

A Noval Optimization Technique for Multimedia Traffic through Header Compression for 802.11 ad-hoc Wireless Networks

Devaraju. R

Department of Electronics and Communications
Sapthagiri College of Engineering, Visvesvaraya
Technological University
Bangalore, Karnataka, India

Puttamadappa. C

Sapthagiri College of Engineering
Visvesvaraya Technological University
Bangalore, Karnataka, India

Abstract— Over few decades, new services and protocols have been integrated into wired networks. The need for incorporating these vital additions in wireless networks is also increasing at an alarming rate. In case of multimedia applications, one of the key issues for the Quality of Service (QoS) is the suitability and adaptability of these additions for real time applications. Wireless systems suffer from constraints like delay and jitter, which result in higher packet loss compared to the wired systems. Secondly the wireless networks are also limited by its operating bandwidth, which is very low when compared against wired systems. Hence there exists a need for optimizing the real time multimedia traffic for band limited wireless systems. Present work proposes a novel methodology to optimize the performance of peer-to-peer ad-hoc wireless networks by implementing header compression technique on Internet Protocol (IP) packets transmitted on a point to point link.

Keywords— *Ad-Hoc Wireless Network, Cellular Wi-Fi, Gateway, Header Compression Techniques, Robust Header Compression.*

I. INTRODUCTION

Existing wireless networks are mostly circuit switched, where the cellular network model supports the architectural needs such as base stations, gateways to a fixed and wired infrastructure backbone network. They are usually referred to as infrastructured networks. Wireless ad-hoc networks also known as decentralized type of wireless network provides license free frequency bands to users in a limited geographical area. In ad-hoc networks, mobile nodes (such as laptop computers, personal digital assistant and mobile phones) can communicate with each other without relying on any pre-existing infrastructure. The data transfer is carried out dynamically, based on the network connectivity. Given a scenario, wireless ad-hoc networks have more sever operating conditions as compared to the traditional infrastructured wireless networks such as limited capacity of data transfer, increased latency etc. Recently wireless Local Area Networks (LAN) has been increasingly used for multimedia traffic such as voice, video and gaming. Due to the limited bandwidth of the wireless links as compared to wired systems, header compression on IP packets is an attractive method for efficient utilization of the frequency spectrum available.

An ad-hoc network is a collection of mobile nodes connected with one other forming a temporary network without using any pre-existing fixed infrastructure or centralized administration. The topology of an ad-hoc network may change dynamically in an unpredictable manner [1]. Data transfer in ad-hoc networks are usually based on datagram switching where information packets are transmitted in a store-and-forward manner from source to an arbitrary destination. Each node functions as a router and is responsible for discovering and maintaining route information to other nodes within the network. Major advantage of ad-hoc networks is their fast deployment. The setup of an ad-hoc network is much faster than that of an infrastructure network because no time is required for building infrastructure. In contrast, having a fixed infrastructure in a network makes traffic administration much easier. However building infrastructure for a network that has a short lifecycle is generally not recommended as it is cost effective. The time and cost required to build the infrastructured network may be so long that the network service may still not be available even when the need for service ends. In such situations, an ad-hoc network is a better solution [2].

Currently wireless networks offers a wide range of IP based multimedia applications (such as voice, video and gaming services), which require more bandwidth. Multimedia applications often use Real Time Protocol (RTP), User Datagram Protocol (UDP) and IP. Each of the protocol layers adds a significant header overhead. Therefore, the higher bandwidth consumption is caused by the application itself and the IP header overhead. While the compression of multimedia payload is explored to an extent, the most promising compression gain could be achieved from IP header compression. There are scenarios where the size of IP header overhead can be greater than the payload for some services. IP header compression techniques have always played an important part of saving bandwidth over bandwidth limited links. Reducing the IP header overhead gives the network providers the possibility for a faster return of investment on their networks and simultaneously enables real-time multimedia services to be implemented by improving the bandwidth consumption, latency of the IP packets over bandwidth limited links. Generally there are variety compression techniques adopted in wireless networks, however

they are either not designed for multimedia services, or not robust in the presence of error-prone links and therefore not suitable for wireless communication [3].

