in packet loss or delay during handoff procedures. Hence, the performance of TCP/IP over such wireless networks without any modification would suffer significant degradation of throughput and large interactive delay.

In order to efficiently use the available resources, one method is to reduce the size of the protocol headers through Header Compression (HC) [4] [5] [6] [7]. HC is advantageous both in interactive, i.e., telnet sessions, and in continuous applications, i.e., FTP sessions. In the former case the packet payload is usually smaller with respect to the latter, but in both cases the header overhead leads to a degradation in terms of throughput. Furthermore, HC reduces the packet loss in force of the reduced packet size.

In this paper, we investigate the performance of classic TCP/IP (IPv4) HC schemes on a WCDMA wireless channel and propose two new HC techniques [8] to reduce the header synchronization loss [4] [5] [6] [7]. The performance has been evaluated by means of a TCP/IP New Reno simulator [9], taking as input WCDMA channel traces generated according to UMTS standard requirements [3]. Radio Link Control (RLC) level [10] operates in transparent mode, while overhead is added to simulate the presence of a Medium Access Control (MAC) level.

The paper is structured as follows: in Section II the existing and new proposed HC schemes are discussed; in Section III the WCDMA channel and TCP/IP HC schemes simulators are described; in Section IV simulation results comparing the HC techniques, by varying some significant WCDMA channel parameters, are reported. Finally, in Section V some conclusions are given.

II. Header Compression Schemes

The existing TCP/IP HC algorithms considered are: Van Jacobson (VJHC) [4], Degermark (DEHC) [5] [6] and Perkins (PEHC) [7]. Let us first give a brief review of these schemes to understand the improvements introduced by the two proposed algorithms (called *Adaptive Base Proposal with Static Compression Start*, **ABP-SCS**, and *Adaptive Base Proposal with Dynamic Compression Start*, **ABP-DCS**).

A. Classical HC Algorithms

Regarding the VJHC scheme [4], let us consider the sender performing header compression (called compressor C) and the receiver performing header decompression (called decompressor D). The compression is achieved by sending the differences between the current n-th header and the previous $(n-1)$-st one, and updating the reference value, i.e., the uncompressed version of the last header transmitted, for each packet sent. The header differences are called *header delta fields* while the reference value is named *header base*.

The decompression is achieved by summing to the last *header delta fields* received the current *header base* and by storing the result. Since VJHC uses a differential encoding scheme, the loss of a packet always leads to a loss of synchronization i.e., the event in which the last transmitted *header delta fields* are not correctly received by D and then the decompression fails [5] [7] [11]. To recover from this situation, C sends uncompressed packets at each TCP retransmission, that are used by D to resynchronize the *header base*. VJHC works well in wired links, while in a wireless environment the high packet loss probability leads to frequent HC synchronization loss.

DEHC [5] improves VJHC by enhancing the synchronization robustness. When synchronism is lost, D assumes that this is due to the loss of a single packet (e.g., $n-1$) and that its *header delta fields* are equal to those of the next packet (i.e., n). The correct *header base* is then computed by adding twice the *header delta fields* of packet n to the

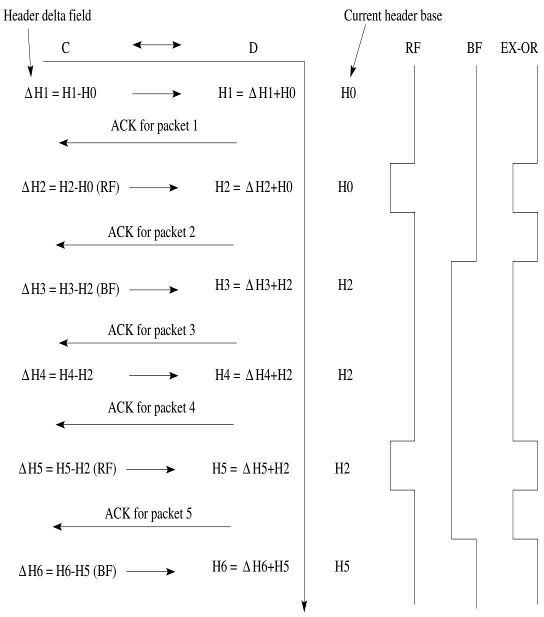

Fig. 1. Proposed *header base* update mechanism.

