Robust Header Compression for More Efficiency in Real-Time Transfer Date

Borislav Naydenov[1], Petar Petrov[2] and Alexander Milev[3]

Abstract – **The Robust Header Compression (ROHC) is a method to reduce the traffic. In the context of the problem, the objective is to analyze the principle of use and technical constraints of implementing ROHC and make a comparative assessment of appropriate use in wireless communication systems in terms of speed up of data transmission, occupied VoIP bandwidth and reliability of decompressed information.**

Keywords – **Robust Header Compression, VoIP traffic, Wireless LAN.**

I. Introduction

The evolution of telecommunication networks requires the use of approaches which allow to reduce the time, as in the implementation of interactive links and the transfer of data. One of the possible ways to optimize the time when transferring large files is through data compression. Data compression is widely studied and optimized. While the networks evolve to provide more bandwidth, applications, services and customers of these applications compete for this band. For network operators is essential to provide Quality of Service (QoS) in order to attract more customers and encourage them to use their network as much as possible.

Wireless networks are characterized by probability of bit error rates, mainly due to interference and greater delay. They are difficult to reach the requirements for wider bandwidth. It is therefore necessary available resources to be applied as efficiently as possible. The TCP header compression (HC) reduces overhead. The reduction in overhead for TCP traffic results in a corresponding reduction in delay; TCP header compression is especially beneficial when the TCP payload size is small, for example, for interactive traffic such as Telnet. In many services and applications such as VoIP, interactive gaming, messaging and other, payload in the IP packet is almost the same size, even smaller than the excess information, such as header. For connections of the type end to end, including multiple nodes, these protocol headers are important, but a links with type hop-to-hop, this service information is useless. It is possible to perform compression of header information, which will provide up to 90 percent size reduction of packet length. Thus reduces the load and save bandwidth of connection. IP header compression also provides significant advantages such as reduction of packet loss and improved interactive response time [1].

[1]Borislav Naydenov is with the Faculty of Electronic, Technical University - “Studentska” 1, Varna, Bulgaria, E-mail: borna@abv.bg.

[2]Petar Petrov is with the Faculty of Electronic, Technical University - “Studentska” 1, Varna, Bulgaria, E-mail: rls1@abv.bg

[3]Aleksandar Milev is with the Faculty of Mathematics and Informatics, University of Shumen, Bulgaria, E-mail: alex_milev@yahoo.com.

II. Actuality of the Problem

If necessary to transfer multiple small volumes of information with minimum delay in packet networks is possible to use a standardized approach to compress the header of packets. This question is especially actual in the transmission of voice information using Internet Protocol (Internet Protocol - IP). In slow serial connections such as wireless links in wireless local area networks (WLAN) and mobile communication networks, phone lines and other low speed lines, the use of packets with HC results in a significant reduction in the whole time to transmit information. There are other reasons that which demand to use the compression on slow and medium speed lines. The duration is reduced for interactive response time and packet loss rate over lossy links is getting smaller. This header compression scheme does not require that all packets in the same stream passes through the compressed link. However, for TCP streams the difference between subsequent headers can become more irregular and the compression rate can decrease. Neither is it required that corresponding TCP data and acknowledgment packets traverse the link in opposite directions.

For wireless links are proposed option for compression, called robust - Robust Header Compression [2]. ROHC involved mechanism for compression, which aims to achieve sustainability in relation on the packet loss and maximum efficiency in compression.

In the context of the problem, the objective is to analyze the principle of use and technical constraints of implementing ROHC and make a comparative assessment of appropriate use in wireless communication systems in terms of speed up of data transmission, occupied bandwidth and reliability of decompressed information.

III. Basic Approaches of Packet Header Compression

As a first implementation of the compression algorithm may point HC of packets of the Transport Protocol (Compressed Transmission Control Protocol - CTCP), standardized in RFC 1144 [3]. To make the optimization of service information, which are conveyed repeatedly with each packet in the slow line between two nodes, we need a compression protocol to be installed on network devices at both ends of the line. In this compression is achieved reductions of greater part of the header information of the Transmission Control Protocol (TCP), as is done shrinking it from 40 octets to 4 octets. Compressor, which implemented CTCP, detected retransmitted repeatedly and no amended on transport level information. After initial sending it stops retransmission. In the headers of next packets are put delta encoding, which characterizes the successive changes in next header fields. This is known as a mechanism to adjust the context. For it is not necessary any additional signaling between the compressor and decompression.

