Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications 28 (2005) 97–107
computer communications
www.elsevier.com/locate/comcom

Corruption and reordering robust TCP-friendly rate control

Rahul Chaudhary*, Lillykutty Jacob

Department of Computer Science, School of Computing, National University of Singapore, Lower Kent Ridge Road, 3 Science Drive2, Singapore, Singapore 117543

Received 19 June 2003; revised 22 April 2004; accepted 18 May 2004

Available online 11 June 2004

Abstract

TCP-Friendly Rate Control (TFRC) mechanism was introduced for regulated multimedia streaming over the Internet. TFRC unnecessarily reduces the application sending rate in wired network scenarios with packet reordering. TFRC also gives poor performance over wireless access networks. The poor performance of TFRC on paths that reorder packets and on wireless access networks can be attributed to its inability to differentiate packet losses due to congestion from other network conditions like packet reordering and corruption. We first examine the performance of TFRC over wireless network with UDP-lite that reduces packet drops over the wireless link. Next we propose few methods to make TFRC robust in packet reordering scenarios and evaluate the performance of TFRC with these schemes incorporated. Then we suggest a modification to TFRC from being loss-based to ECN-marking based. ECN-marking based TFRC performs rate regulation according to actual network congestion experienced and is robust to packet reordering and corruption in the network. We provide the results of our extensive experiments on a network testbed for various scenarios with packet corruption and/or packet reordering. We also observe the impact of ECN-based TFRC on concurrent TCP flows.

Keywords: Wireless; TFRC; ECN; Reordering; UDP-lite

1. Introduction

Delay sensitive real-time multimedia streaming applications prefer using the connection-less UDP that does not provide reliability, in-order delivery and congestion control features like TCP. Using an unregulated transport protocol for streaming is harmful for the health of the network as it can cause an undesirable network situation called congestion collapse [1]. For proper and efficient network functioning, rate regulation of such uncontrolled streaming is a must. Rate regulation for streaming applications using UDP can be achieved through TCP-Friendly Rate Control (TFRC) [2]. TFRC is an equation based congestion control mechanism for unicast UDP flows which aims at providing a smooth transition rate in the short time scale but in the larger time scale consuming no more bandwidth than a TCP flow under similar network conditions. It is a loss based rate regulation mechanism, which controls the sending rate for a UDP stream on the basis of packet losses that occur during the transmission of the stream. This mechanism is devised for wired networks where congestion is the only reason for packet losses during transmission.

TFRC emulates a TCP-like transmission behavior and thus treats reordered packets in a similar way as TCP does. TCP congestion control requires the sender to wait for a small number of duplicate ACKs (called reordering threshold) to arrive before deciding whether a missing packet is really lost [3]. TCP uses a reordering threshold of three. Any packet that is reordered beyond this specified reordering threshold value is treated as lost even if the packet arrives later. For any network path that reorders packets more than minimally the choice of three proves too aggressive in concluding loss [4]. Zhang et al. [4] and Blanton et al. [5] illustrate the impact of reordering on TCP performance and propose several alternatives to dynamically make TCP's fast retransmission algorithm more tolerant of the reordering. Measurement studies on packet reordering over the Internet show that packet reordering is not a rare event, 0.1–2.0% of packets experience reordering over some paths, and the prevalence of reordering varies across different network paths [6,7]. TFRC suffers from packet reordering because it emulates TCP and treats

* Corresponding author. Tel.: +65-945-20420; fax: +65-687-26198.
E-mail addresses: rahulchaudhary@yahoo.com (R. Chaudhary), jacobl@comp.nus.edu.sg (L. Jacob).

0140-3664/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.05.010

a packet reordered beyond the threshold as lost packet. Gharai et al. [8] discuss the effect of packet reordering on the observed throughput of their TFRC flows over the Internet. We propose several methods to make TFRC more robust in the face of packet reordering.

