INTERNET PROTOCOL IN THE SERVICE OF STREAMING

Pavol KOCAN, Ján MOCHNÁČ, Stanislav MARCHEVSKÝ
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel.: +421 55 602 4105, +421 55 602 2030,
e-mail: pavol.kocan@tuke.sk, jan.mochnac@tuke.sk

ABSTRACT

One way, how to deal with the packet losses for video transmitted over congested network is using connection-oriented, end-to-end reliable protocol. In this paper we focused on transmission control protocols, which fulfill the conditions for reliable delivery of information. TCP Tahoe and TCP Reno implementation supported by TCP are part of congestion avoidance mechanisms of the protocol suite. The main difference between them is mechanism of detection and reaction on the packet loss. Thus we have tried to evaluate their performance using Opnet simulation software.

Keywords: *multimedia streaming, Opnet, transmission protocols*

1. INTRODUCTION

With growing interest in digital video delivery also grows the interest in providing of sufficient video quality. There are several ways for satisfying quality demands. On the one hand, there can be used error concealment methods which try to approximate losses in video. Another way how to ensure adequate video quality is utilizing transfer protocol which is based on using acknowledgement for correctly received parts of data. In traditional Internet Protocol (IP) networks, network layer is responsible for lower-layer transmissions and Transmission Control Protocol (TCP) provides services between applications and IP. TCP is widely used for connection oriented transmissions with guaranteed degree of reliability. TCP detect potential problems in a network, e.g. congestion, load balancing and subsequently request retransmission of lost packet or rearranges order of the packets [1][7].

TCP Tahoe and Reno first described the mechanism for detection of packet losses and mechanism for reaction on such loss. Each of these mechanism reduce congestion window but each makes a decision about it on different assumption.

The main disadvantage of transmission protocols which offer reliable end-to-end delivery is adding of delay. Therefore it is necessary to make a decision about using these protocols with respect to used application and it is also necessary to make some performance evaluation to find out whether reliable delivery should not be replaced with different mechanism for preservation of video quality.

This paper is organized as follows. Second section presents the theoretical background of TCP protocols, Tahoe and Reno are compared, the third section presents obtained network simulations with video multimedia trafic. We finish this paper with short conclusion.

2. TCP

The primary purpose of Transmission Control Protocol (TCP) is to provide a reliable logical circuit or connection service between pairs of processes. TCP must guarantee the reliability by itself, which is done with a sequence number to each byte transmitted. A positive acknowledgment (ACK) is expected from the receiving TCP layer. If the ACK is not received within a timeout interval, the data is retransmitted. Because the data is transmitted in blocks (TCP segments), only the sequence number of the first data byte in the segment is sent to the destination host. The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, and to eliminate duplicate segments. TCP chopping the data into basic blocks or datagrams by grouping the bytes into TCP segments, which are passed to the IP layer for transmission to the destination. Application using another TCP facilities like flow control, push function, logical connections and full duplex [4].

The mechanism that indicates to the sender the number of bytes it can receive is also referred to as a window mechanism. It is send in the ACK in the form of the highest sequence number it can receive without problems. The congestion window in TCP determines the number of bytes that can be outstanding at any time. Its size is calculated by estimating how much congestion there is between the two places. The maximum segment size (MMS) is set up for the congestion window at the connection start-up. If all segments are received and the acknowledgments reach the sender on time, some constant is added to the window size by an Additive Increase /Multiplicative Decrease (AIMD) approach.

2.1. Tahoe TCP

The Tahoe TCP implementation added a number of new algorithms and refinements to earlier implementations. The new algorithms include Slow-Start, Congestion Avoidance and Fast Retransmit [5]. The refinements include a modification to the round-trip time estimator used to set retransmission timeout values. [2],[3]. It isn't very suitable for high band-width product links because of the waiting timeout. The problem with Tahoe is that it takes a complete timeout interval to detect a packet loss and in fact, in most implementations it takes even longer because of the coarse grain timeout.

