

Rate-based Congestion Control for Tree-based Reliable Multicast

Van Ngo

Department of Computer Science,
York University,
Toronto, Ontario, Canada M3J 1P3
e-mail: ngo@cs.yorku.ca

Uyen Trang Nguyen

Department of Computer Science,
York University,
Toronto, Ontario, Canada M3J 1P3
e-mail: utn@cs.yorku.ca

Abstract

Although a few rate-based congestion control algorithms have been proposed for multicast, the effectiveness of these protocols on tree-based reliable multicast has not been evaluated. In addition, the tree-based recovery structure of reliable multicast could be exploited to enhance the performance and/or scalability of the congestion control scheme. In this paper, we extend TFMCC, a rate-based congestion control scheme designed for group-based (unreliable) multicast, to support tree-based reliable multicast. We first conducted experiments to investigate how network parameters may affect the sharing of bandwidth between TFMCC and TCP. Observations drawn from these experiments allowed us to avoid the situations in which multicast flows using TFMCC may be disadvantaged by TCP flows. Secondly we extend TFMCC to support tree-based reliable multicast. Specifically, we propose the use of the tree-based structure to calculate RTTs and to consolidate rate values at the designated receivers (DRs) in a scalable manner; and the calculation of retransmission rates of the DRs. Our simulation results show that the proposed protocol is effective, TCP-friendly and responsive to network conditions.

Keywords— congestion control, reliable multicast, TCP friendliness.

1 Introduction

Reliable multicast ensures that every data packet from a source is delivered to all receivers of a multicast group within a finite amount of time and free of errors. Missing packets are retransmitted either by the source or a subset of receivers (which are called Designated Receivers (DRs)). Reliable multicast protocols can be loosely classified into two categories: group-based and tree-based. Example group-based protocols are TRP [2], RMP [12] and SRM [4]. RMTP [9] and TMTP [15] are typical tree-based protocols. Tree-based loss recovery has been shown to be more scalable than group-based recovery [6], and is thus considered in this paper. Tree-based loss recovery can be enhanced with assistance from routers so that the routing tree can also be used as the repair tree. In this case, IP routers are modified to support subtree multicast. Router-assisted reliable multicast offers the following advantages over tree-based protocols such as RMTP and TMTP: less overhead for selecting and maintaining DRs; much better adaptability to topology changes; and the use of original routing paths for packet repairs, which are optimal paths.

Existing router-assisted reliable multicast protocols include LSM [8], Search Party [3], and AIM [7].

Congestion control for multicast can be window-based ([10], [11]), or rate-based ([13], [14]). Window-based schemes are most suitable for applications that need to multicast as much data as possible in a short period of time, because a sender can transmit as many packets as its window size allows. For this reason, window-based rate regulations have higher variations than rate-based. In addition, window-based control requires each receiver to maintain its own window for maximized throughput [5]. As the number of receivers grows, the overhead for maintaining the receivers' windows also increases.

Rate-based control, on the other hand, has lower rate variations than window-based. A sender transmits data at a calculated rate that is deemed optimal given the current network conditions. This makes rate-based schemes suitable for applications that requires relatively smooth transmission rates such as streaming media. In existing rate-based control protocols, at any point in time, the sender maintains only one transmission rate for the whole multicast group. Rate-based control is thus more scalable than window-based control.

Although a few rate-based congestion control algorithms have been proposed for multicast ([13], [14]), the effectiveness of these protocols on tree-based reliable multicast has not been evaluated. In addition, the tree-based recovery structure of the reliable multicast protocol could be exploited to enhance the performance and/or scalability of the congestion control scheme. In this paper, we extend TFMCC [14], a rate-based congestion control scheme designed for group-based (unreliable) multicast, to support tree-based reliable multicast. We first conducted experiments to investigate how network parameters may affect the sharing of bandwidth between TFMCC and TCP, the dominant Internet transport protocol. We observed that TFMCC is friendly to TCP. However in some cases TCP flows are more aggressive and take away bandwidth from TFMCC flows, and this negatively affects the throughput of the multicast session. This observation allows us to avoid the situations in which multicast flows using TFMCC may be disadvantaged by TCP flows. Secondly we extend TFMCC to support tree-based reliable multicast. Specifically, we propose the use of the tree-based structure to calculate RTTs and to consolidate rate values at the DRs for preventing the feedback implosion problem; and the calculation of retransmission rates of the DRs. We ran exten-