last assumed reference value, while ignoring the contents of packet $n-1$. If the assumption is correct, the synchronism is preserved, otherwise a resynchronization request is sent to C, by using the TCP ACKs flowing from D to C. Note that the "twice" mechanism avoids loss of synchronization only if exactly one packet is lost or corrupted and if its *delta fields* are equal to those of the following packet received. For these reasons DEHC has poor performance in a bursty error environment or when the *delta fields* are not stationary.

A second algorithm which aims at reducing the synchronization loss probability is PEHC [7], where the *header base* is updated less frequently, i.e., every $K > 1$ packets, with respect to VJHC where $K = 1$. Furthermore, in PEHC each update is performed by sending the uncompressed header. Thus, in PEHC the packet loss does not necessarily leads to a synchronization loss. K is set as large as possible, while respecting the constraint on the maximum size of the *header delta fields* [4].

B. Proposed HC Algorithms

To improve the performance, we propose to neglect the synchronization loss by continuing to use the current *header base* until the new one is successfully updated at the receiver.

In particular, when C decides to update the *header base*, it sends it to D having set a suitable flag, called Request Flag (RF). This flag notifies D of the *header base* update. If D correctly receives the new *header base*, it sends to C the TCP ACK including the packet sequence number. When C receives this ACK, it starts using the new *header base* for compression. Thus, the next packet transmitted by C is encoded by using the new *header base* and this change is notified to D by means of a Bistable Flag (BF) that implements a sliding window mechanism of length two on the *header base* change. By observing that RF and BF transitions occur at different times, it is possible to encode them into a single flag by using the exclusive-OR operator. This resulting flag is sent from C to D by means of the unused flags currently available, so that no additional overhead in the TCP signaling is introduced. The *header base* update is performed as frequently as possible to minimize the *header delta fields* resource occupancy. Since the *header base* update involves two adjacent ACKs, this is also assumed as the rate of the *header base* change.

The proposed *header base* update mechanism is depicted in Figure 1, where also the RF and BF flags transitions are reported. As shown, when C proposes to D a new *header base*, it starts using this base only when the corresponding ACK has been correctly received from D, otherwise C continues using the current one. We remark that in our proposal K, defined in the preceding subsection, is implicitly set as a function of the channel conditions. Furthermore, the joint mechanism of RF, BF and ACKs permits to avoid the synchronization loss almost always.

If a transmission error occurs, i.e., C does not receive the ACK for a given packet, that packet is retransmitted without compression as in [4]. C restarts the HC mechanism with the first new packet after the last retransmission, without waiting for the corresponding ACK.

The algorithm that implements the new *header base* update mechanism described above is called *Adaptive Base Proposal with Static Compression Start* (**ABP-SCS**).

Another possible scheme continues to transmit uncompressed packets until C receives the ACK relative to the last retransmitted packet, to avoid the performance degradation due to HC in error prone channel. By doing so, some new packets may also be transmitted with uncompressed headers, but loss of synchronization due to corrupted packets after all retransmissions is always avoided. This second strategy proposed is called *Adaptive Base Proposal with Dynamic Compression Start* (**ABP-DCS**).

Let us note that in our proposal, as depicted in Figure 1, each *header base* is maintained for at least 3 steps. Furthermore, the update confirm mechanism, based on double ACKs, makes our update method more efficient if the TCP receiver sends non delayed ACKs, i.e., one ACK is sent for each packet received. In fact, if delayed ACKs are adopted or if some ACK is lost, our update algorithm maintains the fixed base for more than 3 steps by decreasing the compression gain. However, this is balanced by the avoidance of synchronization losses, which would imply retransmissions and consequently lower throughput and higher transmission delay.