2.2. Reno TCP

The Reno TCP modified the Fast Retransmit operation to include Fast Recovery to prevent the communication path ("pipe") from going empty after Fast Retransmit. Fast Recovery operates by assuming each dup ACK received represents a single packet having left the pipe. Thus, during Fast Recovery the TCP sender is able to make intelligent estimates of the amount of outstanding data. Instead of slow-starting, as is performed by a Tahoe TCP sender, the Reno sender uses additional incoming dup ACKs to clock subsequent outgoing packets. In Reno, the sender's usable window becomes where is the receiver's advertised window, is the sender's congestion window, and is maintained at until the number of dup ACKs reaches an initial threshold of dup ACKs, and thereafter tracks the number of duplicate ACKs.

Reno's Fast Recovery algorithm is optimized for the case when a single packet is dropped from a window of data. The Reno sender retransmits at most one dropped packet per round-trip time. Reno significantly improves upon the behavior of Tahoe TCP when a single packet is dropped from a window of data, but can suffer from performance problems when multiple packets are dropped from a window of data [6].

2.3. Packet size

An Ethernet LAN typically will have a maximum transmission unit (MTU) of 1500 bytes; however, this may be lowered by a router. In order to determine the effective MTU of your network, it is best to force some IP fragmentation; that is, forcing IP to split the data up. In the internet traffic packet sizes seem mostly bimodal at 40 bytes and 1500 bytes. It was observed that a strong mode is around 1300 bytes [8].

Some of the application altered their rate by changing the packet size, while some others changed the actual sending rate (the spacing between packets). [9]

TCP behavior can be described by the equation 1. Increasing the packet size is the one way how we can increase overal sending rate

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \quad (1)$$

where: s – packet size (ussually MTU value for the maximum transfer rate), T – sending rate, R – the round trim time, t_{RTO} – the TCP retransmit timeout, p – the loss event rate [10].

3. SIMULATIONS

This section describes some Opnet simulations that show us quality of services, especially the connection congestion window size depending up the transmission control protocol (see Fig. 1 and Fig. 2). We compare two TCP protocols Tahoe and Reno in TCP connection congestion window size.

There was created scenario with following parameters: buffer size at the data transmitter was set on 256 kB (and 1024 kB), transfer rate 11 Mbps, technique for multiple access was DSSS and the transmission power 1 mW. Simulation was for the server and 10 wireless fixed stations in small LAN network with dimensions 100x100 meters, where three clients received H.263 video stream and the other clients another application like e-mail, file transfer, voice communication over IP and web.

