The Transport-Level Requirements of the Internet-Based Streaming

Dávid Tegze
Budapest University of Technology and Economics, Hungary

Gábor Hosszú
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. Today, 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 that 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 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 (Postel, 1981). The protocol mechanisms, implemented in various protocols are hard to investigate in a uniform manner (Hosszú, 2005); therefore, the simulator SimCast (Simulator for multicast) is developed for traffic analysis of the unicast (one-to-one communication) and multicast (one-to-many communication) 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 on 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—because it is not so TCP-friendly yet—the effects of the flow control of the TCP protocol should be determined. 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).

Until now, a lot of comparisons have been done. For example, Wang et al. reviewed the TCP-friendly congestion control schemes on the Internet (Wang, Long, Cheng, & Zhang, 2001). 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 (a) additive increase multiplicative decrease (AIMD)-based schemes with the window- and rate-adaptation schemes, (b) modeling-based schemes, including equation based congestion control schemes and the so called model-based congestion schemes, and (c) a combination of AIMD-based and modeling-based mechanism. Wang's classification is mostly used in our discussion, too.

Yu (2001) proposes another important approach about the survey on TCP-friendly congestion control protocols for media streaming applications, in which several TCP-friendly congestion control protocols were discussed via a comparison of many important issues that determine the performance and *fairness* of a protocol.

It is an important advantage of the simulator *SimCast* that the latest TCP congestion control mechanisms are also implemented. In this way, the cooperation among different TCP protocol entities or various other transport level protocols can be examined (Shalunov, Dunn, Gu, Low, Rhee, Senger, Wydrowski, & Xu 2005).

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 future work is 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 (Jacobson, 1988). The number of segments to be sent without receiving acknowledgment 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*, 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 backoff 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 acknowledgments 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

The 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, Floyd, Padhye, &

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/transport-level-requirements-internet-based/13434

Related Content

Augmented Reality Assistance for Aging People: A Systematic Literature Review
Divya Udayan J., Raija Halonenand Aswani Kumar Cherukuri (2022). *International Journal of Information Systems and Social Change (pp. 1-17).*
www.irma-international.org/article/augmented-reality-assistance-for-aging-people/303604

Prediction of Nurses Allotment to Patient in Hospital through Game Theory
Satya Ranjan Dashand Rekha Sahu (2022). *Journal of Information Technology Research (pp. 1-15).*
www.irma-international.org/article/prediction-of-nurses-allotment-to-patient-in-hospital-through-game-theory/299916

Managing Strategic IT Investment Decisions From IT Investment Intensity to Effectiveness
Tzu-Chuan Chou, Robert G. Dysonand Philip L. Powell (2002). *Advanced Topics in Information Resources Management, Volume 1 (pp. 141-157).*
www.irma-international.org/chapter/managing-strategic-investment-decisions-investment/4583

E-Business Transaction in Web Integrated Network Environment
V. K. Murthyand E .V. Krishnamurthy (2005). *Encyclopedia of Information Science and Technology, First Edition (pp. 934-938).*
www.irma-international.org/chapter/business-transaction-web-integrated-network/14363

Context-Aware Framework for ERP
Farhad Daneshgar (2005). *Encyclopedia of Information Science and Technology, First Edition (pp. 569-572).*
www.irma-international.org/chapter/context-aware-framework-erp/14299