Channel State Dependent Robust TCP/IP Header Compression for 3G Wireless Networks

Yichuan Wu, Limin Sun, Jianping Zheng, Kui Huang, Yong Liao

Institute of Software
Chinese Academy of Sciences
Beijing 100080, P.R. China
{wyc,slm,zjp,hk,liaoyong}@iscas.ac.cn

Abstract

The convergence of mobile technologies and Internet protocols makes "all-IP network" be the actual trend of development for future telecommunication system. In such a system, both wired and wireless parts will be built on a common network platform based on TCP/IP protocol. However, TCP/IP protocol headers are relatively large and need to be compressed to save radio channel bandwidth (the most expensive and scarce resource of the whole wireless system), especially for some services whose payload size is small. In this paper, we propose and analyze a channel state dependent robust TCP/IP header compression algorithm (CSDROHC) for 3G wireless networks. Through adjusting the dimension of Variable Sliding Window (VSW) of W-LSB encoding in header compressor with the accurate estimation of wireless channel state; CSDROHC can achieve the good balance of compression ratio and error-resistant robustness for its adaptive use in wireless links. Simulation results are presented to demonstrate the effectiveness of CSDROHC over wireless link.

1. Introduction

The future telecommunication systems will be surely characterized by the convergence of mobile technologies and Internet protocols, achieved through a network platform totally based on TCP/IP [1,2] protocols ("all-IP network"[3]). The third generation (3G) mobile communication system [4] will be the most important realization of this concept. However, because of the encapsulation process of each layer of the hierarchical TCP/IP architecture, a substantial part of the radio bandwidth will be wasted for the transmission of control information (header) that does not have any specific function for the management of radio channel itself.

Moreover, for applications characterized by a small payload, a large header can introduce a significant extra delay that makes unrealizable the service on the TCP/IP platform. For wireless system, these problems are becoming of increasing importance for two principal reasons: the limited bandwidth available and the wide use of new protocols (as IPv6 [5] and Mobile IP [6]) with variegated and more complex headers. For example an application based on a TCP/IPv4 protocol suite, whose header length is 40 bytes, increases to 60 bytes with a TCP/IPv6 stack, and with mobile IPv6, the header grows to 100 bytes.

Since being the most expensive and scarce resource of the whole 3G wireless systems, the radio bandwidth must be utilized efficiently in order to provide mass-market services at reasonable prices. The overhead introduced by TCP/IP protocol headers must be reduced by header compression. Through the significant reduction of the header size, for the same application we have a lower demand of bandwidth (lower cost) as well as greater efficiency and responsiveness.

The error-proneness (the bit error rate as high as 10e-3, even 10e-2) and the large delays (the round trip time as high as 100-200ms) of cellular links make header compression as defined in [7,8] perform less than well. A viable header compression scheme for usage in cellular systems must produce very small headers to enable efficient usage of the scarce spectrum while still being robust to the error patterns.

On the other hand, all existing header compression schemes do not take wireless channel state into consideration when designed. They just adopt variations of encoding methods and repair mechanisms to minimize the number of lost packets when error happens over wireless links. If the current channel state can be predicted, the header compression scheme may take some actions and mechanisms in advance of compressing headers. Obviously, this type of header compression scheme may suit much better to be used over wireless links.

In this paper, a channel state dependent robust TCP/IP header compression (CSDROHC) algorithm is proposed and analyzed. By adjusting the dimension of Variable Sliding Window (VSW) of Window-based Least Significant Bits (W-

Supported by National Natural Science foundation of China under grants No.60272078.

LSB) encoding in header compressor with the accurate estimation of wireless channel state, CSDROHC can not only compress the TCP/IP headers robustly, but also achieve rather high efficiency. We present experiment results that demonstrate the effectiveness of this algorithm over simulated wireless links.

This paper is organized as follows: Section 2 briefly introduce the state of art of TCP/IP header compression schemes, Section 3 describe the design of CSDROHC scheme at detail, Section 4 show the performance evaluation of CSDROHC, and Section 5 gives concluding remarks.

2. TCP/IP Header Compression: State of Art

Header compression is possible for the presence of a significant redundancy among header fields, both within the same packet header but, in particular, among consecutive packets belonging to the same packet stream. Thus, by sending static fields information only initially, and utilizing dependencies and predictability for other fields, the header size can be significantly reduced for most packets.