The HC was improved when it was used to IP Compression (IP Header Compression - IPHC) with CTCP, RFC 2507 [4]. The mechanism to adjust the packets in CTCP is reinforced with a negative acknowledgment, called message CONTEXT_STATE, which accelerates the correction, when was detected an error. In RFC 2507 described how decreased IP and TCP headers per hop over point to point links.

IPHC does not compress the header of Real-time Transport Protocol - RTP. Compression of RTP header (Compressed RTP - CRTP) is an extension of the IPHC. CRTP compresses the 40 octets header packets including protocols IPv4/UDP/RTP to 2 octets. This is true if not activated error checking to header of the User Datagram Protocol – (UDP). Upon activation of this check the compression make the length of header to be 4 octets. This compression is not from one end to another end of connection, but from line to line [5]. Compression from line to line is characterized by good performances in which are small delays and low losses. IP / UDP / RTP compression is used with IPv4 and IPv6.

CRTP is characterized by loss of packets on slow lines, which in turn leads to the failure of decompression of next successive packets. With the implementation of IPHC is introduced a local adjustment mechanism called TWICE. But with this mechanism, CRTP does not reduce the number of lost packets. To improve the stability of the compression algorithm is proposed an improved version of CRTP - eCRTP [5], which is at the expense of reduced compression.

IV. ROBUST HEADER COMPRESSION

ROHC introduces a new mechanism for compression, which increases the resistance on the compression efficiency [2]. ROHC is expected to be the preferred compression mechanism over links where compression efficiency is important. However, ROHC was designed with the same link assumptions as CRTP, e.g., that the compression scheme should not have to tolerate misordering of compressed packets between the compressor and decompressor, which may occur when packets are carried. CRTP does not perform well on such links: packet loss results in context corruption and due to the long delay, many more packets are discarded before the context is repaired. To correct the behavior of CRTP over such links, a few extensions to the protocol are specified. It is based on Compressed Real-time Transport Protocol (CRTP), the IP / UDP / RTP header compression described in RFC 2508. The extensions aim to reduce context corruption by changing the way the compressor updates the context at the decompressor. With these extensions, CRTP performs well over links with packet loss, packet reordering and long delays [6]. ROHC was developed with wireless links as the main target, and introduced new compression mechanisms with the primary objective to achieve the combination of robustness against packet loss and maximal compression efficiency. If a packet that includes an update to some context state values is lost, the state at the decompressor is not updated. The shared state is now different at the compressor and decompressor. When the next packet arrives at the decompressor, the decompressor will fail to restore the compressed headers accurately since the context state at the decompressor is different than the state at the compressor. Decompressor fails not when a packet is lost, but when the next compressed packet arrives. If the next packet happens to include the same context update as in the lost packet, the context at the decompressor may be updated successfully and decompression may continue uninterrupted. If the lost packet included an update to a delta field such as the delta RTP timestamp, the next packet can't compensate for the loss since the update of a delta value is relative to the previous packet which was lost. But if the update is for an absolute value such as the full RTP timestamp or the RTP payload type, this update can be repeated in the next packet independently of the lost packet.

A "headers checksum" is inserted by the compressor and removed by the decompressor when the UDP checksum is not present so that validation of the decompressed headers is still possible. This allows the decompressor to verify that context sync has not been lost after a packet loss. Enhanced CRTP achieves robust operation by sending changes multiple times to keep the compressor and decompressor in sync. This method is characterized by a number "N" that represents the quality of the link between the hosts.

The Lightweight User Datagram Protocol (UDPLite), which is similar to the User Datagram Protocol (UDP) is propound [7]. It can serve to applications in error-prone network environments that prefer to have partially damaged payloads delivered rather than discarded. If this feature is not used, UDP-Lite is identical to UDP.