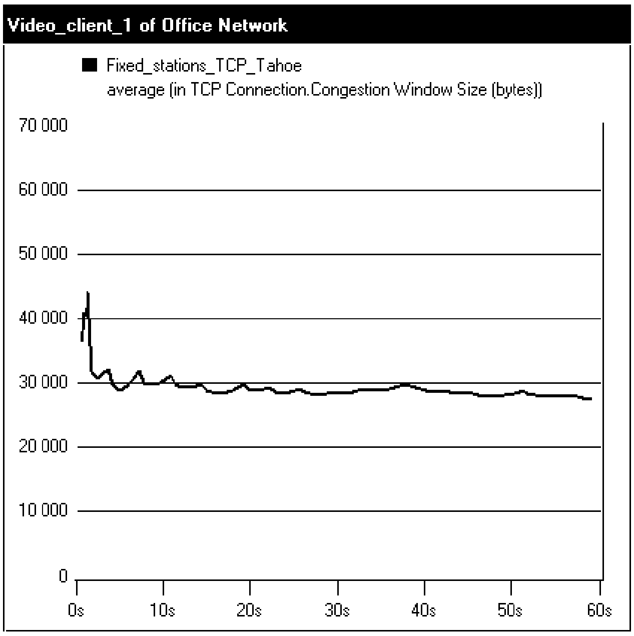

Fig. 1 Size of connection congestion window size in TCP Tahoe

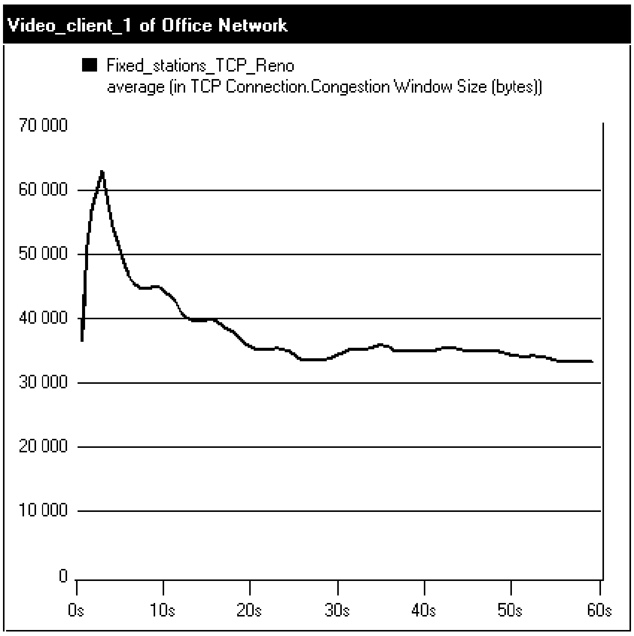

Fig. 2 Size of connection congestion window size in TCP Reno

Connection congestion window size represents the amount of not transferred data from the server to clients. Depending up the congestion during transmission, estimated window size reached maximum rate after starting the transfer and then is controlled by the AIMD (additive increase and multiplicative decrease) congestion control algorithm. As we can see in Figure 1 and Figure 2 newer TCP Reno have at the beginning higher estimated size of lost packets as the Tahoe protocol and then connection is much better prepared for data lost and the recovery mechanism for acknowledge too.

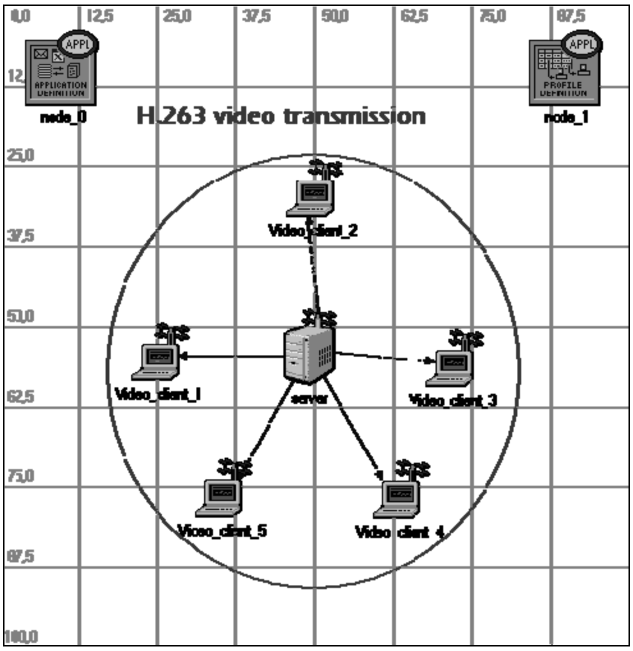

Fig. 3 Simulated Network

Another parameter was simulated and is illustrated on the next figures. The scenario parameters stay the same, only the traffic consists of H.263 video transmission between main server and tree clients (see Fig. 3).