Generally, header compression algorithms maintain, at both the compressor and the decompressor sides, a shared state called context including relevant information which compression and decompression are done relative to. Normally, the decompressor will keep synchronized with the compressor, thus it can decompress the compressed packet header correctly. But when packets are lost over the link, the decompressor context will be brought out of sync with compressor context. and decompressing of subsequent headers will fail. This effect, so-called error propagation, which will last until the contexts are brought into synchronization by some means will degrade the performance of header compression, especially on relative high BER link, such as wireless channel.

The original TCP/IP header compression scheme is proposed by Van Jacobson (VJHC) in [7]. In most cases, VJHC can compress the 40 octets full TCP/IPv4 header to only 4 octets and improves the TCP performance significantly on low-speed links. However, due to the differential encoding resulting in frequent context out of sync once a delta lost on the link, its performance on a high BER wireless link has a break-down [8,9].

Successively, a number of other protocols have been written. Noteworthy is the algorithm proposed by Degermark (IPHC) in [8] who suggests two simply mechanisms, *twice* algorithm and the explicit header request mechanism to improve the header compression synchronization robustness. However, the twice algorithm only works well under some assumptions. The explicit request mechanism needs a feedback channel and its performance is impacted significantly by the link round trip time. On lossy links with long round trip times, such as most wireless links, IPHC does not perform well [8,9].

A viable TCP/IP header compression scheme for usage in wireless systems is needed which must produce very small headers while still being robust to the error patterns. ROHC (RObust Header Compression)[10], the header compression scheme that the homonymous workgroup of the IETF (Internet Engineering Task Force) is developing, satisfies all the requirements for a valid header compression scheme that is destined to a wireless link. However ROHC is not able to manage, efficiently, a stream based on TCP. ROHC+[11] proposed a specific header compression profile for TCP streams within the ROHC platform. This scheme is based on a distinct management of data and acknowledgement streams associated to the TCP stream. To reduce error propagation, ROHC+ adopts a robust encoding technique (W-LSB) and the synergy between three repair mechanisms for damaged compressed packets.

3. Channel State Dependent Robust TCP/IP Header Compression: CSDROHC

In this paper, we propose a channel state dependent robust TCP/IP header compression (CSDROHC) algorithm for 3G wireless networks. In CSDROHC, header compressor adopts a robust encoding technique which is called window-based least significant bits (W-LSB) encoding algorithm. We propose to use the accurate estimation of wireless channel state to control the size of Variable Sliding Window (VSW) of W-LSB encoding in order to achieve a good balance between compress ratio and robustness. CSDROHC may be adaptable to the burst error and delay characteristics of wireless links over which it is used.

Before introducing CSDROHC algorithm it is important to brief the functionalities of W-LSB encoding that are most related to our work.

3.1 W-LSB encoding

Previous header compression schemes VJHC and IPHC do not perform well on wireless links, which are characterized by high bit error rate and transmission delay because the decompressor is required to use always the same reference base value that was used by the compressor to in order to decompress the encoded value correctly. However, the packet including this' reference base value could be lost on the channel because of bad channel conditions. The innovative Window-based Least Significant Bits encoding (W-LSB encoding) does not impose this restriction because it tries to encode a value based on a group of reference base values included by the sliding window (VSW). The decompressor can decompress the encoded value correctly once any value in VSW can be delivered successfully. By using W-LSB encoding, the compressor would not come into the inconsistency with the decompressor unless all values in the VSW are lost.

W-LSB encoding is used for header fields whose values are always subject to small changes among consecutive packets (*changing fields*). With W-LSB encoding, the k least significant bits of the field value are transmitted instead of the original field value because the Most Significant Bits (MSB) remain relatively constant during many sessions. The value of k is calculated in the compressor using N reference values included in the VSW: for each reference value *v_ref*, *k_ref* is determined so that the value to compress belongs to the interpretation interval *f(v_ref, k_ref)*

$$f(v\_ref, k\_ref) = [v\_ref - p, v\_ref - 2^\wedge(k\_ref - 1) - p]$$

Where p is an integer.

<------ interpretation interval (size is 2^k) ------->

```
|------------+--------------------------------------|
v_ref - p  v_ref              v_ref + (2^(k_ref-1)) - p
```

For any value *k_ref*, the *k_ref* least significant bits will uniquely identify a value in *f(v_ref, k_ref)*. The parameter p is introduced so that the interpretation interval can be shifted with respect to *v_ref*. Following, right value of k is the maximum among the N values of *k_ref* calculated above. After receiving k bits, the decompressor derives the original value using a previously received value as reference.

