International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 236-240
Published Online November 2019 in IJEAST (http://www.ijeast.com)

EFFICIENT BANDWIDTH UTILIZATION FOR TCP-BASED ONLINE GAMES

Vijeeta Patil
Department of CSE
KLEIT, Hubballi, Karnataka, India

Vishwanath Patil
Department of ECE
DSCET, Bengaluru, Karnataka, India

***Abstract*-Massive multiplayer online games are becoming popular and prosperous rapidly. The players actions affect the network parameters, the game platform, and the overall perceived quality is highly relevant for the purposes of game design, as well as for the networking infrastructure and network support for games. There exists strong linear relationship between performance metrics in server side and the concurrent player number online. This issue becomes important in some scenarios where many flows of a game share the same path, as would happen between a proxy and the central server of a game. This paper studies the use of a traffic optimization technique named TCM (Tunneling, Compressing and Multiplexing) to reduce the bandwidth of MMORPGs. First, TCP/IP headers are compressed using standard algorithms that avoid sending repeated fields; next, a number of packets are blended into a bigger one and finally, they are sent using a tunnel. The obtained**
Bandwidth saving is about 40 to 60 percent. Packets per second are also significantly reduced.

Keywords – MMORPG, TCM, header compression, multiplexing

I. INTRODUCTION

Online games have become increasingly popular in recent years. Unlike the producing-selling business model of standalone games, MMOG (Massive Multiplayer Online Game) is a service business in which game-playing experience would be a critical factor for success. Since most MMOG available in market are based on some sorts of Client-Server architecture, the performance of MMOG game servers will significantly influence game-playing experience. Modelling system performance in MMOG, however, becomes an essential work for promoting customer satisfaction level [9].

The standard MMORPG infra-structure is a typical multi-tier Client/Server architecture (Fig.1). A proxy server farm communicates with all players. Proxy server acts as a dispatcher between clients and game servers, according to which server contains the client. A processing cycle usually starts with a client sending a message to proxy, and the proxy in turn dispatches the command to the corresponding game server. Game server performs the game logic, modifies the status of game world and sends back the updated player's status to all stakeholders based on AOI (Area of Interest) management.

Although MMORPGs are between the most popular online games, there are other genres that are also played by millions of users. As an example, First Person Shooters (FPSs) are also a consolidated group of games with some characteristics in common: the virtual scenario is shared by a few tens of players, who use fire weapons to kill the enemies or accomplish a mission. The weapon can be improved as the player earns money, depending on their fighting skills

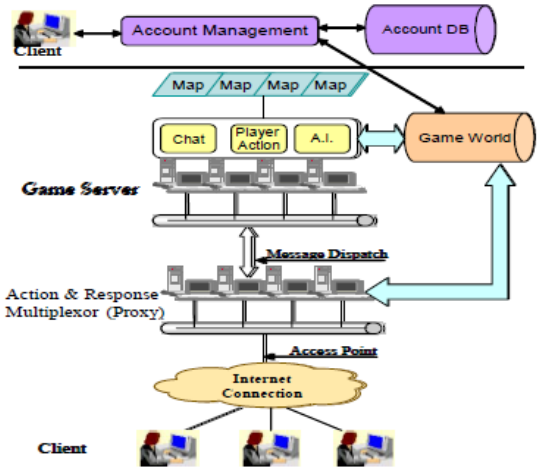

Figure 1. Standard MMORPG infra-structure

.

The use of UDP in FPSs means that there is no retransmission when a packet is lost. This fact has some implications: e.g. a shot can be lost, so players usually use machine-gun bursts so as to kill the enemies. Some games implement packet loss concealment algorithms in order to hide network impairments to the players

On the other hand, MMORPGs normally use TCP, which is reliable and avoids the loss of any information related to players actions. When a packet is lost, the protocol asks for a retransmission.

The two flags ACK and PSH are set for most of the packets. In near real-time usage both flags are used, although uncommon in normal TCP transmission [7].