In this paper, a method to improve the performance in a peer-to-peer ad-hoc wireless network by implementing header compression on IP packets is presented. The endpoints of the point to point link negotiate on a profile on header information for packets transmitted from the first endpoint to the second. A packet exchanged between two endpoints over a point to point link includes an identifier which is based on the context within the packet. Header compression algorithm attempts to maintain the context. This established context is maintained at both the ends. Header compression schemes typically will have mechanisms for detecting the context and applying appropriate compression algorithm. When multiple compression algorithms are used over the same link, there must be some way to determine that the specific compressed header belongs to a specific compression algorithm and uses the correct context for the packet flow to compress that header. The compressed header is transmitted using the identifier to specify the mechanism of which algorithm the header belongs. At the other end of the link, the decompressor receives the compressed header and uses the mechanism to ascertain which algorithm the header belongs to [4].

II. RELATED WORK ON HEADER COMPRESSION SCHEMES

Van Jacobson's Header Compression (VJHC) documented in RFC 1144 was the first of its kind introduced to improve the efficiency for a serial link. VJHC processes TCP and IP headers together to achieve better compression ratios. VJHC is based on delta coding. The differences between two header packets are referred to as the delta. Instead of transmitting the whole header, VJHC transmits only the delta. Due to this approach high compression is achieved. Simultaneously vulnerability comes along with this approach. In case anyone delta coded header got corrupted; all the consecutive packets are erroneous [5].

Robustness at the cost of minimal compression was introduced by Perkins. The delta-coding for the neighboring packets has been replaced by a reference frame. Several consecutive packets are aggregated to a frame. The first packet of a frame is sent uncompressed and the subsequent packets use the delta coding by referring to base information of the first packet. The usage of shorter delta coding range proves advantages for such situations. In this case packet loss does not cause endpoints to lose synchronization as the next frame will contain uncompressed packet at the beginning. The differences to packets towards end of a frame are larger than for those at the beginning.

IP header compression (IPHC) documented in RFC 2507 was designed with a view to working well over links with considerable packet error rates, such as wireless links. IPHC is designed to handle both TCP and UDP along with the underlying IP protocol. Compression algorithms can be applied to IPv6 base and extension headers like multicast, multi-access links, and other compression schemes which ride over UDP [6].

Compressed Real-time Transport Protocol (CRTP) documented in RFC 2508, supports compression of IP/UDP/RTP headers for low speed serial links. CRTP bears similarities to IPHC. CRTP was designed to work well with

reliable and fast point-to-point links. Experiments results highlight that CRTP is not suitable for Voice over IP (VoIP) applications. CRTP makes use of a sequence number that is carried by both compressed and uncompressed packets. The sequence number is incremented by one every time a packet is transmitted. Although the performance of CRTP is remarkable in terms of compression efficiency, most of its advantage is constrained by the fact that it has been designed for relatively error-free links with low round-trip delay.

Enhanced Compressed Real-time Transport Protocol (ECRTP) documented in RFC 3545 is an upgrade to CRTP mainly developed with intent to make CRTP work well over links with high delay, high rate of packet loss and out-of-sequence packets. To maintain synchronization the compressor sends a context update multiple times to the decompressor. ECRTP allows updating the differential RTP header values while still managing to continue compressing the underlying UDP and IP Protocols.

RObust Checksum based header COmpression (ROCCO) was introduced to adapt the channel characteristics like the packet rate and pattern of losses over the link. ROCCO is adaptable to the characteristics of the link in terms of handling pattern changes for various changing header fields. The RTP sequence number ensures that context state is kept consistent even if consecutive packet losses are encountered. ROCCO header compression technique computes a Cyclic Redundancy Checksum (CRC) and includes the latter in each compressed header. The decompressor verifies the CRC checksum and reconstructs the compressed header correctly [7].