The Robust Header Compression (ROHC) protocol provides an efficient, flexible, and future-proof header compression concept. It is designed to operate efficiently and robustly over various link technologies with different characteristics. To improve and simplify the ROHC specifications, the new RFC explicitly defines the ROHC framework and the profile for uncompressed separately [8]. More specifically, the definition of the framework does not modify or update the definition of the framework specified by RFC 3095.

An updated version was defined for compression of RTP/UDP/IP, UDP/IP, IP and ESP/IP (Encapsulating Security Payload) headers [9]. Additional profiles for compression of IP headers, and UDP-Lite headers were later specified to complete the initial set of ROHC profiles for each of the above mentioned profiles, and the definitions depend on the ROHC framework as found in RFC 4995. Instead of compressing all RTP or all TCP packets that are going through network, it is possible to make RTP header compression to compress only those packets that belong to a class called "voice."

In [10] is proposed available method for the determination of Variable Sliding Window *(VSW)*, using a time interval, called Memory timeout interval during which the timeout values are registered. Indicating the actual value with *VSW (n)* and the previous value with *VSW (n-1)*. At current time, Variable Sliding Window dimension is expressed as:

$$VSW(n) = VSW_{\min}(n) + VSW_{BER}(n) \quad (1)$$

$VSW_{MIN}(n)$ is defined as follows:

$$VSW_{MIN}(n) = \left\lceil \beta . \frac{T.\left(\frac{MaxPkSize}{N}\right)}{MeanArrivalTime} \right\rceil \quad (2)$$

where MaxPkSize is the size, expressed in bit, of the biggest compressed or not compressed IP packet transmitted on Wireless channel; N bits are the maximum payload of a baseband packet; T ms is twice Wireless slot-time; MeanArrivalTime is the mean time gap between packets in the same application flow; β is a constant bigger than one.

V. OPNET IMPLEMENTATION

For investigation into VoIP traffic using ROHC we will use software OPNET [11], [12]. In this section we describe the basic steps in creating a model for the selected topology, as seen on Fig.1.

Select objects Application Config, Profile Config and QoS Attribute Config. Take two subnet models and ip32_cloud, which will simulate various traffic in our setting. With subnet elements we open two new areas, called sublevels of the topography. They are shown on Fig.2. Insert mobile computers that we chose – respectively 17 in the subnet London and 19 for subnet – Varna. In subnets located into Progect Editor of program, we insert 36 number of wireless ethernet_wkstn. To loading the links in a network we include two ethernet_server - respectively for HTTP and FTP traffic.

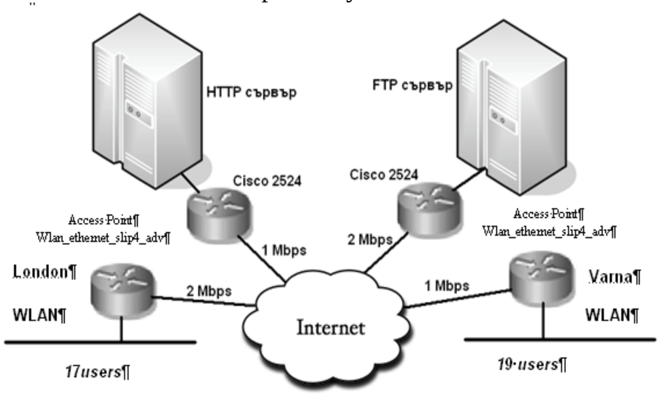

Fig. 1. Opnet project

The main components constitute our end-to-end system: the RTP/UDP/IP generator and sink, the RTP/UDP/IP header compressor/decompressor, and the transport network. The remaining sections are logically arranged in the organization seen in Fig. 1. This project can be regarded as two identical data paths, traveling in opposite directions.

