Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 4377–4395
Computer Networks
www.elsevier.com/locate/comnet

SSVP: A congestion control scheme for real-time video streaming

Panagiotis Papadimitriou *, Vassilis Tsaoussidis

Democritus University of Thrace, Electrical and Computer Engineering Department, 12 Vas. Sofias Street, Xanthi, Greece

Received 1 September 2006; received in revised form 1 May 2007; accepted 13 June 2007
Available online 29 June 2007

Responsible Editor: Buyurman Baykal

Abstract

In this paper, we present a new end-to-end protocol, namely *Scalable Streaming Video Protocol* (SSVP), which operates on top of UDP and is optimized for unicast video streaming applications. SSVP employs *Additive Increase Multiplicative Decrease* (AIMD)-based congestion control and adapts the sending rate by properly adjusting the inter-packet-gap (IPG). The smoothness-oriented modulation of AIMD parameters and IPG adjustments reduce the magnitude of AIMD oscillation and allow for smooth transmission patterns, while TCP-friendliness is maintained. Our experimental results demonstrate that SSVP eventually adapts to the vagaries of the network and achieves remarkable performance on real-time video delivery. In the event where awkward network conditions impair the perceptual video quality, we investigate the potential improvement via a layered adaptation mechanism that utilizes receiver buffering and adapts video quality along with long-term variations in the available bandwidth. The adaptation mechanism sends a new layer based on explicit criteria that consider both the available bandwidth and the amount of buffering at the receiver, preventing wasteful layer changes that have an adverse effect on user-perceived quality. Quantifying the interactions of SSVP with the specific adaptation scheme, we identify notable gains in terms of video delivery, especially in the presence of limited bandwidth.

Keywords: End-to-end protocols; Congestion control; Quality of Service; Video streaming

1. Introduction

Time-sensitive applications, such as streaming media, gain popularity and real-time data is expected to compose a considerable portion of the overall data traffic traversing the Internet. These applications generally prefer timeliness to reliability. Real-time video streaming, in particular, calls for strict requirements on end-to-end delay and delay variation. Furthermore, reliability parameters, such as packet loss and bit errors, usually compose an impairment factor, since they cause perceptible degradation on video quality. Unlike bulk-data transfers, video streaming seeks to achieve smooth playback quality rather than simply transmit at the highest attainable bandwidth.

* Corresponding author. Tel.: +30 25410 84382.
E-mail addresses: ppapadim@ee.duth.gr (P. Papadimitriou), vtsaousi@ee.duth.gr (V. Tsaoussidis).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2007.06.018

Such stringent requirements necessitate explicit management techniques in order to preserve the fundamental *Quality of Service* (QoS) guarantees for video traffic. In this context, *Internet Engineering Task Force* (IETF) attempted to facilitate true end-to-end QoS on IP networks by defining *Integrated* (*IntServ*) and *Differentiated Services* (*DiffServ*) models [2,19]. IntServ follows the signaled-QoS model, where the end-hosts signal their QoS need to the network, while DiffServ works on the provisioned-QoS model, where network elements are set up to service multiple classes of traffic with varying QoS requirements. However, both models are associated with high implementation costs and limited applicability; hence, they have not yet received wide appeal from the majority of users. Essentially, most end-users still rely on the best-effort services of the Internet which strives to meet the high demands of the emerging multimedia applications.

Today's Internet is governed by the rules of *Additive Increase Multiplicative Decrease* (AIMD) [5], which effectively contribute to its stability. Essentially, the goal of such algorithms is to prevent applications from either overloading or under-utilizing the available network resources. Although *Transmission Control Protocol* (TCP) provides reliable and efficient services for bulk-data transfers, several design issues render the protocol unsuitable for time-sensitive applications. More precisely, the process of probing for bandwidth and reacting to observed congestion causes oscillations to the achievable transmission rate. With TCP's increase-by-one and decrease-by-half control strategy, even an adaptive and scalable source coding scheme is not able to conceal the flow throughput variation. Furthermore, TCP occasionally introduces arbitrary delays, since it enforces reliability and in-order delivery. In response to standard TCP's limitations, several TCP protocol extensions [1,9] have emerged providing more efficient bandwidth utilization and sophisticated mechanisms for congestion control. *TCP-friendly* protocols, presented in [9,23,24], achieve smooth window adjustments, while they manage to compete fairly with TCP flows. In order to achieve smoothness, they use gentle backward adjustments upon congestion. In [21,26] we showed that this modification has a negative impact on responsiveness.

User Datagram Protocol (UDP) has been widely used instead of TCP by real-time applications, since it allows for transmission attempts at application rate and consequently, induces minimal fluctuations in the transmission rate. However, UDP poses a threat to network stability, as it lacks all basic mechanisms for flow/congestion control. Furthermore, as the success of the Internet primarily relies on self-regulated TCP, it is crucial to enforce compatible traffic regulations for non-TCP flows. In this context, Internetworking functionality evolves towards punishing free-transmitting protocols.

Congestion control algorithms are, therefore, necessary for multimedia applications in order to deal with the diverse and constantly changing conditions of the Internet. An overview of Internet's current congestion control paradigm reveals that routers play a relatively passive role: they merely indicate congestion through packet drops or *Explicit Congestion Notification* (ECN). It is the end-systems that perform the crucial role of responding appropriately to these congestion signals. Numerous video streaming applications have implemented their own congestion control mechanisms, usually on a case-by-case basis on top of UDP. However, implementing application-level congestion control is difficult and not part of most applications' core needs.

Time-sensitive application constraints and the limitations of existing congestion control schemes circumscribe a framework for potential improvements. We hereby identify distinct cases that motivate end-to-end protocol design for real-time traffic, especially if efficiency is considered on the basis of the application requirements:

- TCP's insistence on reliable delivery without timing considerations has an adverse effect on the performance of the system, especially for time-sensitive applications where data packets bear information with a limited useful lifetime.
- Multiplicative decrease with a factor of 1/2 (e.g. TCP, Rate Adaptation Protocol [16]) causes transmission gaps that hurt the performance of real-time applications, which experience jitter and degraded throughput.
- Slow-responding TCP-friendly protocols [23,24] yield sufficient performance in stationary environments. However, responsiveness is critical for the Internet which operates in the transient than in the stationary regime.
- *TCP-Friendly Rate Control* (TFRC) [9] arguably enables smoother delivery than TCP; however, TFRC's throughput model is quite sensitive to parameters (e.g. packet loss rate, Round-Trip Time), which are often difficult to measure efficiently and to predict accurately. TFRC is