Later a new working group was setup within Internet Engineers Task Force (IETF), named RObust Header Compression (ROHC) group. ROHC was the first header compression algorithm of its kind which mainly concentrates to perform well over links where the error rates, packet loss and round-trip times are high, such as wireless links. ROHC as in RFC3095 is a header compression scheme with profiles for three protocol suites: RTP/UDP/IP, UDP/IP and ESP/IP. ROHC turned out to be a very powerful scheme that performs well over lossy links with long round-trip delay, while achieving high compression ratios at the same time.

All the above header compression techniques can be effectively used across mobile radio channels only if there exists a feedback channel from the de-compressor to the compressor [8].

III. METHODOLOGY OF PROPOSED DESIGN

A method is proposed to improve the performance in a peer-to-peer ad-hoc wireless network by implementing Header Compression of packets transmitted on a point to point link. The endpoints of the point to point link negotiate on a profile for header information for packets transmitted from the first endpoint to the second endpoint on the point to point link. Packets exchanged between two endpoints over a point to point link include an identifier which is based on the context in the packet. Header Compression algorithm attempts to maintain a context. The Context is maintained at both the ends. Header Compression schemes typically will have mechanisms for detecting the context, and applying appropriate compression algorithm. When multiple compression algorithms is used over the same link, there must

be some way to determine that the specific compressed header belongs to a specific compression algorithm, and uses the correct context for that packet flow to compress that header. The compressed header is transmitted using the identifier to specify the mechanism of which algorithm the header belongs. At the other end of the link, the decompressor receives the compressed header, and uses the mechanism to ascertain to which algorithm the header belongs.

methods, Scenario based methods etc. The performance analysis statistics results will be recorded and displayed using appropriate graphs and charts.

We have implemented a unique header compression algorithm which is a variant of ROHC. The header compression attempts to retain the key fields of RTP header such as sequence number and timestamp. The remaining fields of RTP along with the underlying protocol are compressed similar to ROHC. Using the uncompressed RTP fields we have implemented feedback from decompressor to compressor. Feedback is immediately sent to the compressor when an error packet or lost packet is noticed. This helps the compressor to resend the lost packet again. Automatically delay will be introduced due the feedback and retransmission operations. Real time applications such as voice and video might not require feedback as packet loss to an extent can be tolerable. Also the time slot would have elapsed for voice and video before the decompressor receives the retransmitted packet. But applications such as text will contain sensitive information and all the packets will be required for decompressor. Hence a trade-off should be done whether to use feedback technique or not. Our design framework has the option to either use the feedback or disable the same.

The results are compared with the results without header compression and again interfaced to the database for further analysis.

IV. RESULTS AND DISCUSSIONS

Present work in this paper aims at developing and testing the series of communication scenarios. The coding language used is Core Java. The complete detail of software's and tools used for development is tabulated in the below section.

TABLE I. SOFTWARE TOOL KIT

Software	Version	Description
Java Development Kit (JDK)	1.7.0_15	JDK contains Java Runtime Environment (JRE) along with various development tools like Java libraries, Java source compilers, Java debuggers, bundling and deployment tools.
NetBeans	7.2.1	NetBeans is an integrated development environment (IDE). It is also an application platform framework for Java desktop.
Oracle	11g	The Oracle database is an Object-Relational DataBase Management System (O-RDBMS). The Oracle RDBMS stores data logically in the form of tablespaces and physically in the form of data files.

It is a good practice to take multiple measurements of experimental parameters like delay, throughput etc using a different packet size. While there is no correct value to represent the best result, the general approach is to consider

