Simulation–Based Comparison of TCP and TCP–Friendly Protocols

Gábor Hosszú
Budapest University of Technology and Economics, Hungary

Dávid Tegze
Budapest University of Technology and Economics, Hungary

Ferenc Kovács
Budapest University of Technology and Economics, Hungary

INTRODUCTION

Internet streaming media changed the Web from a static medium into a multimedia platform, which supports audio and video content delivery. In our days streaming media turns into the standard way of global media broadcasting and distribution. The low costs, worldwide accessibility, and technical simplicity of this telecommunication way make media streams very attractive for content providers.

Streaming works by cutting the compressed media content into packets, which are sent to the receiver. Packets are reassembled and decompressed on the receiver side into a format that can be played by the user. To achieve smooth playback, packets are buffered on the receiver side. However, in case of a network congestion, the stream of packets slows down, and the player application runs out of data, which results in poor playback quality.

This article presents the comparison of different transport level congestion control schemes, including variants of the TCP. The protocol mechanisms, implemented in various protocols, are hard to investigate in a uniform manner; therefore, the simulator *SimCast* (*Sim*ulator for multi*Cast*) is developed for traffic analysis of the unicast and multicast streams. In this article the TCP and other transport protocol mechanisms will be compared using the SimCast simulator (Orosz & Tegze, 2001). The simulated results are presented through examples.

Due to spreading of traffic lacking end-to-end congestion control, congestion collapse may arise in the Internet (Floyd & Fall, 1999). This form of congestion collapse is caused by congested links that are sending packets to be dropped only later in the network. The essential factor behind this form of congestion collapse is the absence of end-to-end feedback. On the one hand an *unresponsive flow* fails to reduce its offered load at a router in response to an increased packet drop rate, and on the other hand a *disproportionate-bandwidth flow* uses considerably more bandwidth than other flows in time of congestion. In order to achieve accurate multicast traffic simulation—being not so TCP-friendly yet—the effects of the flow control of the TCP protocol should be determined (Postel, 1981). However, there are many different kinds of TCP and other unicast transport protocol implementations with various flow control mechanisms, which make this investigation rather difficult (He, Vicat-Blanc Primet, & Welzl, 2005).

Up to now a lot of comparisons have been done. For example, Wang et al. (2001) reviewed the TCP-friendly congestion control schemes in the Internet. They differentiated two groups of the TCP-friendly congestion control algorithms as follows: (1) *end-to-end* and (2) *hop-by-hop* congestion control mechanisms. The end-to-end mechanisms are grouped into (1) AIMD-based schemes (AIMD: additive increase multiplicative decrease) with the window- and rate-adaptation schemes, (2) modeling-based schemes, including equation-based congestion control schemes and the so-called model-based congestion schemes, and (3) a combination of AIMD-based and modeling-based mechanism. Wang's classification is mostly used in our discussion, too.

In this article various TCP congestion control mechanisms as well as congestion control mechanisms for media streams are reviewed. Then a novel simulator for transport protocols is described and the various

simulation results summarized. Lastly, conclusions are drawn and work to be done identified.

OVERVIEW OF THE TCP CONGESTION CONTROL

The Basic Control Mechanisms

The framework of the TCP congestion control is the use of a *sliding window*. Its main concept is that the sender can only send a limited number of unacknowledged segments to the receiver (Van Jacobson, 1988). The number of segments to be sent without receiving acknowledgement is determined by the *congestion window* (*Cwnd*). The *Cwnd* is given in bytes, which is the total length of the segments that belong to the congestion window (Floyd, 2001).

The basis of TCP congestion control is based on *additive increase multiplicative decrease (AIMD)*, halving the *Cwnd* for every window containing a packet loss, and increasing the *Cwnd* by roughly one segment size per *round trip time* (RTT) otherwise.

The *retransmit timers* are of fundamental importance in highly congested systems, which have exponential back off of the retransmit timer when a retransmitted packet itself is dropped.