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 236-240
Published Online November 2019 in IJEAST (http://www.ijeast.com)

These techniques were first developed for RTP streams over network paths that carry multiple Real-time Transport Protocol (RTP) streams in parallel between two endpoints, as in voice trunking [8]. The technique was adapted to UDP flows of FPS games share the same path. It is based on tunneling, compressing, and multiplexing many packets into a bigger one [3].
In this paper we try to find whether the technique is also suitable to compress TCP flows of MMORPGs, taking into account the special issues that appear when multiplexing is applied to a reliable protocol.

A] Scenarios where TCM can be applied

A proxy may receive the traffic of all the users of a zone (e.g. a town or a district), and forward it to the central server, so the aggregated traffic between the proxy and the server can be compressed and multiplexed, while adding small delays. This may provide some flexibility to the supporting infrastructure: when bandwidth gets scarce, traffic is multiplexed; and when the number of users diminishes, the traffic is sent in its native form, thus avoiding additional delays. (Fig 2.a)

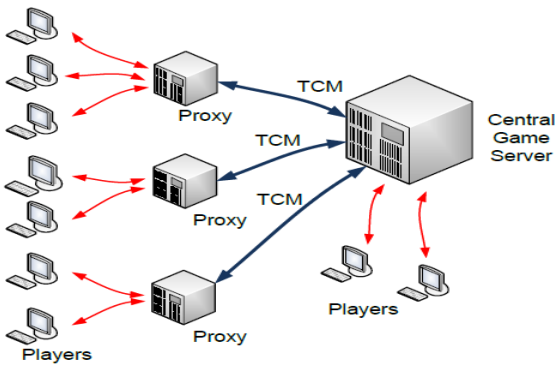

Figure 2.a

Other scenarios, as an Internet Cafe (Fig. 2.b), where a number of users simultaneously play the same game, can also be suitable for this technique. However, in this case the number of users may be significantly smaller than the one for the server-to-server scenario.

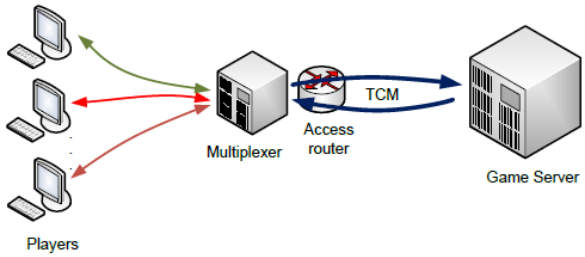

Figure 2.b

II. REVIEW of HEADER COMPRESSION AND MULTIPLEXING

P2P pub-sub is not feasible for MMOG's such as World of Warcraft and other similar games with current residential broadband. Message aggregation reduce bandwidth consumption, it improved message latency in most scenarios, and provided significant latency improvements for P2P pub-sub solutions [5].

A number of header compression have been defined and standardized. The first method for compressing TCP/IP headers was proposed by Van Jacobson [11]. It describes a method for compressing the headers of TCP/IP datagrams to improve performance over low speed serial links. Later, IPHC[16]The methods can be applied to of IPv6 base and extension headers, IPv4 headers, TCP and UDP headers, and encapsulated IPv6 and IPv4 headers. The compression algorithms are specifically designed to work well over links with nontrivial packet-loss rates. Later cRTP [10] was defined, describes a method for compressing the headers of IP/UDP/RTP datagrams to reduce overhead on low-speed serial links. In many cases, all three headers can be compressed to 2-4 bytes. Last proposed was ROHC, it is designed to operate efficiently and robustly over various link technologies with different characteristics.
Multiplexing methods were first designed for RTP flows, due to the existence of scenarios where a number of real-time flows may share the same path. We study the feasibility of this method when applied to (soft) real-time TCP flows. There are many policies that can be used to decide the number of native packets to be included in each multiplexed one. For instance, a fixed number of packets or a size limit could be used, but they would not take into account the added delays, which have a strong influence on QoS. So in this work we will consider two policies capable of controlling the delays, comparing their performance and behavior: first, *timeout* policy, in Fig. 3.a, sends a multiplexed packet if a native one arrives and the time since the last multiplexed packet departure is above a timeout *TO*. If there is only one packet, it will be sent in its native form, as the use of a tunnel would make it bigger. The other policy is named *period*, shown in Fig. 3.b, where a multiplexed packet is sent at the end of each interval or period *PE*. There are two exceptions: if there is no packet to multiplex, nothing will be sent; and if there is only one packet, it will be sent in its native form [4].