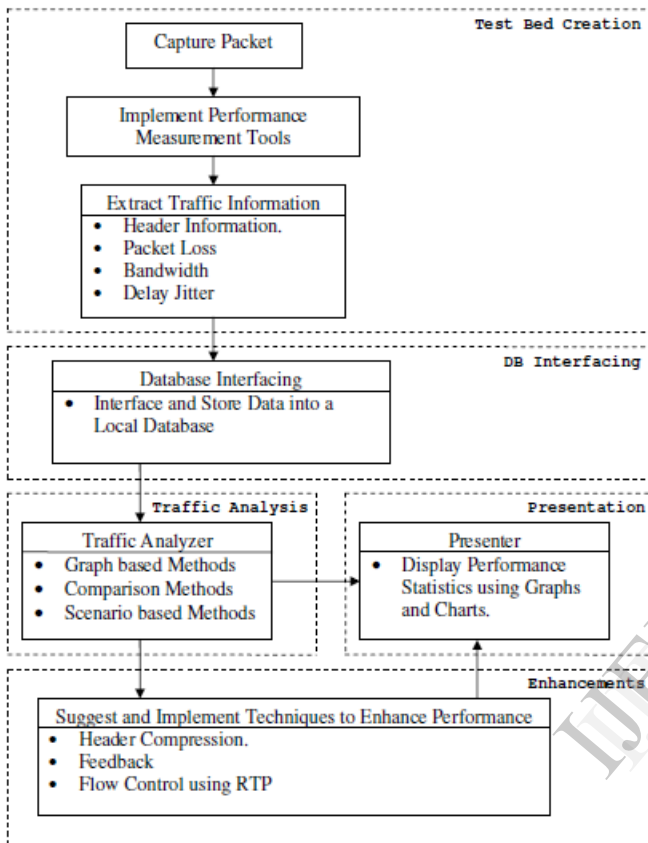


Fig. 1. Gateway Traffic Compression Architecture

The figure 1 represents the Gateway Traffic Compression Architecture used to implement this proposed idea. The flow of the proposed design can be explained with reference to the architecture. To start up with, the packets are captured from the gateway device. Here gateway device can be termed as the junction where data transfer takes from wired network to wireless network and vice versa. Packets are captured using the available open source performance measurement tools such as Wireshark, wbest etc. It may be required to combine the functionality of multiple tools to get the desired performance parameters. We are currently concentrating on the performance parameters like packet loss and bandwidth availability. The extracted information could be listed as follows: header information, packet loss, and bandwidth utilizations. The dynamically generated data from the performance measurement tools is then interfaced to the local database. This interfacing of the database is necessary for detailed analysis of the traffic across the network when new wireless users dynamically join or leave the network.

Detailed analysis on the above results is carried out using the techniques like Graphs based methods, Comparison based

the packet size depending on the type of experiment, hardware sizing and average size used for the respective technology.

Internet traffic is a mix of long and short length packets. Average internet packet length lies between roughly around 300 to 400 bytes. Short packets with length around 50 to 80 bytes are appropriate for testing the device performance by checking the number of packets processed per unit time. On the other hand long packets with length around 1200 to 1500 bytes are best suited to test the performance of the communication network. Usually header size dominates when compared to the payload size in short packets while payload ratio is on the upper hand in long packets. While each of these packet lengths offers a useful way to describe system performance, there is no concept of one-size-fits-all approach. Every enterprise network carries a different distribution of packet lengths, and for best results it's necessary to develop a custom test traffic profile to test suitability for a given network. Hence we have implemented a generic code design with the flexibility to use the packet size of our choice which suits the traffic and the network.

In order to characterize ROHC's value, a new compression metric was defined; Robust Header Compression gain (RoHCGain).

$$RoHCGain = \frac{TotalPayloadRcvdWithRoHC}{TotalPayloadRcvdWithoutRoHC} - 1 \quad (1)$$

Gateway Traffic Compression performance results obtained from the experiments is tabulated below in table 2. Payload size of 512 bytes is considered for the test.

TABLE II. PERFORMANCE RESULTS OF VARIOUS COMMUNICATION PROTOCOLS

Protocol Used	Packet Size (Bytes)	Total File Size (Bytes)	Total Time (Seconds)	Consumed Bandwidth (Mega Bytes)
Simple TCP/IP	552	1492652	0.011037193	0.001020
Simple UDP/IP	540	1460192	0.010748268	0.000993
ROHC TCP/IP	515	1392567	0.010624443	0.000982
ROHC UDP/IP	514	1389856	0.010411143	0.000979
Simple RTP/TCP /IP	564	1525112	0.019737193	0.001034
Simple RTP/UD P/IP	552	1492652	0.011037193	0.001020
ROHC RTP/TCP /IP	517	1397977	0.010684443	0.000982
ROHC RTP/UD P/IP	515	1392560	0.010688443	0.000979