The *slow-start* mechanism is for initial probing available bandwidth, instead of initially sending it at a high rate that might not be supported by the network (Stevens, 1997). At the beginning of the *slow-start* state the *Cwnd* equals one segment size. During *slow-start* the *Cwnd* is increased with a squared function in time.

ACK-clocking is the mechanism that uses the arrival of acknowledgements at the sender to clock out the transmission of new data.

Congestion Avoidance

The TCP sender could enter this state from the state *slow start*, if the *Cwnd* reaches the value of the *target window* (*Twnd*). In state *congestion avoidance* the increase of the *Cwnd* in response to a received ACK is:

$$\Delta Cwnd = \frac{B^2}{Cwnd}, \tag{1}$$

where *B* is the size of one segment in bytes. In the case of timeout the TCP goes to the *slow start* state.

Fast Retransmit—Fast Recovery

This method uses *repeated ACK*s to detect packet loss. After receiving three *repeated ACK*s, the sender retransmits the packet determined by the *SeqNum* (sequence number) of the ACK immediately and halves the *Cwnd*. After this the sender enters state *fast recovery*. At this point it increases the *Cwnd* with three segments; Then it increases with one segment in the case of arrival of additional repeated ACKs. Using this method a lot of unnecessary retransmissions can be avoided; it is effective in the case of sequential errors. Applying this method, better network utilization and throughput can be reached, since the receiver does not need to wait for the *retransmission timeout*. The sender leaves *Fast Retransmit* when it receives a useful ACK, or when a timeout occurs.

Selective Acknowledging (SACK)

This method is efficient in the case of multiple packet losses (Mathis, Mahdavi, Floyd, & Romanow, 1996). The receiver reports the segments that were received to the sender. In such a way the sender retransmits the absent segments, only.

CONGESTION CONTROL OF MEDIA STREAMS

TCP-friendly rate control (TFRC) is proposed for equation-based congestion control that explicitly adjusts the sending rate as a function of the measured rate of loss events (Handley et al., 2003). The TFRC is a receiver-based congestion control mechanism, with calculation of the loss event rate performed in the data receiver rather than in the data sender. This is appropriate for an application where the sender is a large server handling many concurrent connections. Therefore this is suitable as a building block for multicast congestion control. The TFRC is not a complete protocol; it is a congestion control mechanism only. It could be implemented in a transport protocol like *Real-time transport protocol* (Schulzrinne et al., 1996) or in an application incorporating end-to-end congestion control at the application level.

Related Content

Perceptual Semantics
Andrea Cavallaroand Stefan Winkler (2006). *Digital Multimedia Perception and Design (pp. 1-20).*
www.irma-international.org/chapter/perceptual-semantics/8420

Generating Personalized Explanations for Recommender Systems Using a Knowledge Base
Yuhao Chen, Shi-Jun Luo, Hyoil Han, Jun Miyazakiand Alfrin Letus Saldanha (2021). *International Journal of Multimedia Data Engineering and Management (pp. 20-37).*
www.irma-international.org/article/generating-personalized-explanations-for-recommender-systems-using-a-knowledge-base/301455

Application of the P2P Model for Adaptive Host Protection
Zoltán Czirkos (2009). *Encyclopedia of Multimedia Technology and Networking, Second Edition (pp. 54-60).*
www.irma-international.org/chapter/application-p2p-model-adaptive-host/17382

Audio Classification and Retrieval Using Wavelets and Gaussian Mixture Models
Ching-Hua Chuan (2013). *International Journal of Multimedia Data Engineering and Management (pp. 1-20).*
www.irma-international.org/article/audio-classification-and-retrieval-using-wavelets-and-gaussian-mixture-models/78745

Objective Measurement of Perceived QoS for Homogeneous MPEG-4 Video Content
Harilaos Koumaras, Drakoulis Martakosand Anastasios Kourtis (2005). *Encyclopedia of Multimedia Technology and Networking (pp. 770-777).*
www.irma-international.org/chapter/objective-measurement-perceived-qos-homogeneous/17327