III. General Simulator Description

In order to test and compare the classic and proposed HC schemes on a WCDMA air interface, we have developed a UMTS simulator which generates WCDMA *channel traces* and a TCP/IP HC algorithm simulator, taking these traces as input.

The UMTS simulator is based on two steps. The first step simulates multi-user communications in a multi-cell system, accounting for user placements, signal propagation, power control, intra and inter-cell interference, Doppler frequency (which in our simulation accounts for user movements) and other factors. In this first step Signal to Interference Ratio (SIR) traces are computed for each user. The second step is based on a post-processing

PARAMETER	VALUE
n_bases	25
n_users	90,100,120
Cell Side	200 m
β	3.5
A	-30 dB
Max. TX Power	-16 dBW
Power Range	80 dB
σ (shadowing)	4 dB
Doppler Frequency	2, 6, 20, 40 Hz
Time Unit	0.666 ms
number of Oscillators (Jakes)	8
n_rays (Selective Channel)	5
TTI	1,2,4,8 frames
Chip Rate	3.84 Mchip/s
Data Rate	240 kbit/s
SF (Spreading Factor)	16
SIR target	6 dB
δ (Power Control Step)	0.5 dB
Noise	-132 dBW

TABLE I
SIMULATION PARAMETERS

of the SIR traces accounting for channel coding and interleaving. From this we obtain sequences of block error probabilities, one for each user, called *channel traces*.

These *channel traces* are used to randomly generate sequences of block error, BESs, one for each user. Finally, BESs, are fed to the TCP/IP HC algorithm simulator. This simulator implements some known HC schemes (VJHC, DEHC and PEHC) and the new HC schemes (ABP-SCS and ABP-DCS), and returns the average throughput, S, for each user as a function of the average block error probability for that user, P_e. The throughput is defined as the fraction of the channel transmission rate (considered at the IP level) which provides correct information bits at the receiver (not counting erroneous transmissions, spurious transmissions, idle time and overhead). The final throughput, S, of each user is averaged on the entire simulation. The HC simulator also returns the average *header base* update rate, f_H, defined as the inverse of the average number of packets between two successive *header base* changes. This new performance figure is an index of the header compression gain; in fact, high f_H implies lower differences on the *header delta fields* and so higher degree of compression. Another output of the simulator is the average compressed header size, H_s, expressed in bytes, by not considering the effect of window, ACK and urgent pointer fields.

A. UMTS Simulator

Regarding the WCDMA channel simulator, we have considered a hexagonal cell layout with 25 cells, wrapped onto itself to avoid border effects. The cell side is 200m.

The signal propagation has been simulated by considering noise, path loss, log-normal slow fading and Rayleigh fast fading. The noise power is -132dBW. Path loss relates the average received power to the distance r according to the path loss law $P_r(r) = Ar^{-\beta}$, where $A = -30$dB and $\beta = 3.5$. Shadowing is modeled by a multiplicative log-normal random variable with spread $\sigma = 4$dB. Path loss and shadowing are kept constant throughout each simulation. No explicit user mobility is considered, but implicit mobility is taken into account by specifying the Doppler frequency, which characterizes the variation speed of the Rayleigh fading process. Fast fading has been simulated according to Jakes simulator (with 8 oscillator), and to the Doppler frequency selected; furthermore, a frequency-selective fading model with five rays has been used. The presence of a RAKE receiver at the base station is assumed. The Doppler frequency is fixed at $f_d = 2, 6, 20, 40$Hz, to simulate different user mobility. The spreading factor, SF, has been fixed to 16. The raw channel transmission bit rate before spreading is 240Kbit/s (Data Rate); after spreading its value is simply $240 \cdot SF = 3.84$Mchip/s (Chip Rate).