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 236-240
Published Online November 2019 in IJEAST (http://www.ijeast.com)

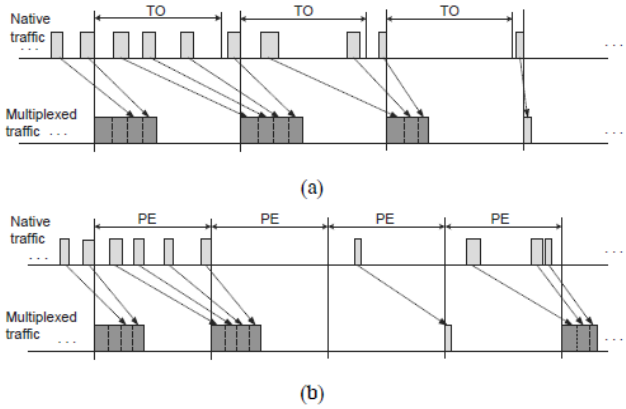

Figure 3.a) Timeout b) Periodic Policy

III. METHODOLOGY

It consists of three steps: first, a header compression algorithm is applied to the headers of the packets. Next, compressed packets are multiplexed using PPPMux and finally, the bundle is sent to the destination using PPP and L2TPv3 tunneling, which allows end-to-end delivery. the tunnel is only established from the multiplexer to the demultiplexer, where the packets are rebuilt exactly as they were generated by the application. This means that multiplexing is transparent to the game and to the server, so it can be independently applied for client-to-server and/or server-to-client traffic.

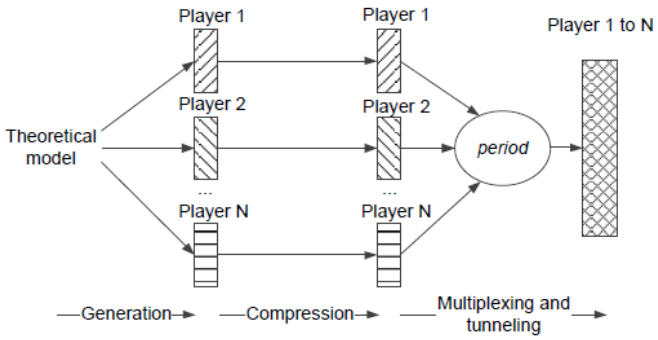

Figure. 4. Stages for TCM

A. Header Compression Algorithm

IPHC [10] and ROHC [6] are protocol capable of compressing TCP/IP headers. As the scenarios considered in the present work are wired networks with a very low packet loss rate, IPHC is considered more adequate for our proposal. Headers of typical UDP or TCP packets can be compressed down to 4-7 octets including the 2 octet UDP or TCP checksum. This compression method uses four packet types in addition to the IPv4 and IPv6 packet types.

FULL_HEADER - indicates a packet with an uncompressed header, including a CID and, if not a TCP packet, a generation. It establishes or refreshes the context for the packet stream identified by the CID.
COMPRESSED_NON_TCP - indicates a non-TCP packet with a compressed header. The compressed header consists of a CID identifying what context to use for decompression, a generation to detect an inconsistent context and the randomly changing fields of the header.
COMPRESSED_TCP - indicates a packet with a compressed TCP header, containing a CID, a flag octet identifying what fields have changed, and the changed fields encoded as the difference from the previous value.
COMPRESSED_TCP_NODELTA - indicates a packet with a compressed TCP header where all fields that are normally sent as the difference to the previous value are instead sent as-is. This packet type is only sent as the response to a header request from the decompressor. It must not be sent as the result of a retransmission.