sive simulation-based experiments to evaluate the effectiveness and performance of the proposed algorithm TFMCC-RM (Reliable Multicast). Simulation results show that TFMCC-RM is effective, TCP-friendly and responsive to network conditions.

The paper is organized as follows. Section 2 presents simulation results of TFMCC interacting with TCP under different network parameters. The TFMCC-RM protocol is discussed in section 3 along with experimental results that show the effectiveness, responsiveness and TCP-friendliness of the protocol. Section 4 summarizes the paper.

2 TFMCC versus TCP

TFMCC [14] is a rate-based congestion protocol for group-based multicast in which each receiver uses an equation (given below) to calculate the optimal rate for the path from the sender to that receiver. The sender periodically consolidates the rate values from all the receivers and selects the lowest rate as its transmission rate. The sender employs a feedback suppression mechanism to minimize the implosion of feedback at the sender. The control equation used in TFMCC is as follows [14]:

$$T_{TCP} = \frac{s}{t_{RTT} \sqrt{\frac{2bp}{3}} + t_{retr} \min[1, 3\sqrt{\frac{3bp}{8}}] * p(1 + 32p^2)} \quad (1),$$

where T_{TCP} is data rate in bytes/sec; p is the packet loss rate; s is the packet size in bytes; b is the number of packets acknowledged by each TCP ACK; t_{retr} is retransmission timeout (e.g., $t_{retr} = 4 \times t_{RTT}$); and t_{RTT} is round trip time between the sender and the receiver computing the rate.

We carried out simulation-based experiments to investigate the interaction between TFMCC and TCP under different network parameters. We used the simulator provided by the authors of TFMCC [14], and the single-bottleneck configuration illustrated in Figure 1 that contains one TFMCC flow and ten TCP flows. The following setting was used in the experiments, unless otherwise stated: The capacity of link (2, 3) is 5Mbps; all the other links have a capacity of 90Mbps. The propagation delay of every link is 5ms. The buffer size of routers 2 and 3 is 120 packets. All routers use the drop-tail packet discard mechanism, the algorithm currently employed by Internet routers. Following are our observations with respect to the throughputs of and the interactions between the TFMCC and TCP flows.

1. When the buffer size is large, TCP and TFMCC compete fairly. However when the buffer size is small, the average TCP rates are noticeably higher than the TFMCC rate. Figures 2(a) and 2(b) shows the throughputs of the TFMCC flow and two of the TCP flows when the buffer size of router 2 is 50 packets and 200 packets respectively. An intuitive explanation is as follows. TCP flows send data in bursts. Since there are ten TCP flows, it is likely that there are always bursts of packets from at least one TCP

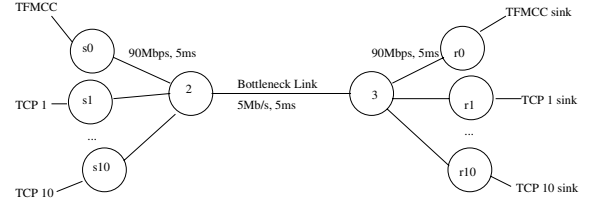


Figure 1: One TFMCC flow and ten TCP flows

flow occupying the buffer. Incoming TFMCC packets are queued at and thus dropped from the tail of the buffer, according to the drop-tail packet discard policy, resulting in high loss rate and thus low sending rate. When the buffer size is large, TFMCC can compete better with TCP since more TFMCC packets can be accommodated by a larger buffer.