The number of users in the system is variable from 90 to 120, to consider medium and highly loaded situations. The user locations and the base station assignments are set at the start of the simulation, uniformly distributed random way in the first case and by selecting the best radio path in the second one.

A power control (PC) mechanism has been implemented, characterized by a SIR target $\mathrm{SIR}_{ta} = th + \delta$dB, where th is the threshold used to establish if the transmitted power must be increased or decreased, and δ is the power control step. We have considered $\mathrm{SIR}_{ta} = 6$dB, $\delta = 0.5$dB, maximum transmitted power equal to -16dBW and transmitted power range equal to 80dB.

The system time has been discretized and the slot duration matches the time unit used in the PC algorithm, where the time unit is 0.666ms (the block error probabilities are relative to a radio frame, composed by 15 time units, during $15 \cdot 0.666 = 10$ms [3]).

The first step of the simulator gives the SIR traces for each user. In the second step a convolutional code with rate 1/2 and constraint length 8 with Viterbi decoding and interleaving have been considered. The number of radio frames over which interleaving is performed has been fixed to TTI$= 1, 2, 4, 8$. All simulation parameters are reported in Table I.

B. HC Simulator

In this simulator, TCP/IP New Reno [9] version 4 is considered and all HC algorithms described above have been implemented. The data transmission is unidirectional, i.e., packets flow from C to D whereas in the reverse direction only ACKs flow. Receiver generates non delayed ACKs.

The direct link packet generation is assumed continuous, as in a long FTP session, the bit rate at the IP level output is 103.5Kbit/s, the full TCP/IP header is 40 bytes, the MTU size is 128 or 512 bytes and a single path delay of 16ms is considered between C and D.

In VJHC the compressed header size is fixed at 4 bytes. In fact, as observed in [4], in VJHC the compressed header size spans typically from 3 to 5 bytes. Since DEHC shows the same header compression as VJHC, we consider the same header size. On the other hand, in all other HC schemes, showing larger compressed header sizes, the header size has been derived from simulations.

For the sake of comparison, in addition to the HC schemes previously described, we have defined and simulated a special HC algorithm, labeled as IDEAL. In this scheme, VJHC is considered, under the assumption that packet losses due to channel errors affect TCP operation but never lead to loss of synchronization. Note that VJHC is the HC scheme which provides the best compression performance, so, our assumption for the ideal case may

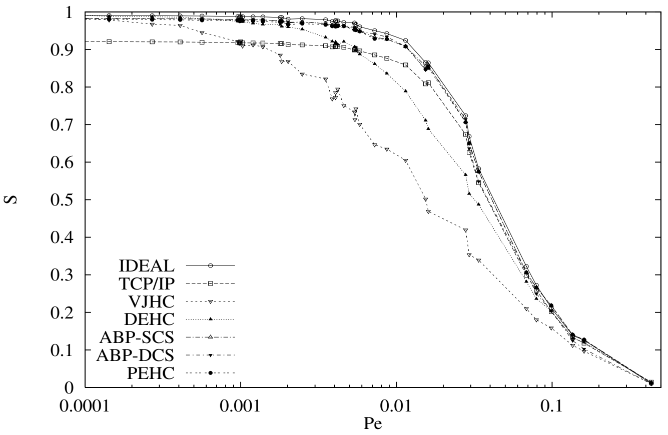

Fig. 2. Throughput S as a function of P_e, for all investigated HC schemes, with 90 users, Doppler frequency f_d = 6Hz, TTI=1 and MTU 512 bytes.