Also, the inability of TFRC in differentiating congestion losses from corruption losses attributes to poor TFRC performance in wireless access networks where packet losses can also occur due to error–prone wireless links. To have a unified rate control mechanism for both wired and wireless networks it is required that the mechanism performs rate control only on the basis of actual congestion situation experienced in the network and not be affected by situations like packet reordering or packet corruption. We hypothesize that making TFRC based on Explicit Congestion Notification (ECN) [9] marking is a good solution for efficient rate control in both situations of packet reordering as well as packet corruption. ECN-marking based TFRC performs sending rate control on the basis of actual network congestion information that is provided by ECN marking done in the packet headers by intermediate routers. The use of ECN marking by routers that are based on active queue management strategy like Random Early Detection (RED) [10] is very common these days.

In this paper, we first measure the performance improvement obtained through the combination of UDP-lite, which reduces packet drops over the wireless link, and TFRC. We then propose methods to make TFRC robust in packet reordering situation. Instead of a loss being detected by the arrival of three packets with higher sequence numbers than the lost packet, the requirement for three subsequent packets is made adaptive based on experienced packet reordering. We also propose an ECN based TFRC that accounts only congestion losses for sending rate adjustment. It neither accounts the losses occurring over the wireless hop nor the effect of packet reordering encountered in the Internet. In this rate controlling scheme intermediate routers detect incipient network congestion and inform the multimedia sources using ECN marking, which then calculate TCP friendly sending rate on the basis of ECN-marking probability and not packet loss probability as in TFRC.

We begin by providing some background on TFRC mechanism, UDP-lite and ECN. In Section 3 we describe several methods to make TFRC robust against packet reordering. Section 4 describes our ECN based TCP-Friendly Rate Control mechanism. Section 5 presents the test setup used. Section 6 describes various test experiments done by us, followed by the conclusion of our work in Section 7.

2. Background

In this section we briefly discuss the various mechanisms used in the paper

2.1. TCP-friendly rate control

TFRC is an equation based rate control mechanism for unicast UDP flows. In order to compete fairly with TCP, TFRC uses the TCP throughput equation to calculate its sending rate. The TCP throughput equation roughly describes TCP's sending rate as a function of the loss event rate, round-trip time, and packet size [2]:

$$X = \frac{s}{R\sqrt{\frac{2bp}{3}} + \mathrm{RTO}\left[3\sqrt{\frac{3bp}{8}}p(1+32p^2)\right]} \tag{1}$$

where X is the transmit rate, s is the packet size, R is the round-trip time, p is the loss event rate of the number of loss events as a fraction of the number of packets transmitted, RTO is the TCP retransmission time out value, and b is the number of packets acknowledged by a single ACK. The TFRC protocol relies on the following steps:

- The receiver measures the loss event rate p and feeds this information back to the sender.
- The sender receives these feedback messages and also uses them to measure the round-trip time R.
- Next the loss event rate and RTT are fed by the sender into the above throughput equation to compute the acceptable transmit rate.
- The sender then adjusts its transmit rate to match the calculated rate.

Calculating the loss event rate rather than simply taking the packet loss rate is an important part of TFRC. TFRC assumes that all packets contain a sequence number. The receiver maintains a data structure that keeps track of which packets have arrived and which are missing. The loss of a packet is detected by the arrival of at least three packets with a higher sequence number than the lost packet. The receiver maps the packet loss history into a loss event record, where a loss event is one or more packets lost in an RTT. This RTT is supplied periodically by the sender. If the starting packet sequence number of a loss event A is S_A and the starting packet sequence number of the next loss event is S_B, then the number of packets in loss interval A is given by $(S_B - S_A)$. Then the average loss interval I_{mean} is calculated using a filter that weights the n most recent loss event intervals. The loss event rate, p is simply: $p = 1/I_{mean}$.

2.2. UDP-lite

UDP-lite is a modification to the classical UDP protocol that can serve applications, which in a lossy network environment prefer to have partially damaged packets delivered rather than being dropped [11,12]. It provides the sending application with a checksum that has an optional partial coverage. Usage of this option causes a datagram to be separated into a part sensitive to errors and a part