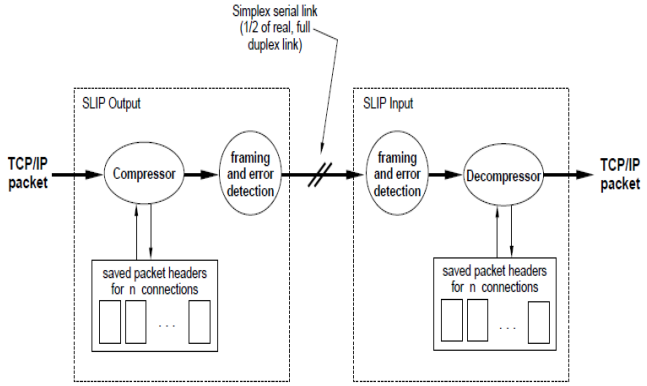

Figure. 5. Compression/decompression Model

Compressible TCP packets are looked up in an array of packet headers. If a matching connection is found, the incoming packet is compressed; the (uncompressed) packet header is copied into the array, and a packet of type COMPRESSED TCP is sent to the framer. If no match is found, the oldest entry in the array is discarded; the packet header is copied into that slot, and a packet of type UNCOMPRESSED TCP is sent to the framer.

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 236-240
Published Online November 2019 in IJEAST (http://www.ijeast.com)

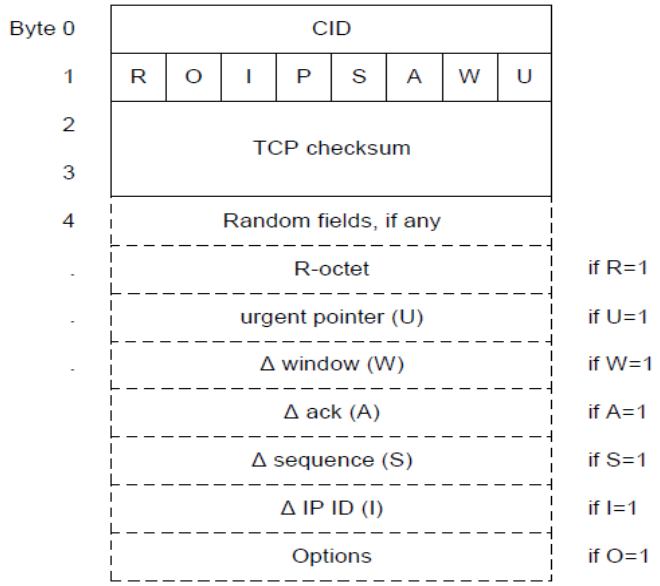

Figure. 5. Header of Compressed TCP Packet

- $E[P]$: The expected value of the TCP payload, which depends on the application.
- $E[k]$: The average number of native packets included into a multiplexed one.
- $E[RH]$: The expected value of the reduced header.

The expected size in bytes of native and multiplexed packets arrived in a period will respectively be:

$$bytes_{native} = E[k]\,(NH + E[P])$$

$$bytes_{mux} = CH + E[k]\,(MH + E[RH] + E[P])$$

As a consequence, the bandwidth saving *(BWS)* can be obtained as:

$$BWS = 1 - (bytes_{mux} / bytes_{mux})$$

B. Multiplexing and Tunneling

A policy based on a period is used, as shown in Fig. 7. All the packets that arrive during a period are multiplexed together, despite the flow they belong. A threshold of 1,350 bytes is set so as to avoid multiplexed packets bigger than the MTU.

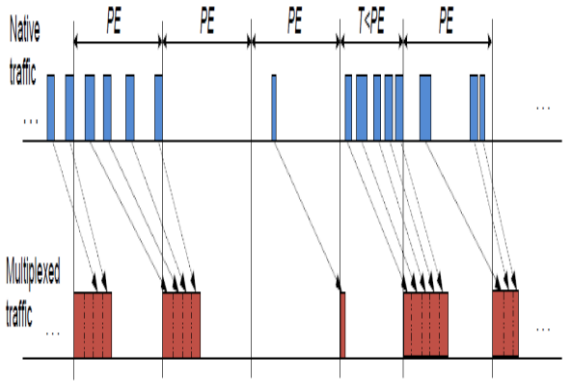

Figure. 6 Multiplexing Policy

IV. RESULTS

The bandwidth saving for different numbers of players, with a period ranging from 10 to 100 ms. We can observe that the curves present an asymptote around 60%, as expected. If the number of players is small, a period of 50 ms has to be used so as to obtain bandwidth savings above 50 %.