2. When the buffer at the bottlenecked router (e.g., router 2 in Figure 1) is small, the bandwidth allocation is sensitive to small changes in the bandwidth of the bottlenecked link. This observation is illustrated by the following experiment in which the buffer size of router 2 is 50 packets, and the bandwidth of the bottlenecked link (2, 3) is set to 4.8Mbps and 4.9Mbps respectively. The throughput graphs given in Figures 2(c) and 2(d) show that the TFMCC rate is very low compared with those of the TCP flows when the bandwidth is 4.8 Mbps. However, when the bandwidth is 4.9Mbps, the TFMCC rate is noticeably higher than the average TCP rate.

3. The more competing TCP flows there are, the less fair it is to TFMCC, especially when the bandwidth of the bottlenecked link is narrow. The reason is the burstiness and aggressiveness of TCP flows, as discussed above. Figures 2(e) and 2(f) illustrate the results when there are three and ten competing TCP flows respectively.

4. In general, TFMCC flows get better throughputs with the RED (Random Early Detection) packet discard mechanism than with drop-tail, especially when the buffer at the bottlenecked router is small. This point is illustrated by comparing the graphs in Figures 3(a) and (b). In these experiments the buffer size of router 2 was set to 50 packets.

5. All the above experiments use TCP SACK (one ACK for every three data packets). We repeated all experiments with Tahoe TCP. All the above observations regarding the interactions between TFMCC and TCP are consistent for both Tahoe TCP and TCP SACK. The only difference we observed is that in many cases, the TFMCC flow gets a somewhat better throughput when competing with Tahoe TCP than with TCP SACK. The simulation results are not shown here due to space limitation.

3 TFMCC-RM: Protocol and Performance Evaluation

We propose the following extensions to TFMCC to support tree-based reliable multicast: rate feedback consolidation at DRs, calculation of RTTs, and calculation of DRs'

retransmission rates.

Rate feedback consolidation

Every receiver calculates the rate using equation (1) and sends the calculated rate upstream to the sender, as in TFMCC. However we exploit the tree-based structure to prevent feedback implosion. Every DR consolidates rate values from its children, selects the minimum value and sends that value to its parent DR. The parent DR in turn consolidates rate feedback from its own children, and so on, until the consolidated rate values reach the sender. Note that a consolidation timer is used to handle non-responsive branches.

Calculation of RTTs

We consider two cases: general reliable multicast in which the recovery tree may not be the same as the routing tree (e.g., RMTP, TMTP), and router-assisted reliable multicast in which both trees agree (e.g., LSM, Search Party). For general reliable multicast, the RTT between the sender and each receiver is calculated using the algorithm proposed by Basu and Golestani [1]. For router-assisted reliable multicast, the calculation of RTTs is simpler. Every receiver i maintains the RTT between its parent DR p and itself, $subRTT(i, p)$, using a timestamp included in each rate feedback message. Upon receiving the feedback message, the parent p echoes back to the child i the timestamp in addition to the value $RTT(p, s)$, the RTT between p and the sender s . The child then calculates $RTT(i, s)$, the RTT between the sender and itself, as follows:

$$RTT(i, s) = subRTT(i, p) + RTT(p, s)$$

The receiver then applies $RTT(i, s)$ to equation (1).

Calculation of DRs' retransmission rates

Every receiver calculates the rate using equation (1) and sends the calculated rate to its DR. The DR consolidates rate values from its children, and selects the minimum value. The DR then forwards this minimum rate value upstream to its parent, and also uses it as its retransmission rate.

Experimental results

We carried out experiments to evaluate the effectiveness, TCP-friendliness and responsiveness of the proposed TFMCC-RM protocol. We used RTMP [9] as the (general) tree-based reliable multicast protocol, and LSM [8] as the router-assisted reliable multicast protocol in all the experiments. We present here a set of experiments conducted on the balanced tree topology depicted in Figures 4(a) and 4(b). The shaded nodes denote the DRs. In these experiments, we used the following settings: The bottleneck links have a capacity of 6Mbps and a propagation delay of 5ms; all other links have a capacity of 10Mbps and a propagation delay of 10ms. All routers use the drop-tail packet discard mechanism. Five TCP flows compete with