The values from the previous table are shown in the figure 2 charts:

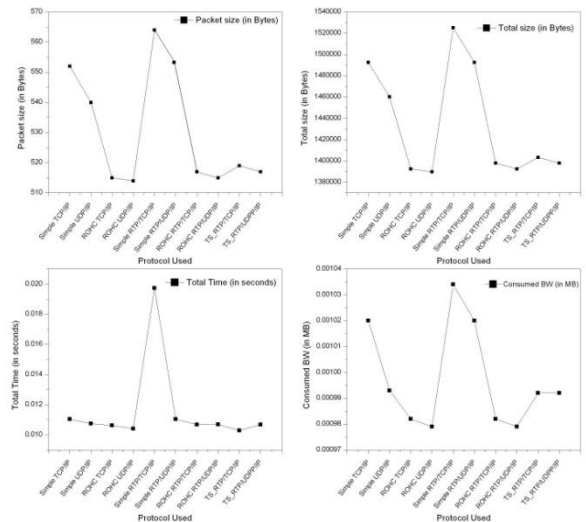


Fig. 2. Gateway Traffic Compression performance results

The Compression gain and bandwidth gain results are shown in figure 3 below:

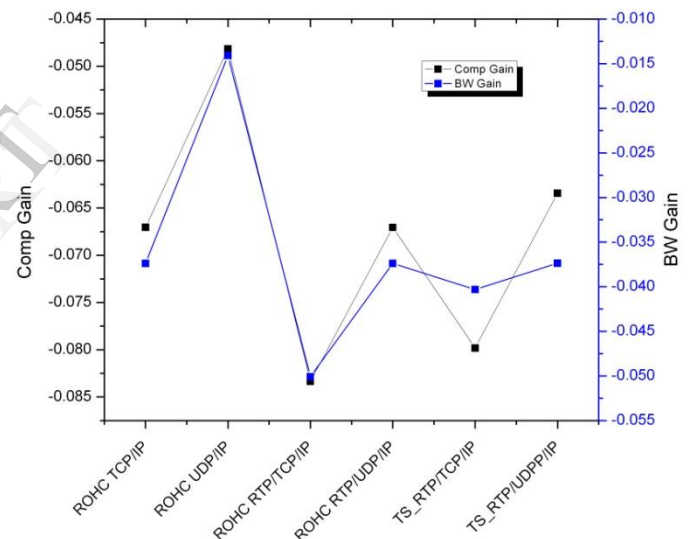


Fig. 3. Compression and Bandwidth Gain results

V. ADVANTAGES OF THE PROPOSED DESIGN

Many variants of header compression techniques are proposed where the entire header is compressed and sent over the wireless communication channel. But it is evident from the above survey that header compression techniques can be effectively used across mobile radio channels only if there exists a feedback channel from the de-compressor to the source. Hence there is a need for implementing a feedback technique explicitly with the header being complete compressed. In this paper we have designed a header compression algorithm where we retain the key fields of RTP header. It is obvious that the compression gain will be less when compared to the other schemes, but the overall performance of the multimedia traffic over an 802.11 ad-hoc

wireless network will be optimized in terms of rate synchronization, packet reordering and QoS.

RTP was designed by the IETF's Audio-Video Transport Working Group. RTP is commonly used in internet telephony for real time applications. RTP does not in itself guarantee real-time delivery of multimedia data since this is dependent on network characteristics. RTP combines its data transport with a control protocol (RTCP), which makes it possible to monitor data delivery for large multicast networks. Monitoring allows the receiver to detect if there is any packet loss and to compensate for any delay jitter. Both protocols work independently of the underlying transport layer and network layer protocols. Information in the RTP header tells the receiver how to reconstruct the data and describes how the codec bit streams are packetized.