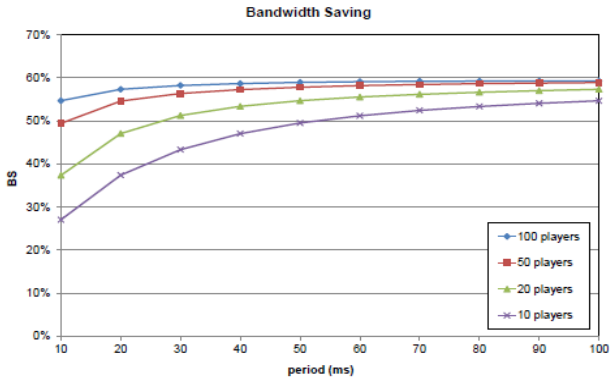

Figure. 7. Bandwidth Savings

Fig 8.the packets per second, which can be reduced from 900 to 10

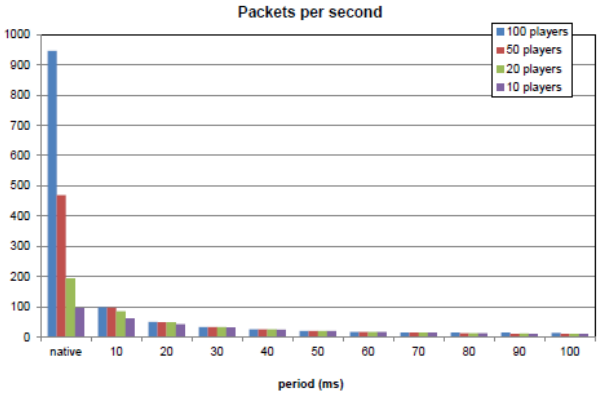

Figure. 8. Packets per Second

V. CONCLUSION

In scenarios where the flows of a number of players share the same path a multiplexing and header compression method has been applied to the traffic. The obtained savings are significant, and can be about 60 percent. An important reduction in the amount of packets per second can also be observed.

VI. REFERENCES

[1] J. Saldana, J. Fernandez Navajas, F. Pascual Blanco, July 13, 2012. "Tunneling Compressed Multiplexed Traffic Flows (TCMTF)", Transport Area Working Group-Internet-Draf.,

[2] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, J.I. Aznar, L. Casadesus, E. Viruete: (2011). "Comparative of Multiplexing Policies for Online Gaming in terms of QoS Parameters". IEEE Communications Letters, vol.15, no.10, pp.1132-1135

[3] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, J.I. Aznar, E. Viruete, L. Casadesus: (2011) First Person Shooters: Can a Smarter Network Save Bandwidth without Annoying the Players?. IEEE Communications Magazine, vol. 49, no.11, pp. 190-198

[4] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, J.I. Aznar, L. Casadesus, E. Viruete: (2011) Comparative of Multiplexing Policies for Online Gaming in terms of QoS Parameters. IEEE Communications Letters, vol.15, no.10, pp.1132-1135

[5] J.L. Miller, J. Crowcroft: (2010) The Near-Term Feasibility of P2P MMOGs. In: Proc. International Workshop on Network and Systems Support for Games (NetGames)

[6] K. Sandlund, G. Pelletier, L-E. Jonsson: (2010) RFC 5795. The Robust Header Compression (ROHC) Framework

[7] P. Svoboda, W. Karner, M. Rupp, M.: (2007)Traffic Analysis and Modeling for World of Warcraft. In: Proc. ICC, Urbana-Champaign, IL, USA

[8] B. Thompson, T. Koren, D. Wing. RFC 4170(2005): Tunneling Multiplexed Compressed RTP (TCRTP), November.

[9] G. Huang, M. Ye, L. Cheng,2004 Modeling System Performance in MMORPG. In: Globecom 2004 Workshop, pages 512-518. IEEE

[10] S. Casner V. Jacobson: RFC 2508,1999 Compressing IP/UDP/RTP Headers for Low-Speed Serial Links

[11] V. Jacobson: RFC 1144:1990 Compressing TCP/IP Headers for Low- Speed Serial Links

[12] Zhuo Jiang, Inwhee Joe,,Efficient bandwidth utilization and congestion control through network traffic analysis