Each data path consists of some traffic sources, each sending a unique compressed VoIP RTP/UDP/IP traffic stream to an aggregator. The aggregator combines the packet streams from the random number generators into a single point-to-point link, which is connected to the RTP/UDP/IP ROHC compressor/decompressor. The RTP/UDP/IP header compressor compresses the packet headers, which coming by line as necessary and sends each packet through the WLAN encapsulator. Sending the compressed packet, the WLAN encapsulator encapsulates the packet into an ethernet cell, and transmits the cell over a wireless channel to the terminal decapsulator through access point. The decapsulator removes the compressed packet from the ethernet cell and sends the packet to the RTP/UDP/IP header decompressor. The decompressor reconstructs the packets based upon its stored context state information.

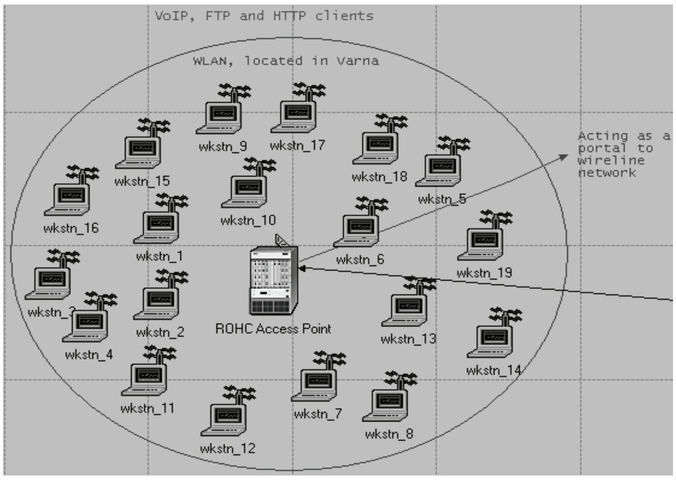

Fig. 2. WLAN subnetwork, located in Varna

The same data transfer process takes place in the opposite direction, using the counterparts of the same components used in the first direction of data transfer.

In the simulation we set up several streams of information such as voice traffic, FTP traffic and HTTP traffic. Then we create profiles, which can be set to already created applications to define the traffic generated for this profiles. Once the profiles have been created, we set up bi-directional compression of VoIP traffic from terminal to access point and contrariwise. We define the type of VoIP traffics generated by the application. For one of the stream we choose voice with parameters G.711, PCM Quality Speech - 64kbps, ToS - Interactive Voice, without RSVP parameters. For other flows we choose encoder scheme - G.723.1, G.728, G.729.

VI. SIMULATION RESULTS

Different main possible statistics are received, like packets end-to-end delay (sec) – Fig. 3 and traffic sent (bytes/sec), total delay of a packets from end to end (s), the speed with which information transmitted on the channel (bit/sec) and throughput of VoIP traffic through WLAN without HC (1) and with HC (2) – Fig.4. The analysis of current researches, in order to visualize the main results, some of them are presented in graphic form. The other parameter which will take into consideration is utilization of bandwidth (%) of VoIP traffic.

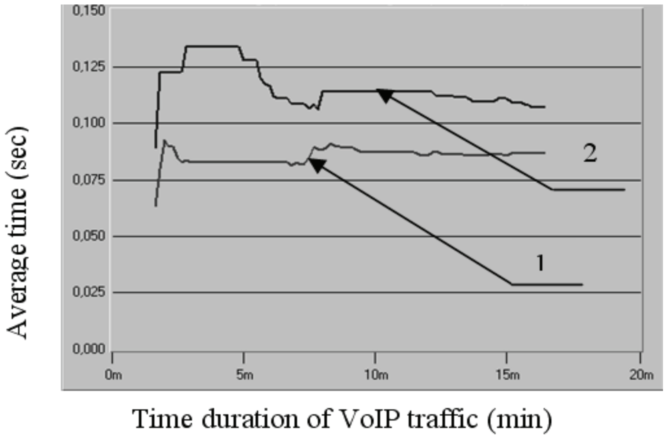

Fig. 3. Average packet end to end delay time (s) of VoIP traffic with ROHC (1) and without ROHC (2)