It should be clear how the dimension of VSW influences k and therefore the compression ratio too as k is typically a monotonous increasing function of the number of reference values N, i.e. the dimension of VSW. When N is small the scheme gives high compression ratio but losses of packets are not allowed on the channel. Otherwise, when VSW is large the scheme has strong robustness but low compression ratio and low bandwidth efficiency. When the compressor knows what values in VSW have been received by the decompressor, VSW can be shrunk to obtain a rather high compression ratio. To shrink the VSW, the compressor needs some means to get feedbacks that indicate what value has been received by the decompressor.

Considering the features of W-LSB encoding, the header compression mechanism we are going to propose introduces an innovative algorithm estimating optimum VSW dimension in wireless links: the channel state based robust header compression.

3.2 CSDROHC algorithm

Many different variations of methods to estimate the wireless channel state have been proposed in the literature. In this paper, we adopt the method based on RTS/CTS handshake mechanism used in [12] to predict channel state.

The header compressor sends an RTS (ready-to-send message) to the decompressor, indicating that it has n packets with compressed headers to send. The decompressor may not receive the RTS correctly due to errors. If it does, it replies with a CTS (clear-to-send reply) accepting the transmission. The compressor may also not receive the CTS reply, because, for example, the link is currently in a bad state. In this last case too he will perceive an RTS-CTS failure. If the compressor fails with the RTS-CTS transmission, it persists, up to a specific maximum number of times. We assume that the probabilities of unsuccessful RTS-CTS completion are incorporated in the two-state (good state, and bad or faded state) Markov model that we use to model the link between the compressor and decompressor. The RTS-CTS message exchange has been used for the purpose of probing the state of the channel. However, RTS-CTS use imposes an overhead to the channel throughput. Since the size of RTS and CTS packet (in our simulation, they are both set to be 1 byte) is much smaller than that of an actual packet, the throughput loss introduced by RTS/CTS is much less. What's more, when the value of n is set to be larger than 1, the overhead of RTS/CTS becomes even less. But in this case, the accuracy of channel state estimation decreases.

Many studies in literature such as [13, 14] have established that finite state Markov models can be effectively used to characterize the error behaviour of wireless links. We adopt a variation of this approach in our work: we model the actual links as two-state Markov models. In header compressor, the dimension of VSW is adjusted according to the current state of link. When the link is in good state, the compressor should decrease the dimension of VSW to get a bigger compression ratio. Otherwise, when the link is in bad state, the dimension of VSW should be increased to get a better robustness to resist the non-trivial packet loss in wireless link. Ideally we would like the compressor to have exact knowledge of the state each link is in. But this in reality never happens. So in our simulations the compressor keeps its estimate of how good it thinks the link is.

The goodness of the link is expressed through parameter g. The compressor updates this parameter whenever n packets headers are to be compressed according to the current state of channel. The parameter g is allowed to take a number of integer values. The intuition is, that the better we think the link to the decompressor is (the bigger the g), the more we should persist when we try to transmit to the decompressor, because we indeed have better chances to succeed. What's more, the smaller we should set the dimension of VSW, because the probability of packet loss in the link is much small. On the other hand, if we think that the link is really bad, we should not lose much time on it and maintain a VSW whose dimension is big enough to assure the decompressor would not come into the inconsistency with the compressor in case of there are many packets lost in the link. In our simulations g is translated to the max number of RTS attempts allowed: if a CTS is not successfully received even after g tries at sending RTS, then g is decreased exponentially. If we succeed with less than g tries, g is increased (up to a max value) inversely proportional to the number of attempts.

4. Simulation Environment

In order to evaluate the performance of CSDROHC algorithm, we apply it into existing header compression schemes ROHC+. The new scheme implemented is called CROHC. In CROHC, our algorithm is applied to the following fields: TCP sequence number, TCP acknowledgement number, TCP window size and the extra field SNR TCP (a special field defined in ROHC+, not included in TCP/IP headers).

Our simulations have been done according to the block diagram shown in figure 1.

In our simulation, we have implemented NONE, ROHC+ and CROHC to study the effectiveness of header compression in various scenarios by the methods of comparing the performance of ftp downloads several times. NONE is the one without header compression. The connection is traced using the tcpdump program. A one-way ftp connection is considered, i.e. the packet flow from ftp server to ftp client, whereas in the reverse direction only the ACKs flow. The direct link packet generation is consumed continuous, as in a long ftp session. We use Reno algorithm. TCP over IP version 4 stack with a total of 40 bytes for an uncompressed header. Three different MTU are chosen: 168bytes, 296bytes and 576 bytes.