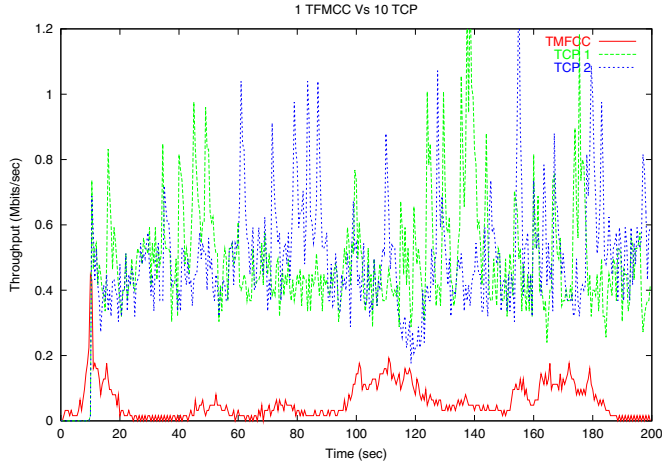
the TFMCC flow over the path from node 0 to node 7 (results for TCP SACK are similar). Figures 4(c) and 4(d) show the rates of the TFMCC flows and two of the TCP flows on the path from node 0 to node 7. The graphs illustrate the responsiveness and TCP-friendliness of TFMCC-RM when used in conjunction with LSM and RMTP.

4 Conclusions

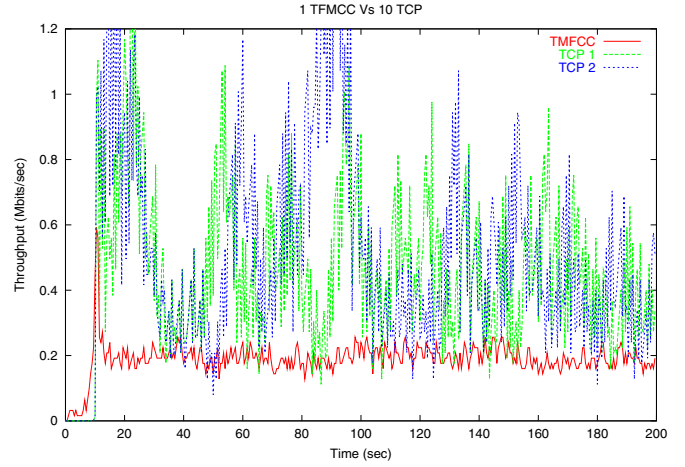
Rate-based congestion control for multicast is considered more scalable and offers lower rate variations than window-based control. In the paper we investigated the interactions between TFMCC and TCP flows under various network configurations. We then extended TFMCC to support tree-based reliable multicast. Specifically, we proposed the use of the tree-based structure to calculate RTTs and to consolidate rate values at the DRs for preventing the feedback implosion problem; and the calculation of retransmission rates of the DRs. Our simulation results showed that the proposed TFMCC-RM protocol is effective, TCP-friendly and responsive to network traffic demands.