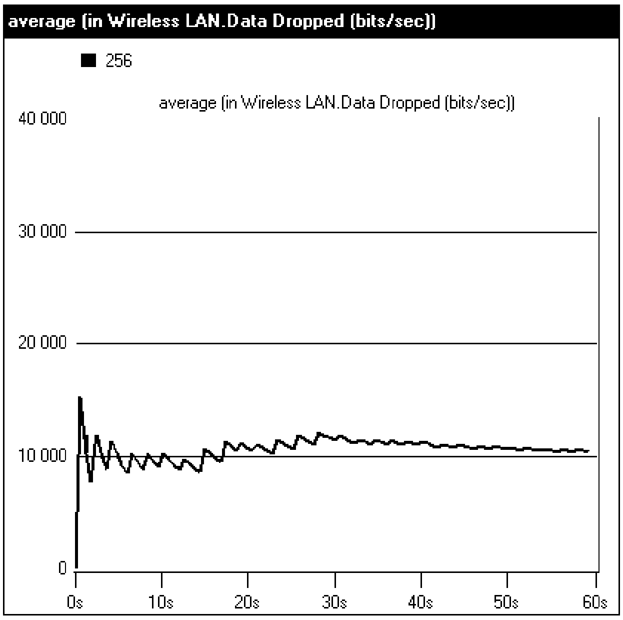

Fig. 4 Dropped data depending on packet size in bits

Overal load offen depends up on the packet size. In the wireless network with variable environment QoS parameters is more important to choose the right values to ensure the highest load if possible. There was set up two different values, 1024 and 256 bites for packet. Load for wireless network is higher in case of 1024 bites per one packet, because there are less acknowledgment transport on the network (Figure 4 and Figure 5). And again on the last figure dropped data depending on packet size (see Fig. 6).

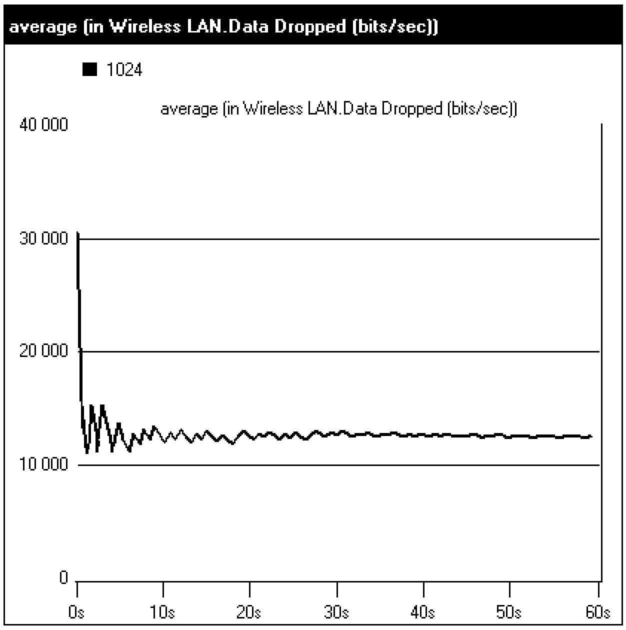

Fig. 5 Dropped data depending on packet size in bits

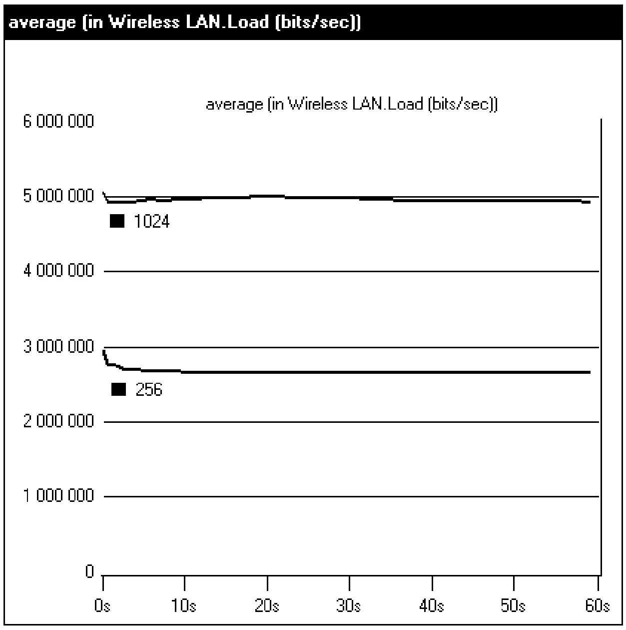

Fig. 6 Network load depending on packet size in bits

4. CONCLUSIONS

In this paper we focused on brief review on transmission control protocols, TCP Tahoe and TCP Reno and theirs parts of congestion avoidance mechanisms of the protocol suite that is so necessary to guarantee the quality of services for multimedia streaming. The main difference between them is mechanism of detection and reaction on the packet loss. We have tried to evaluate their performance in the Opnet network simulator that is sphere of interest of our research over the last years.