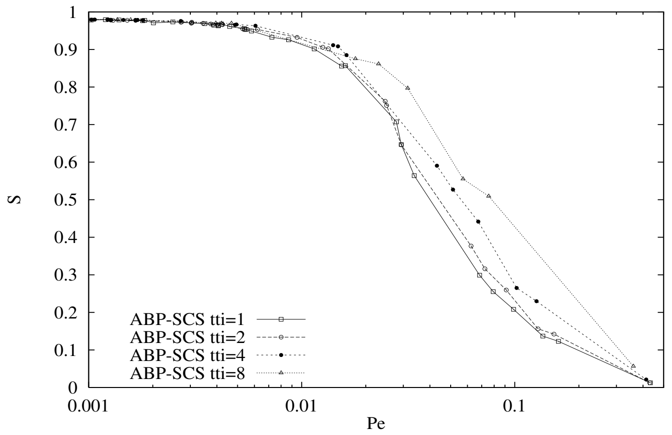

Fig. 4. Throughput S as a function of P_e, for ABP-SCS, with 90 users, Doppler frequency 6Hz, TTI=1,2,4,8 and MTU 512 bytes.

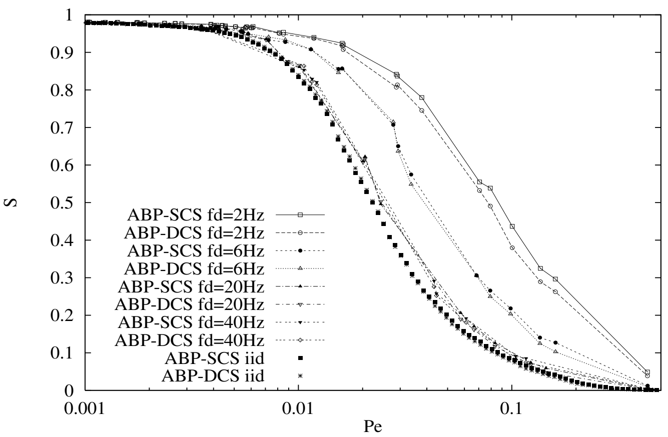

Fig. 3. Throughput S as a function of P_e, for ABP-SCS and ABP-DCS, with 90 users, Doppler frequency $2, 6, 20, 40$Hz TTI=1 and MTU 512 bytes.

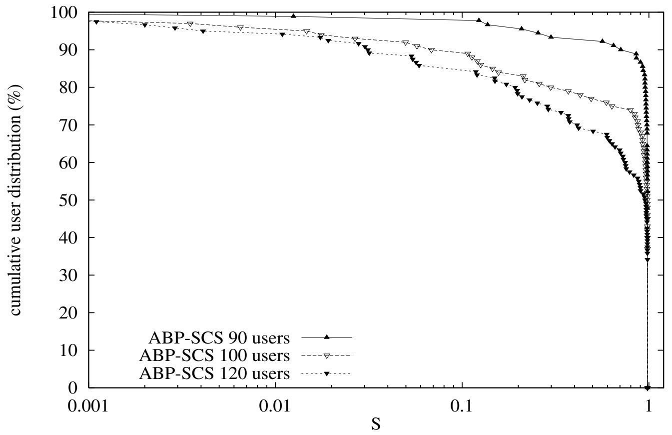

Fig. 5. Cumulative number of users with throughput $\geq S$ vs S (%) for ABP-SCS, with 90,100,120 users, Doppler frequency 6Hz, TTI=1 and MTU 512 bytes.

be expected to provide an optimistic bound for all other schemes, as discussed in [5].

In the simulator we have not considered RLC and MAC levels, supposing that they operate in transparent mode [10]. The bit rate used in the TCP/IP HC algorithm simulator, i.e., the bit rate at the output of the IP level, fixed to 103.4 Kbit/s, has been derived by considering that the raw channel transmission bit rate is 240Kbit/s, the convolutional code has rate 1/2, and some overhead due to a possible control channel and to RLC/MAC has been introduced.

IV. Numerical Results

In this Section some numerical results to test the HC performance on WCDMA channel are reported.

In Figure 2 the throughput S as a function of the average block error probability per user P_e is reported for all investigated HC schemes, with 90 users, Doppler frequency f_d = 6Hz, TTI=1 and MTU 512 bytes. Note that PEHC, ABP-SCS and ABP-DCS show performance close to IDEAL; while TCP/IP without compression and all other HC schemes have lower performance. Note that in some zones ABP-DCS slightly outperforms PEHC and ABP-SCS, because the compression restarts only if the base has been correctly received.