References

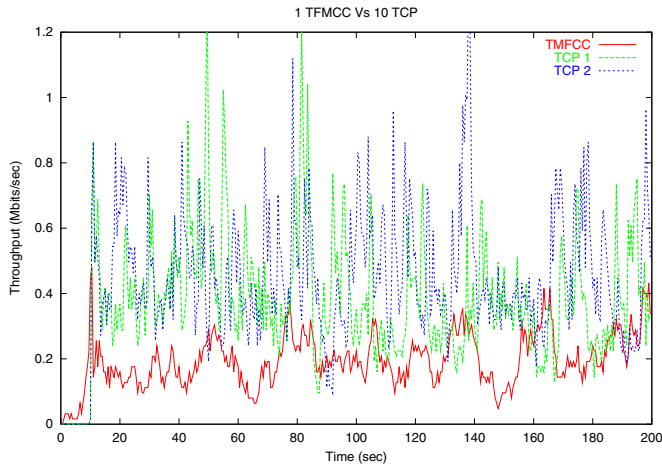
- [1] A. Basu and S. Golestani, "Estimation of Receiver Round Trip Times in Multicast Communications," *RMRG Meeting*, December 1998.
- [2] J-M Chang and N.F. Maxemchuk, "Reliable broadcast protocols," *ACM Transactions on Computer Systems (TOCS)*, vol. 3, issue 3, 1984.
- [3] A. Costello and S. McCanne, "Search Party: Using Randomcast for Reliable Multicast with Local Recovery," *Proc. IEEE INFOCOM*, 1999.
- [4] S. Floyd, V. Jacobson, C. Liu, S. McCanne and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, December 1997, Volume 5, Number 6, pp. 784-803.
- [5] S. J. Golestani and K. K. Sabtani, "Fundamental Observations on Multicast Congestion in the Internet," *Proc. IEEE INFOCOM*, 1999.
- [6] B.N. Levine and J.J. Garcia-Luna-Aceves, "A Comparison of Known Classes of Reliable Multicast Transport Protocols," *Proc. IEEE ICNP*, 1996.
- [7] B.N. Levine and J.J. Garcia-Luna-Aceves, "Improving Internet multicast with routing labels," *Proc. IEEE ICNP*, 1997.
- [8] C. Papadopoulos and G. Parulkar and G. Varghese, "An error control scheme for large-scale multicast applications," *Proc. IEEE INFOCOM*, 1998.
- [9] S. Paul, K. K. Sabnani, J. C. Lin and S. Bhattacharya, "Reliable multicast transport protocol (RMTP)," *Proc. IEEE INFOCOM*, 1996.
- [10] I. Rhee, N. Ballaguru and G. Rouskas, "MTCP: Scalable tcp-like congestion control for reliable multicast," *Proc. IEEE INFOCOM 99*, March 1999.
- [11] L. Rizzo, "A tcp-friendly single-rate multicast congestion control scheme," *Proc. ACM SIGCOMM*, 2000.
- [12] B. Whetten, T. Montgomery and S. Kaplan, "A high performance totally ordered multicast protocol," *Theory and Practice in Distributed System*, LNCS 938, Springer Verlag, 1994.
- [13] B. Whetten and J. Conlan, "A rate-based congestion control scheme for reliable multicast," Technical Report, Reliable Multicast Research Group working document, 1998.
- [14] J. Widmer and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification," Internet Draft, July 2003.
- [15] R. Yavatkar, J. Griffioen and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," *Proc. ACM Multimedia*, 1996.



