[RLSSt03] *Ivica Rimac, Wolfram Liese, Jens Schmitt, Ralf Steinmetz;* **Equation-Based Approach to TCP-Compatible Multicast Congestion Control for Layered Transmission in Low-Multiplexing Environments;** 22nd IEEE International Performance Computing and Communications Conference (IPCCC 2003), Phönix/Arizona, April 2003, S. 469-473.

Equation-Based Approach to TCP-Compatible Multicast Congestion Control for Layered Transmission in Low-Multiplexing Environments

Ivica Rimac, Wolfram Liese, Jens Schmitt, and Ralf Steinmetz
KOM — Multimedia Communications
Department of Electrical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
Email: {Ivica.Rimac, Wolfram.Liese, Jens.Schmitt, Ralf.Steinmetz}@KOM.tu-darmstadt.de

Abstract

Multi-rate multicast has been proposed as a scalable solution to transmitting video over the Internet to receivers with heterogeneous and dynamic rate requirements. In this paper, the applicability of an equation-based mechanism to congestion control for a protocol which bases its join and leave actions on the calculation of the TCP response function is investigated. We focus on the rate calculation algorithm proposed in TFRC as it is currently a very promising and mature approach to calculating a TCP-compatible rate. By means of a network simulator and an adjusted TFRC protocol implementation we show that the TCP-compatible rate calculated with the algorithm as originally proposed tends to be biased when applied in environments with a low degree of statistical multiplexing. To improve the performance of the basic algorithm, a simple heuristic approach is proposed.

1. Introduction

With a progressing trend towards more continuous media distribution we are facing problems with the existing Internet, where end systems are expected to adopt the "social" rules and be cooperative by reacting to congestion signals and adapting their transmission rates properly and promptly. TCP, which provides appropriate mechanisms to meet the above requirements and prevents the Internet from congestion collapse, is unfortunately not applicable to multicast streaming. These applications commonly rely on UDP as the underlying transport protocol and do not integrate TCP-compatible congestion control mechanisms. In the context of the current best-effort Internet, this leads to highly unfair situations and in the worst case will lead to starvation of TCP traffic or even congestion collapse. As a consequence, since TCP is the predominant transport protocol and the fairness definitions for multicast are still subject to research, TCP-compatibility of multicast flows is a valid fairness criterion in today's Internet.

In the rest of the paper, we will focus on layered schemes but the results obtained are transferable to non-hierarchical encoded data as well. The primary goal and main contribution of this work is the investigation of the applicability of an equation-based approach to TCP-compatible congestion control for layered transmission. In Section 2 we provide background information and an overview of some related work in the area of TCP-compatible congestion control. The underlying TCP throughput model and the limitations of this approach in an environment with a low degree of statistical multiplexing are described. In Section 3 we present our prototype implementation and validate the behavior of the basic protocol by means of simulations. To overcome the limitations observed, we propose a simple heuristic but effective approach in Section 4. Finally, in Section 5 we conclude the paper with an outlook.

2. Background and related work

There has been increasing interest in providing TCP-compatible solutions to congestion control for unicast as well as multicast flows. For an excellent survey of TCP-friendly congestion control protocols the reader is referred to [12]. The notion of a TCP-compatible flow refers to a flow that, in steady-state, uses no more bandwidth than a conformant TCP flow running under comparable conditions, according to [1].

In single-rate multicast sessions, congestion control can be performed by the sender collecting feedback from the limiting receiver and adjusting the sending rate accordingly [13], quite similar as in the unicast case [2]. However, with a single transmission rate conflicting requirements of a set of receivers cannot be satisfied simultaneously, i.e., receivers with lower capacities may suffer congestion while others may have their capacities underutilized.

One of the first working examples of layered multicast transmission in the Internet was Receiver-driven Layered Multicast (RLM) developed by McCanne et al. [5]. The use of RLM to control congestion is problematic since RLM's mechanisms of adding or dropping a single layer based on the detection of packet loss are not TCP-compatible. Vicisano et al. address this problem in their work on Receiver-driven Layered Congestion Control (RLC) [11], which is based on exponentially dimensioned layer sizes, and generation of periodic bursts for bandwidth inference. The limitations of both protocols, RLM and RLC, are presented in [3], which are namely the inference mechanism and speed of convergence among others.