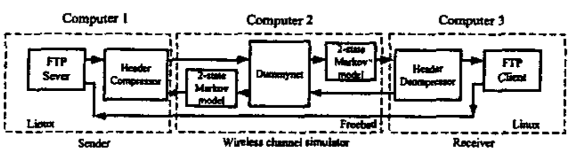

Figure 1 Simulation Scenario

Wireless channel simulator includes two parts: dummynet[13] and 2-state Markov model[14,15]. Dummynet is used to simulate bandwidth and delay management of wireless channel. In our experiments, the wireless link in UMTS system is simulated shown in table 1.

Table 1Wireless Link in UMTS System

Link type	Bandwidth (kbit/s)	Delay (ms)	BW*Delay
UMTS[16]	384	70	3.3kb

2-state Markov model illustrated in figure 2 is used to simulate the burst error characteristic of wireless channel. The good state represents the error free behaviour of the link, while the bad state represents a fade. In each state, bit errors are considered to be uniformly distributed whose mean bit error rate respectively is B_E and B_G. The durations of each state are also uniformly distributed with average durations T_B and T_G. The bit error rate can be got through setting the different values of B_B, B_G, T_B and T_G.

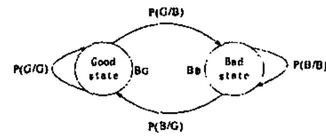

Figure 2 2-state Markov Model

In addition, the feedback link used in simulation for header compression is supposed to be independent of wireless link although they have the same link characterizations.

5. Simulation Results and Analysis

In our simulation, we compared the performance of NONE and ROHC+ and CROHC schemes in UMTS wireless link, by using the following performance criteria:

- Throughput is the ratio between the total amount of payload bytes correctly delivered to the ftp client and the time of connection (100 seconds);
- Overhead is the ratio between header bytes and total bytes transmitted by the ftp server;
- Goodput is the ratio between the total amount of payload bytes correctly delivered to the ftp client and the total amount of bytes transmitted over the wireless link.

Table 2 shows the overhead versus channel BER in each scheme. The overhead introduced by TCP/IP protocol headers decreased significantly when using ROHC+ and CROHC schemes. The overhead when using CROHC is less than that when using ROHC+, which also shows that CROHC has a larger header compression ratio than ROHC+.

Table 3 and table 4 illustrate the throughput and goodput plotted against channel BER over UMTS wireless link for various schemes. The improvement of throughput and goodput shows positive effect on TCP/IP transmission imposed by ROHC+ and CROHC header compression schemes. The throughput and goodput of CROHC is better than that of ROHC+.

Table 2, table 3 and table 4 also show that ROHC+ and CROHC schemes assure good values of overhead, throughput and goodput for every value of channel BER, but they obtain the best performance when the BER is not excessively high ($<10^{-4}$), since in these conditions TCP SN and ACK SN fields are sufficiently regular (due to the small number of packet losses) and the compressor can reach the SO state more frequently and for long periods. On the contrary, under bad channel conditions, the high number of packet losses implies a great number of retransmissions and duplicated ACKs, restraining the compressors in their transmission towards the SO state.

Furthermore, the performance of CROHC is better than that of ROHC+. It is suggested that our proposed TCP/IP robust header compression algorithm can achieve better compression ratio and robustness performance since it can adjust the dimension of VSW in W-LSB encoding quickly when channel enter into different states so as to avoid more packet losses.

Table 2 Overhead% Versus Channel BER

MTU		BER						
		10^{-6}	$5*10^{-6}$	10^{-5}	$5*10^{-5}$	10^{-4}	$5*10^{-4}$	10^{-3}
168B	NONE	23.80						
	ROHC+	4.85	4.90	4.98	5.32	5.85	6.59	9.84
	CROHC	4.56	4.73	4.80	5.11	5.57	6.29	9.14
296B	NONE	13.50						
	ROHC+	2.45	2.50	2.55	2.70	3.00	3.65	9.01
	CROHC	2.30	2.37	2.46	2.52	2.88	3.32	8.59
576B	NONE	6.90						
	ROHC+	1.23	1.25	1.26	1.28	1.57	2.43	3.95
	CROHC	1.21	1.23	1.24	1.26	1.54	2.27	3.86