In Figure 3 the throughput sensitivity to the Doppler frequency of the new proposed HC schemes is shown, by varying $f_d = 2, 6, 20, 40$Hz. The case with independent and identically distributed (i.i.d) packet errors is also reported. Note that for increasing f_d the throughput decreases, as expected, and that ABP-SCS and ABP-DCS show quite similar performance. Furthermore, for $f_d > 20$Hz the curves do not show significant changes, because the channel becomes quite similar to that of an i.i.d packet error situation, which is the worst case for TCP/IP, independent of the HC used. For low f_d values, where the error burst length is higher, ABP-SCS slightly outperforms ABP-DCS. This trend has been observed also in [8].

When TTI grows, with values in the range indicated in Table I, the throughput increases, as reported in Figure 4, where the ABP-SCS algorithm has been considered (DCS has a similar trend), with 90 users, f_d = 6Hz and MTU 512 bytes. This behavior is due to the well known inter-

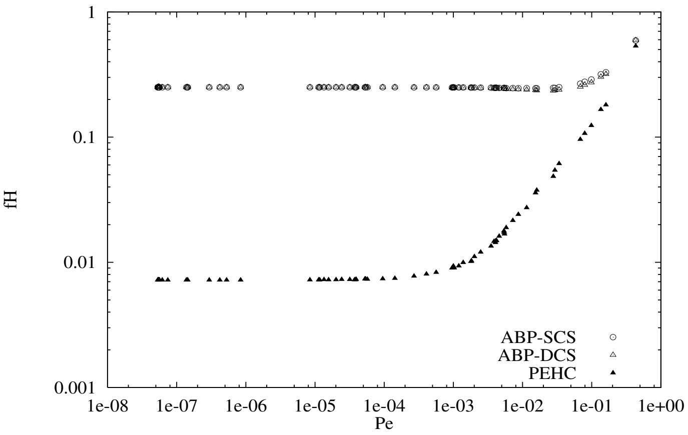

Fig. 6. Header base update rate as a function of P_e for ABP-SCS, ABP-DCS and PEHC, with 90 users, Doppler frequency 6Hz, TTI=1 and MTU 512 bytes.

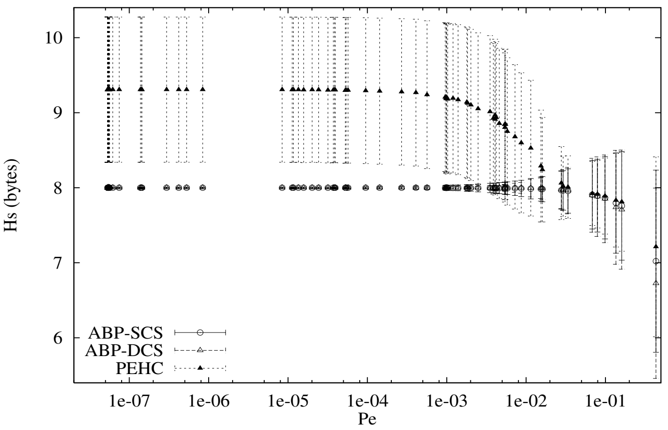

Fig. 7. Compressed header size H_s as a function of P_e for ABP-SCS, ABP-DCS and PEHC, with 90 users, Doppler frequency 6Hz, TTI=1 and MTU 128 bytes.

leaving effects, i.e., operation that mitigates the effect of the channel memory and breaks error bursts, so that error correcting codes as used in the UMTS physical layer are more effective, giving lower P_e.

Regarding the sensitivity to the network load, when the number of users increases the general trend of the throughput showed for 90 users does not change. The only difference is that the number of users with lower performance increases, as depicted in Figure 5, where the fraction of users with throughput $\geq S$ is plotted vs S, for ABP-SCS (DCS has the same trend), with f_d = 6Hz, TTI=1 and MTU 512 bytes.

