Computer Communications 33 (2010) 1578–1588

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

TCP-friendly congestion control for the fair streaming of scalable video

Sheng-Shuen Wang *, Hsu-Feng Hsiao

Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Available online 24 February 2010

Keywords:
Bandwidth estimation
Congestion control
Packet train
Scalable video coding
TCP-friendly

ABSTRACT

Dynamic bandwidth estimation serves as an important basis for performance optimization of real-time distributed multimedia applications. The objective of this paper is to develop a TCP-friendly and fair congestion control algorithm which regulates the sending rate robustly by inferring the end-to-end available bandwidth. In addition to network stability, we also consider the characteristics of streaming applications, such as the bandwidth resolution in scalable video coding (SVC) which can achieve fine granularity of scalability at bit level to fit the time-vary heterogeneous networks. The congestion control algorithm is mainly composed of two phases: start phase and transmission phase to better utilize the network resource by subscribing SVC layers. In the start phase, we analyze the relationship between the one-way delay and the dispersion of packet trains, and then propose an available bandwidth inference algorithm which makes use of these two features without requiring administrative access to the intermediate routers along the network path. Instead of either binary search or fixed-rate bandwidth adjustment of the probing data as proposed in literature, a *top-down* approach is proposed to infer the initial available bandwidth robustly and much more efficiently. After acquiring the initial available bandwidth, the missions of the transmission phase include the adaptation of the sending rate fairly by progressive probing and also the accommodation of the network resource to TCP flows.

In case of the unavoidable network congestion, we unsubscribe scalable video layers according to the packet loss rate instead of only dropping one layer at a time to rapidly accommodate the streaming service to the channels and also to avoid persecuting the other flows at the same bottleneck. In addition, the probing packets for the estimation of the available bandwidth are encapsulated with RTP/RTCP. The simulations show that the proposed congestion control algorithm for real-time applications fairly utilizes network bandwidth without hampering the performance of the existing TCP applications.

1. Introduction

There has been explosive growth of emerging audio and video streaming applications recently. Many applications of multimedia communication over IP network, such as VoIP, multimedia on demand, IPTV, and video blog, have been integrated into our daily life rapidly. However, efficient delivery of media streams over the Internet is confronted with many challenges. Congestion control plays an important role to avoid congestive collapse by attempting to avoid oversubscription of any network resource of the intermediate nodes in the growing demand of those multimedia services over the Internet. TCP is a connection-oriented unicast protocol that offers reliable data transfer as well as flow and congestion control. However, multimedia streaming protocols have stricter requirement of transfer latency rather than reliable delivery. Many TCP-friendly congestion control protocols which are usually built upon UDP (User Datagram Protocol) with some specified congestion control algorithm are being developed so that the multimedia flows can behave fairly with respect to the coexistent TCP flows that dominate the network traffic so as to avoid the starvation of TCP traffic and to prevent the network from congestive collapse. In addition, the RTT unfairness introduced by the AIMD scheme of TCP leads to unequal bandwidth distribution among the competing flows with different round-trip time under the same congested links also should be taken into consideration. However, most of the existing streaming protocols have no consideration for the fairness and the characteristics of video streams which might affect the quality of streaming services significantly.

According to the general rate distortion curve shown in Fig. 1, the larger bit rate acquired, the better quality displayed. Some works focus on optimizing rate-distortion by multipath routing [28], or by active queue management and receiver feedback [27]. As to various video coding technologies, bit-stream scalability is a desirable feature for many multimedia applications so that graceful adaptation of transmission requirements can be achieved. Scalable Video Coding (SVC) [13], as an amendment G to the H.264/MPEG-4 AVC standard created by Joint Video Team (JVT), intends to encode a video sequence once and the encoded bit stream is able

* Corresponding author.
E-mail addresses: sswang@cs.nctu.edu.tw (S.-S. Wang), hillhsiao@cs.nctu.edu.tw (H.-F. Hsiao).

0140-3664/$ - see front matter Crown Copyright © 2010 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.02.022

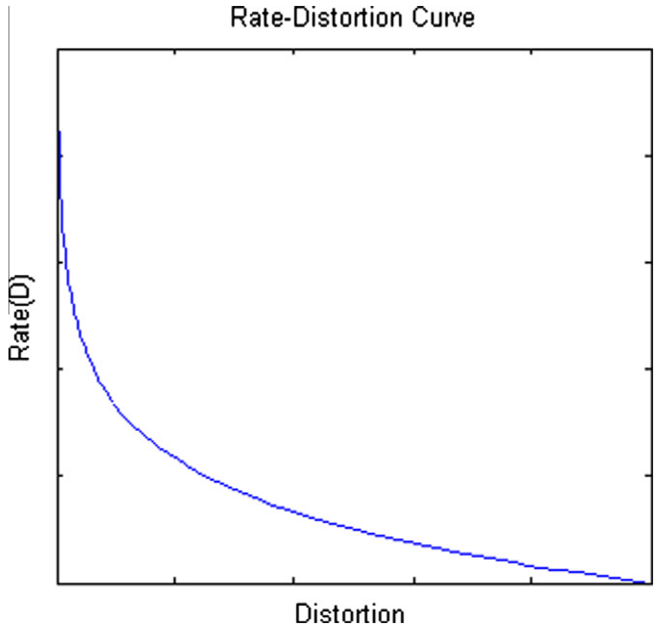

Fig. 1. Rate-distortion relationship.