Instead of probing for available bandwidth, Turletti et al. [10], Tan and Zakhor [9], Sisalem and Wolisz [8], and Liu et al. [4] use a TCP response function to adjust the rate, which has been derived in [6]. Following this basic approach, we have developed an equation-based congestion control for layered multicast, which is based on the TCP throughput model and rate calculation algorithm as used in TFRC [2] and TFMCC [13]. Both protocols are accepted as quite promising solutions for the delivery of continuous media via unicast and single-rate multicast, respectively.

2.1 TCP throughput model

The steady-state TCP throughput according to the TCP model is calculated as a function of the round-trip time t_{RTT}, packet size s, and the steady-state loss rate p:

$$T_{TCP} = \frac{s}{t_{RTT}\left(\sqrt{\frac{2p}{3}} + 12\sqrt{\frac{3p}{8}}p(1+32p^2)\right)} \quad (1)$$

The obvious way to measure the loss rate is as a loss fraction calculated by dividing the number of packets that were lost by the number of packets transmitted. However, this does not accurately model the behavior of recent TCP implementations (NewReno, Sack), which halve the congestion window only once in response to several losses in a window of data. As a consequence, the loss rate measures the loss event rate rather than the packet loss rate. A loss event is defined as one or more losses during a round-trip time. The loss event rate can then be defined as $p = 1/n_\tau$ with n_τ denoting the number of packets transmitted in the time τ between two consecutive loss events.

To prevent rate oscillations, it is necessary to accurately measure and smooth loss and round-trip time values. Appropriate filters are presented and evaluated in [2], and since our implementation is based on TFRC, we refer to its original algorithms and parameters.

2.2 Limitations of the model

The TCP throughput model described in Section 2.1 assumes that both the round-trip time and the loss event rate are independent of the sending rate. This holds in environments with a high level of statistical multiplexing. But when only few flows share a bottleneck link, changes to the sending rate alter the conditions at the bottleneck link, which in turn can render the equation invalid. In our simulations we observed that the loss rate varies significantly with the sending rate, while the impact on the round-trip time is comparably low.

3. Basic approach

Our goal is to investigate the practicability of the TCP throughput equation for layered multicast in principle. For this purpose, we use the network simulator ns-2. To our knowledge, an implementation of a multicast protocol which could serve the purpose does not exist. But since at this stage we are only interested in the performance of the rate calculation algorithm based on the underlying control function, we modified the ns-2 code of TFRC. In the following, we refer to the modified protocol as L-TFRC. In contrast to the original implementation, in L-TFRC the sending rate r_{send} can only be coarse-grained adjusted to one of the discrete rates (L_1,..., L_m) with $L_i = \sum_i l_j$, where m denotes the number of layers and l_j the size of the jth layer. It is obvious that with a larger number of layers m, the scheme becomes fine-grained, which improves fairness. On the other hand, since this increases the overhead and complexity of the scheme, as a trade-off our scheme provides $m = 3$ possible sending rates [7].

3.1 Functionality

The basic functionality of L-TFRC is as follows:

1. The session starts with the transmission of the first layer for a fixed time interval t_{start}, to get an estimate of the current loss rate and round-trip-time.
2. The receiver measures and updates the loss event rate p and reports this value to the sender.
3. The sender measures the round-trip time and calculates the TCP-compatible rate r_{calc} according to the chosen TCP equation.
4. Depending on the value of r_{calc}, the sending rate is set to $r_{send} = \max(L_i \mid L_i < r_{calc})$.
5. 2–4 are repeated until the end of the session.

3.2 Simulation configuration

All our simulations were conducted using the network simulator ns-2. The network topology we used is the single bottleneck ("dumbbell") as shown in Figure 1. It consists of one L-TFRC source and receiver pair with $m = 3$ layers, and a varying number of competing TCP NewReno instances. All access links have a delay of 5 ms, and they are sufficiently provisioned to ensure that packet drops due to congestion occur only at the bottleneck link from R1 to R2. The bottleneck link is configured to have a bandwidth of $B = 2.5$ Mbps and a propagation delay of 20 ms with a RED Queue and FIFO scheduling. The buffer size is 20 packets and the packet size of all flows are set to 1,000 bytes.

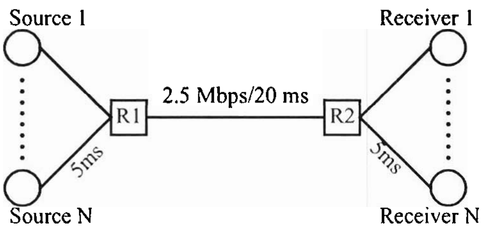

Figure 1: Simulation topology.