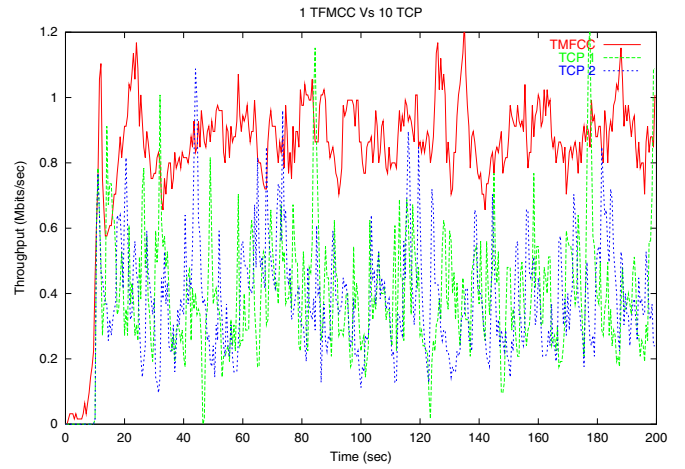
(a) Buffer size = 50 packets



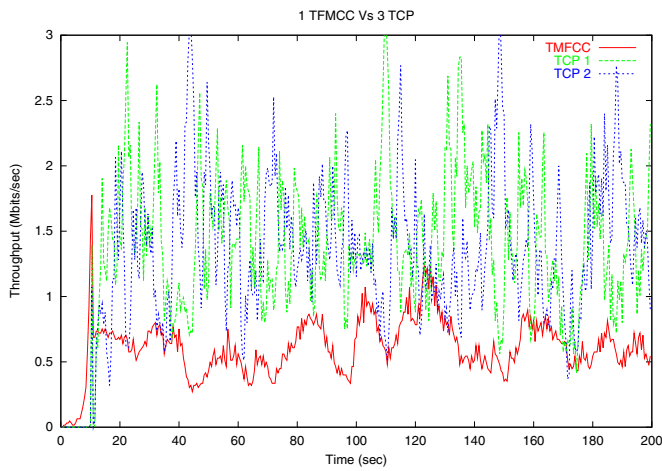
(b) Buffer size = 200 packets



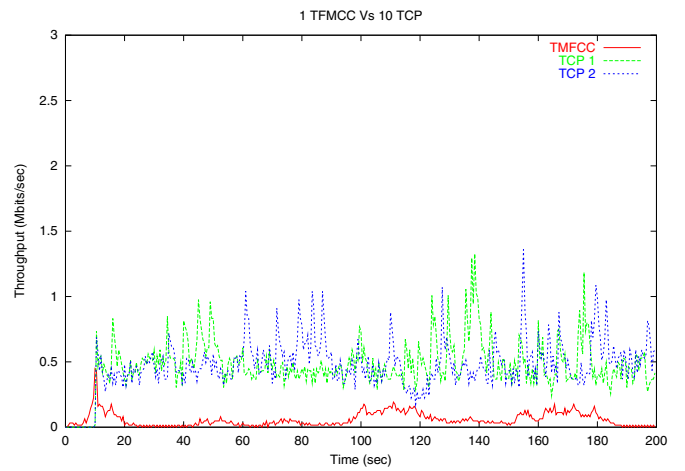
(c) Bottlenecked link = 4.8Mbps



(d) Bottlenecked link = 4.9Mbps

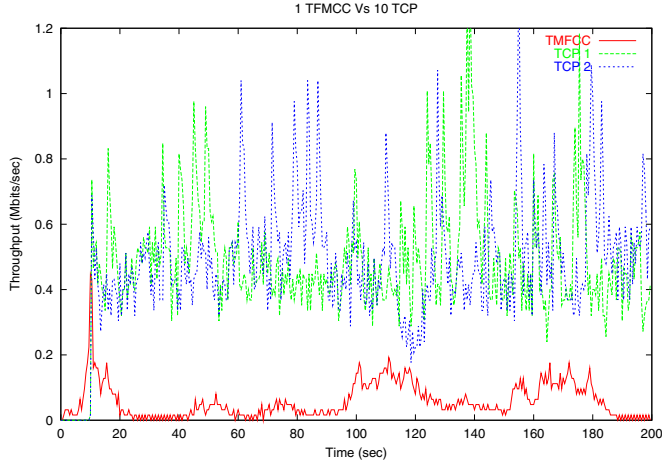


(e) 3 TCP flows

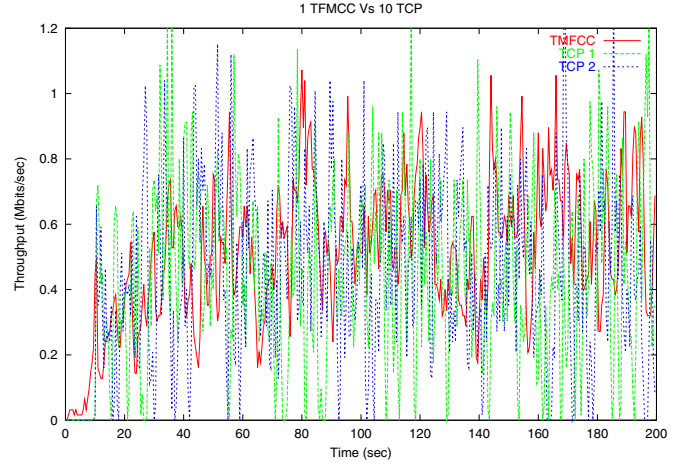


(f) 10 TCP flows