To show the effectiveness of the new HC schemes, we compare the ABP-SCS, ABP-DCS and PEHC algorithms, which show better performance in terms of throughput, by investigating the degree of header compression. In particular, in Figure 6, the average *header base* update rate is reported as a function of P_e, for f_d = 6Hz, 90 users, TTI=1 and MTU 512 bytes. Note that high f_H implies lower differences on the *header delta fields* and so a higher compression degree. The ABP-SCS and ABP-DCS schemes show higher f_H with respect to PEHC, for about all P_e values. Furthermore, f_H increases with P_e, in fact, by increasing the error probability, the rate of TCP retransmissions grows, and since for each retransmission the uncompressed packet sent is used to update the base, f_H increases. Note that ABP-DCS is slightly outperformed by ABP-SCS as a consequence of its intrinsic periods without HC activities.

Finally, in Figure 7 the average compressed header size, H_s and its standard deviation are reported as a function of P_e for ABP-SCS, ABP-DCS and PEHC, with f_d = 6Hz, 90 users, TTI=1 and MTU 128 bytes. Note the convergence of all methods to the same H_s value for high P_e, and the better performance both in terms of average value and standard deviation of the new proposed schemes with respect to PEHC. VJHC and DEHC are not reported since they achieve a higher degree of compression (with compressed header size spanning from 3 to 5 bytes), even if they are guaranteeing lower throughput as shown in Figure 2.

V. Conclusions

In this paper, we have described two new TCP/IP HC schemes for wireless networks to avoid the synchronization loss problem. The performance of these schemes has been investigated by considering a WCDMA air interface. Performance has been compared with the TCP/IP HC schemes proposed in the literature, by considering various performance figures. Compared to the best existing schemes, the proposed techniques show a slightly higher throughput and achieve a higher degree of compression.

References

[1] W. Richard Stevens, "TCP/IP Illustrated, Vol.1: The protocols," New York: Addison Wesley, 1994.

[2] T. Ojampera, R. Prasad, "An Overview of Air Interface Multiple Access for IMT-2000/UMTS," IEEE Communication Magazine, vol. 36 n. 9, Sept. 1998, pp. 82-95.

[3] ETSI SMG2, "Evaluation Report for ETSI UMTS Terrestrial Radio Access (UTRA) ITU-R RTT Candidate," Sept. 1998 (http://www.itu.int/imt/2-radio-dev/reports/etsi/eva_utra.pdf).

[4] V. Jacobson, "Compressing TCP/IP Headers for Low Speed Serial Links," Request for Comment 1144, February 1990.

[5] M. Degermark, M. Egan, B. Nordgen, S. Pink, "Low Loss TCP/IP Header Compression for Wireless Networks," in Proc. of ACM Mobicom'96, New York, USA, November 1996.

[6] M. Degermark, "IP Header Compression," Request for Comment 2507, February 1999.

[7] S. J. Perkins, M. W. Mutka, "Dependency Removal for Transport Protocol Header Compression over Noisy Channels," in Proc. of IEEE ICC'97, Part 2 of 3, Montreal, Canada, June 1997, pp. 1025-1029.

[8] A. Giovanardi, G. Mazzini, M. Rossi, M. Zorzi, "Improved Header Compression for TCP/IP over Wireless Links", IEE Electronic Letters, vol. 36, n. 23, Nov. 9 2000, pp. 1958-1960.

[9] S. Floyd, T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm," Request for Comment 2582, April 1999.

[10] Third Generation Partnership Project (3GPP), "Radio Link Control (RLC) Protocol Specification," March 2000, http://www.3gpp.org.

[11] Anna Calveras, Josep Paradells Aspas, "TCP/IP over wireless links: Performance Evaluation," Vehicular Technology Conference, 1998. VTC'98. 48th IEEE. Volume: 3, 1998, pages: 1755-1759.