First we have compared two TCP protocols where Reno version is the better choice than Tahoe. The connection congestion window size during the transmission in Reno is higher that the Tahoe window size, so the transmission is less incoherent for losses.

Second we have change the packet size for transmission in wireless local area network. Although better load is for 1024 bytes packet size, it suffer higher dropped data at transmission beginnig. Small packet size at 256 bytes is better for transmission the video.

According to simulations results it's possible to improve the average load for wireless local area network with suitable packet size when the H.263 video is transmitted.

ACKNOWLEDGMENTS

This publication is the result of the project implementation Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020) supported by the Research & Development Operational Programme funded by the ERDF (50%). This work is part of a research project supported by VEGA No.1/0045/10 (50%).

REFERENCES

[1] RFC793 – Transmission Control Protocol, September 1981.

[2] VAN JACOBSON: Congestion Avoidance and Control, SIGCOMM Symposium on communications Architectures and Protocols, pp. 314–329, 1988.

[3] W. RICHARD STEVENS: TCP/IP Illustrated, Volume 1: The Protocols. Addison Wesley, 1994.

[4] PARZIALE, L. – BRITT, D. T. – DAVIS, Ch. – FORRESTER, J. – LIU, W. – MATTHEWS, C. – ROSSELOT, N.: TCP/IP Tutorial and Technical Overview, IBM, ISBN 0738494682, December 2006, pp. 150–151.

[5] BRUYERON, R. – HEMON, B. – ZHANG, L.: Experimentations with TCP Selective Acknowledgment, ACM SIGCOMM Computer Communication Review, Vol. 28, pp. 54–77, April 1988.

[6] FALL, K. – FLOYD, S.: Simulationbased Comparisons of Tahoe, Reno, and SACK TCP, Computer Communication Review, 1996, Vol. 26, pp. 5–21.

[7] http://www.opnet.com/solutions/

[8] SINHA, R. – PAPADOPOULOS, Ch. – HEIDEMANN, J.: Internet packet Size Distribution: Some Observations, University of Southern California, May 2007.

[9] WELZL, M.: Network congestion control, managing internet traffic, ISBN-13 978-0-470-02528-4, 2005, pp. 207–208.

[10] FLOYD, S. – HANDLEY, M. – PADHYE, J. – WIDMER, J.: Equation Based Congestion Control for Unicast Applications, March 2000.

Received November 8, 2009, accepted April 8, 2010

BIOGRAPHIES

Pavol Kocan was born on 12.3.1984. In 2007 he graduated (MSc.) at Technical University in Košice, Slovakia. He is working on his PhD. at the Department of Electronics and Multimedia Communications of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. His scientific research is focusing on congestion control for multimedia streams.

Ján Mochnáč was born on 28.06.1984. In 2007 he graduated (MSc.) at the Department of Electronics and Multimedia Telecommunications at Technical University in Košice. Since 2007 he is an internal PhD. student at the faculty. His research interests are in the fields of loss concealment methods for packet video.

Stanislav Marchevský received the M.Sc. in electrical engineering at the Faculty of Electrical Engineering, Czech Technical University in Prague, in 1976 and PhD. degree in radioelectronics at the Technical University of Košice in 1985. Currently he is a Professor of Electronics and Multimedia Communications Department of Faculty of Electrical Engineering and Informatics of Technical University of Košice. He teaching interests include switching theory, digital television technology, and satellite communications. His research interests include image nonlinear filtering, neural networks, genetic algorithms, and multiuser detection, spacetime communication, diversity communication over fading channel, and power and bandwidthefficient multiuser communications.