Figure 2: TFMCC vs TCP

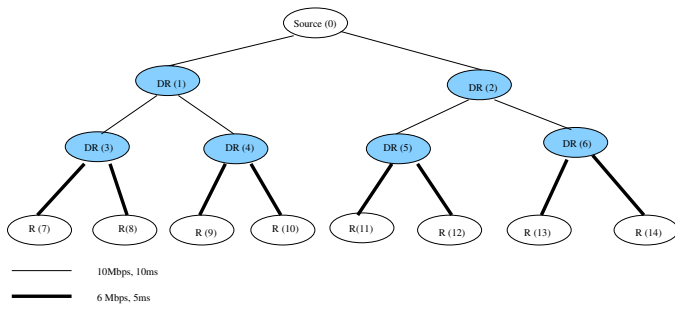


(a) Drop-tail

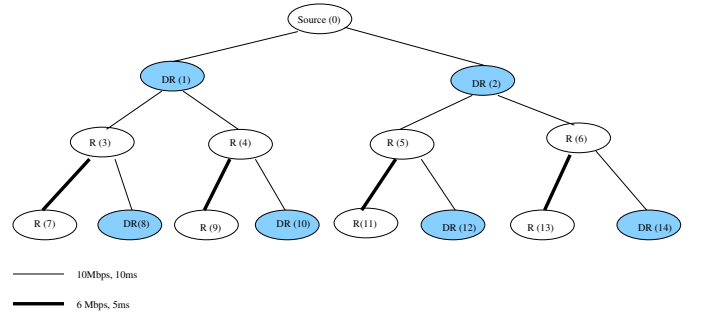


(b) RED

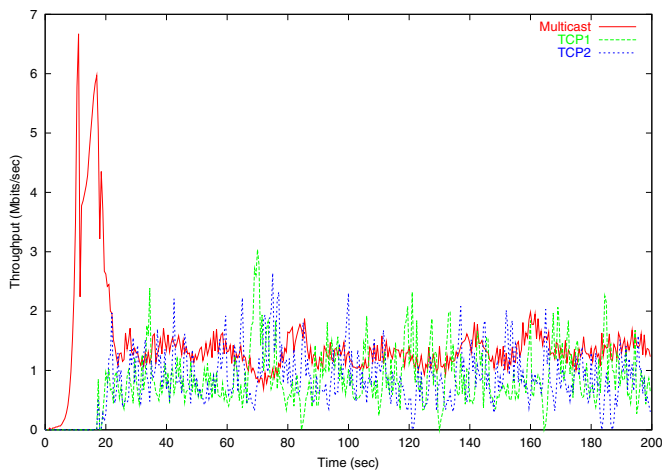
Figure 3: TFMCC vs TCP (continued)



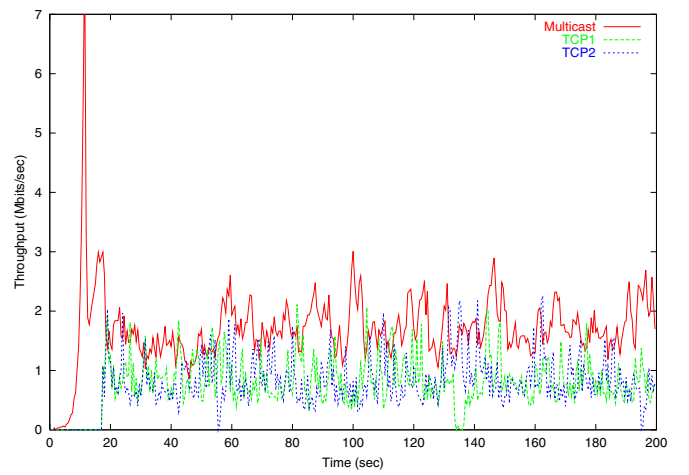
(a) Router-assisted reliable multicast LSM



(b) Reliable Multicast RTMP



(c) LSM



(d) RTMP

Figure 4: TFMCC-RM used with LSM and RTMP