RTP components include:

- Sequence Number – This is used to detect lost packets and to handle packet reordering.
- Payload Identification – This describes the specific media encoding so that it can be changed if it has to adapt to a variation in bandwidth.
- Frame Indication – This marks the beginning and end of each frame.
- Source Identification – This identifies the originator of the frame.
- Intramedia Synchronization – This uses timestamps to detect different delay jitter within a single stream and compensate for it.

RTCP components include:

- Quality of Service (QoS) feedback – This includes the numbers of lost packets, round-trip time, and jitter, so that the sources can adjust their data rates accordingly.
- Session Control – This uses the RTCP BYE packet to allow participants to indicate that they are leaving a session.
- Identification – This includes a participant's name, e-mail address, and telephone number for the information of other participants.
- Intermedia Synchronization – This enables the synchronization of separately transmitted audio and video streams.

Unlike other kinds of traffic, voice and video do not need the occasional dropped packets to be retransmitted (this is partly by the design of voice and video codec's, where occasional lost packets can be tolerated, also the playback would have moved on by the time of retransmitted packet arrives, so it would no longer be useful). This is quite different from the requirement for file transfer, for example, which is more flexible as far as delay tolerance is concerned, but where all packets must eventually arrive at the destination

RTP consists of data and a control part. The data part of RTP is a thin protocol providing support for applications with real-time properties such as continuous media (e.g., audio and

video), including timing reconstruction, loss detection, security and content identification. The control part RTCP provides support for real-time conferencing of quality-of-service feedback from receivers to the multicast group as well as support for the synchronization of different media streams.

Considering the features available in the RTP, it would rather result in optimizing the communication between the peers by retaining the key fields rather than compressing the complete header and explicitly defining a feedback from the de-compressor to the source.

VI. CONCLUSION

IEEE 802.11 wireless LAN has been widely deployed in various areas. The wireless LAN provides a robust data communication mechanism along with user mobility. Nodes of a wireless ad-hoc network can communicate with one other without relying on any pre-existing infrastructure. Due to the limited bandwidth of the wireless links as compared to wired systems, Header Compression on IP Packets is an attractive method for efficient usage of the frequency spectrum available. In this paper, we present a method to improve the performance in a peer-to-peer ad-hoc wireless network by implementing Header Compression on IP packets transmitted on a point to point link. This proposed header compression design not only reduces the packet size for real time applications but also monitors the communication between the peers for a session. Monitoring allows the receiver to detect if there is any packet loss and to compensate for any delay jitter. Information in the RTP key fields tells the receiver how to reconstruct the data and describes how the codec bit streams are packetized.

REFERENCES

- [1] Special Issue on Wireless Ad Hoc Networks, IEEE Journal on Selected Areas in Communications, Aug1999
- [2] Pravin Ghosekar, Girish Katkar and Pradip Ghorpade, "Mobile Ad Hoc Networking: Imperatives and Challenges," IJCA Special Issue on "Mobile Ad-hoc Networks" MANETs, 2010.
- [3] Giuseppe Anastasi and Luciano Lenzini, "QoS provided by the IEEE 802.11 wireless LAN to advanced data applications: a simulation analysis," Wireless Networks, vol. 6, May 2000, pp. 99-108.
- [4] F.H.P. Fitzek, S. Hendrata, P. Seeling, M. Reisslein, "Chapter in Wireless Internet – Header Compression Schemes for Wireless Internet Access,"CRC Press, July 2003.
- [5] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links, Request for Comments 1144," February 1990.
- [6] Casner S., and Jacobson V., "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, Request for Comments 2508," February 1999.
- [7] Jonsson L, Degermark M, Hannu H, and Svanbro K, "Robust Checksum-based header Compression (ROCCO)," IETF Internet Draft, January 2000.
- [8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "A Transport Protocol for Real-Time Applications, Request for Comments 3550," July 2003.
- [9] D. Dujovne, T. Turletti, F. Filali, "A Taxonomy of IEEE 802.11 Wireless Parameters and Open Source Measurement Tools,"- IEEE Communications Surveys & Tutorials, Vol. 12, January 2010.
- [10] A. Boukerche, "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks," Mobile Networks and Applications, Netherlands, 2004, pp. 333-342.