to allow a diversity of different receivers to acquire and to decode a subset of the encoded bit stream without the need for transcoding. Scalable video coding enables not only efficient distribution of real-time multimedia streaming over heterogeneous networks but also a most promising solution for one-to-many congestion control over multicast networks. To fulfill above requirements, a scalable video bit stream contains a non-scalable base layer which is in compliance with the H.264/MPEG-4 AVC and one or more enhancement layers which may result from spatial, temporal or fidelity scalabilities of the scalable tools. The third octet in a Network Abstraction Layer (NAL) unit header records the layer identifications, including temporal_level (3bits), dependency_id (3bits), and quality_level (4bits). Thus, there can be $2^3 \times 2^3 \times 2^4$ video layers. Besides, the layers are subscribed layer by layer and the bit rate allocation between neighboring layers may vary significantly according to the streaming applications.

Due to the large number of possible scalable video layers and various bit rate allocation between layers, how to quickly converge to the time-varying available bandwidth without violating the existing flows under fairness competition becomes a critical factor in real-time video streaming. As to the information of available bandwidth, Multi-Router Traffic Grapher (MRTG) can use Simple Network Management Protocol (SNMP) to obtain the information from intermediate routers in the past. However, it is often difficult if not impossible due to various technical difficulties and privacy considerations or due to an insufficient level of measurement resolution [12]. One-way delay trend detection is utilized in Pathload [3] to measure the end-to-end available bandwidth by sending periodic packet trains. Since each packet train is used to determine only one decision that if the probed bit rate is greater or smaller than the available bandwidth, usually binary search is adopted to adapt the probing rate to the available bandwidth gradually. In contrast to acquiring initial available bandwidth over unicast networks, layered congestion control algorithm proposed in BIC [14], similar to Pathload, generates periodic burst packet trains from the upper layer over multicast network so that the probing periods of each receiver can be synchronized. Each receiver uses one-way delay trend detection to make the decision of joining one additional layer at a time and leaves a scalable video layer when packet loss rate exceeds a specified threshold. As a result, it is not suitable for receivers that might require joining or leaving several scalable video layers in a short time, due to dramatically fluctuant channels. In [11], a hierarchical sub-layer probing scheme which adopts coarse to fine layer partitioning to improve the efficiency of the probing interval was proposed. It might be helpful to reduce the number of probing periods when compared to BIC, but on the other hand, the probing packets might overshoot easily. Besides, network-layer multicast is still not widely deployed due to cost and management problems. Furthermore, it does not take TCP-friendly into account because only one probing rate is allowed for each synchronization point.

In this paper, we focus on TCP-friendly congestion control of fair end-to-end video streaming by inferring available bandwidth. We regulate the sending rate by using probing packets periodically such that a client running SVC applications can subscribe video layers gradually. In addition, we also consider the fairness property of both intra-protocol and inter-protocols, especially under different RTTs. Furthermore, RTP/RTCP [16] which relies on additional protocols to provide congestion control and to guarantee QoS to real-time multimedia steaming is integrated with the proposed congestion control algorithm. The remainder of this paper is organized as follows. In Section 2, the background and related works about bandwidth estimation and congestion control algorithm are presented. In Section 3, we describe our TCP-friendly congestion control algorithm with the consideration of RTT-fairness. In Section 4, the performance of our proposed algorithm is evaluated and the conclusion of this paper is given in Section 5.

2. Background and related works

In this section, we introduce the bandwidth estimation model based on the one-way delay (OWD) and the packet dispersion by sending probing packets. The selected TCP-friendly congestion control protocols in the literature will also be presented.

2.1. Bandwidth estimation

Among the results of emerging research in bandwidth estimation, link capacity and available bandwidth are of interest. The prior is constrained by the underlying transmission bandwidth. Given that packets are delivered from sender S to its receiver R through a fixed network path P, which consists of a sequence of store-and–forward links, the narrow link of a network path P is defined as the link with minimum capacity along the path. Assuming C_i is the link capacity of link i, and there are H hops in P, the capacity C of the narrow link is:

$$C = \min_{i=1\dots H} C_i. \tag{1}$$

The technology of packet pair [1] with two back-to-back packets of packet size S is usually used to measure the capacity by observing the dispersion δ ($\delta = S/C$) passing through narrow link if there is no background traffic.

On the other hand, available bandwidth depends on the traffic load of the path and it is typically a time-varying random variable. Assume $\lambda_i(t)$ is the traffic load of link i at time t, the available bandwidth $A_i(t, T)$ of link i is the average unused bandwidth over some time interval T as shown in (2).

$$A_i(t,T) = \frac{1}{T}\int_t^{T+t} (C_i - \lambda_i(t))\,dt. \tag{2}$$

The available bandwidth $A(t,T)$ of the tight link which is defined as the link with minimum available bandwidth along a path is:

$$A(t,T) = \min_{i=1\dots H} A_i(t,T). \tag{3}$$