Table 3 Throughput (bytes/s) Versus Channel BER

MTU		BER						
		10^{-6}	$5*10^{-6}$	10^{-5}	$5*10^{-5}$	10^{-4}	$5*10^{-4}$	10^{-3}
168B	NONE	15326	12390	9012	6370	3110	1625	523
	ROHC+	16002	13520	10954	6713	3371	1769	690
	CROHC	16197	14271	11007	6864	3428	1839	700
296B	NONE	31865	28013	21460	10023	6004	2143	459
	ROHC+	32964	29698	23793	11980	6392	2847	471
	CROHC	33223	30184	24298	12053	6521	2945	489
576B	NONE	43915	41342	36224	27421	11342	3278	128
	ROHC+	46634	45298	40994	28673	12375	6783	153
	CROHC	47893	46076	41191	29123	13009	6964	202

Table 4 Goodput% Versus Channel BER

MTU		BER						
		10^{-6}	$5*10^{-6}$	10^{-5}	$5*10^{-5}$	10^{-4}	$5*10^{-4}$	10^{-3}
168B	NONE	72.0	71.8	71.4	70.8	68.2	65.8	56.9
	ROHC+	92.4	91.6	90.5	89.1	85.8	80.8	67.9
	CROHC	94.3	93.1	91.0	90.1	88.0	82.4	71.7
296B	NONE	82.1	82.0	81.6	80.3	[illegible]	74.0	61.5
	ROHC+	95.8	95.5	94.2	91.9		83.7	64.4
	CROHC	96.3	96.1	94.5	93.0	91.3	86.8	69.5
576B	NONE	85.0	86.6	86.2	85.2	82.7	73.2	56.8
	ROHC+	97.7	97.3	96.5	94.9	90.8	80.1	61.3
	CROHC	98.1	97.8	97.2	96.1	92.2	87.1	70.0

6. Conclusions

This paper proposes a new channel state dependent robust TCP/IP header compression (CSDROHC) algorithm for 3G wireless networks. This algorithm can achieve the good balance of compression ratio and error-resistant robustness for the adaptive use in wireless link, by adjusting the dimension of Variable Sliding Window (VSW) in W-LSB encoding with the accurate estimation of wireless channel state With applying of this algorithm, we implement a new header compression scheme: CROHC.

Simulation results show that the performance of TCP with ROHC+ and CROHC schemes are both much better than TCP without header compression both in compression ratio and error-resistant robustness when used in wireless link. ROHC+ and CROHC schemes can obtain the best performance when channel BER is not excessively high ($<10^{-4}$). The performance of CROHC is improved to some extent when compared with that of ROHC+, which shows the enhancement effect imposed by our proposed CSDROHC algorithm.

References

[1] W. R. Stevens, *TCP/IP Illustrated Volume 1 – The Protocolos*, Addison-Wesley, 1994.

[2] Postel J. Internet protocol, Request For Comments RFC791. DARPA September 1981.

[3] 3GPP TR 23.922 v.1.0.0 (1999-07), *Architecture for an All IP Network*, July 1999.

[4] 3GPP TR 23.821 v.1.0.1 (2000-07), *Architecture principle for Release 2000*, July 2000.

[5] Deering S, Hinden R. Internet protocol, version 6(IPv6), Request For Comments RFC2460.IETF,December 1998.

[6] Montenegro G., Internet Draft (work in progress), *Reverse Tunneling for Mobile IP*, draft-ietf-mobileip-rfc2344-bis-01.txt, march 2000.

[7] Jacobson, February 1990, RFC 1144 - Compressing TCP/IP Headers for Low-SpeedSerial Links.

[8] M. Degermark, B. Nordgren, S. Pink, February 1999, RFC 2507 -IPHeader Compression.

[9] S.Kent, R.Atkinson, IP Encapsulating Security Payload, RFC2406, November 1998.

[10] Bormann C etc. RObust Header Compression(ROHC): Framework and four profiles: RTP,UDP,ESP,and uncompressed, Request For Comments RFC3095.IETF,July 2001.

[11] G.Boggia, P.Camarda, and V.G.Squeo," ROHC+: A New Header Compression Scheme for TCP Streams in 3G Wireless Systems," IEEE ICC2002

[12] Fragouli.C, Sivaraman.V, Srivastava.M, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State Dependent Packet Scheduling", IEEE infocom 1998.

[13] URL: http://info.iet.unipi.it/~luigi/ip_dummynet

[14] F. Swarts, and H Ferreira, "Markov characterization of digital fading mobile VHF channels," IEEE Trans. on Vehicular Technology, vol. 43, no. 4, pp. 977-985, Nov. 1994.

[15] H. Wang, and N. Moayeri, "Finite State Markov Channel – a useful model for radio communications," IEEE Trans. on Vehicular Technology, vol. 44, no. 1, pp. 163-171, Feb. 1995.

[16] Flavio Muratore. UMTS - Mobile Communications for the Future.Wiley,2000.