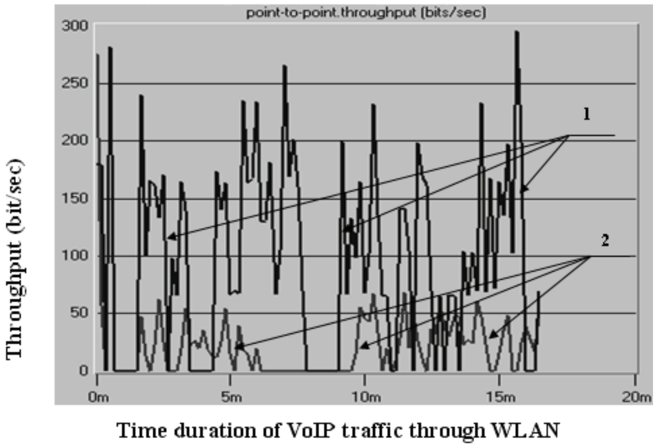

Fig. 4. Throughput of VoIP traffic through WLAN without HC (1) and with HC (2)

Table I shows the results of speed up of data transmission with HC, occupied bandwidth of VoIP traffic using priority (QoS), packet loss, reliability and Variable Sliding Window VSW_{MIN} *(n)* when is used ROHC and without ROHC.

TABLE I
ROHC PARAMETERS

ROHC Parameters	With ROHC	Without ROHC
Speed up of FTP date transmission	17 s	22 s
Occupied VoIP bandwidth	63,2%	74,5 %
Packet loss	4,1%	3,8 %
Reliability	10^{-2}	$10^{-2} – 10^{-3}$
$VSW_{MIN}(n)$	41	47

The voice codecs generating a VoIP packet every 20 ms, while the UDP checksums were enabled. In this scenario with ROHC scheme, we are interested in two result aspects: packet loss and occupied bandwidth. The first column represents the header compression scheme's performance regarding bandwidth savings, while the second column evaluates the parameters without use of ROHC.

VII. Conclusions

In narrowband networks application of HC is reflected in improvements in response time caused by the smaller size of the packets. The results prove that the smaller size of the packets also reduces the possibility of errors. For VoIP transmission quality is increased while using less bandwidth. HC improves quality of transmission and speeds up the network. It is a result of saving in bandwidth, reduces packet loss, improves interactive response time and reduces the cost of infrastructure due to including more users per channel.

HC is hop-to-hop process and is not applicable to end to end connection. For each node in the IP network, it is necessary to uncompress the packet to be able to perform operations such as routing, ensuring quality of service (QoS), etc. ROHC is best applied to specific links in the network, characterized by relatively narrowband, high bit error rate and more delay.

References

[1] EFFNET AB, An introduction to IP header compression, White paper, February, 2004.
[2] C. Bormann, C. Burmeister etc., RObust Header Compression (ROHC), Framework and four profiles: RTP, UDP, ESP, and uncompressed, RFC 3095, July, 2001.
[3] V. Jacobson, "Compressing TCP/IP headers for low-speed serial links", RFC 1144, February 1990.
[4] M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression", RFC 2507, February, 1999.
[5] S. Casner, V. Jacobson, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, RFC 2508, February, 1999.
[6] T. Koren, S. Casner, J. Geevarghese, B. Thompson, and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, July 2003.
[7] L-A. Larzon, S. Pink etc., The Lightweight User Datagram Protocol (UDP-Lite), RFC 3828 July 2004.
[8] L-E. Jonsson, G. Pelletier, K. Sandlund, The RObust Header Compression (ROHC) Framework, RFC: 4997, July 2007
[9] G. Pelletier,K. Sandlund, RObust Header Compression Version 2 (ROHCv2):Profiles for RTP, UDP, IP, ESP and UDP-Lite, RFC 5225, April 2008
[10] L. Marzegalli, M. Masa, M. Vitiello, Adaptive RTP/UDP/IP Header Compression for VoIP over Bluetooth, ICC – 2009.
[11] URL:http://www.opnet.com.
[12] N. Rendevski, A. Tentov, E. Vlahu-Gjorgievska, B.Ristevski, FPGA Synthesis of Opensparc Processor's Cores, Network Implementations and Performances, 29th International Conference on Organizational Science Development, 2010, Portoroz, Slovenia.