3.3 Simulation results

The objective of our initial simulation is to capture the principle steady-state behavior and transient behavior of the L-TFRC protocol. We set the sizes of the three layers such that $r_{send} \in \{L_i | L_i = 0{,}2 \cdot 2^{i-1} \cdot B, i = 1..3\}$. In order to examine the transient behavior, we change the number of competing flows from $n = 2$ to $n = 1$, and finally back to $n = 2$. Furthermore, to study the steady-state behavior, we keep each condition for a fixed time of 200 s starting at $t = 0$.

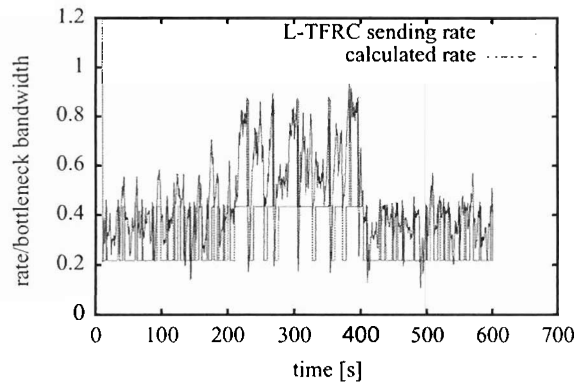

Figure 2: Transient and steady-state behavior of the basic L-TFRC when competing with a changing number of flows.

As demonstrated in Figure 2, L-TFRC seems to adapt to changing conditions appropriately, but in steady-state oscillates heavily between layer i and i+1 if $L_i \le r_{fair} < L_{i+1}$. Since the layers are distributed coarse-grained, in a streaming scenario this effect leads to pronounced oscillatory leaps in display quality, which is annoying for the user and should be reduced as much as possible.

To gain insight in the underlying reason for the steady-state effects described above, in the next simulation scenario one L-TFRC instance is competing with a single TCP instance. As this simple scenario is far from a statistical multiplexing environment, our interest is to study the rate calculation algorithm of the TFRC protocol implementation in cases where the sending rate differs from the fair share. Thus, the sending rate is set to each of the rates $r_{send} \in \{L_i | L_i = 0{,}25 \cdot i \cdot B, i = 1..3\}$ for a certain time interval in order to find the value, the algorithm converges to. We start sending at $t = 10$ s with L_1, continue with L_2 at $t = 140$ s, and L_3 at $t = 270$ s until $t = 400$ s.

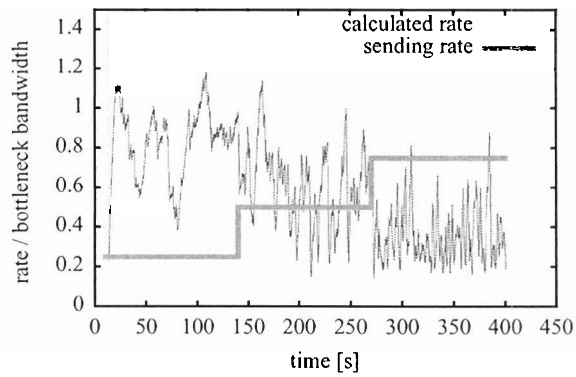

Figure 3: Calculated and sending rate.

The calculated rate in Figure 3 shows a quite pronounced variance, which would render solid join and leave decisions of the receivers difficult. This effect can be reduced through adjustment of the smoothing filters, in turn negatively impacting the responsiveness of the protocol. But neglecting the variance, the mean value of the calculated tends to be biased, a second effect which we are very interested in. When r_{send} is close to the fair share, the algorithm produces quite good estimations, whereas when the sending rate is below or above the fair rate, the calculated value is overestimated or underestimated, respectively. This is a quite important observation since with a layered transmission, where only few discrete sending rates are possible, we cannot assume the cumulative rates L_i to match the fair share. In general, the sending rate will be biased to a lower value since for the purpose of TCP compatibility $r_{send} = \max(L_i \mid L_i < r_{fair})$ should hold. Nevertheless, the erroneous estimation of the calculated rate will result in unwanted join and leave actions leading to the already observed oscillation between L_i and L_{i+1} such that $L_i \le r_{fair} < L_{i+1}$.

To further investigate the dependency of the calculated on the sending rate, in the next simulation we vary the latter such that $r_{send} = i \cdot 0.05 \cdot B$, where B denotes the bottleneck bandwidth and $i \in [1,19]$. We repeat this simulation for a different number n of competing TCP flows.

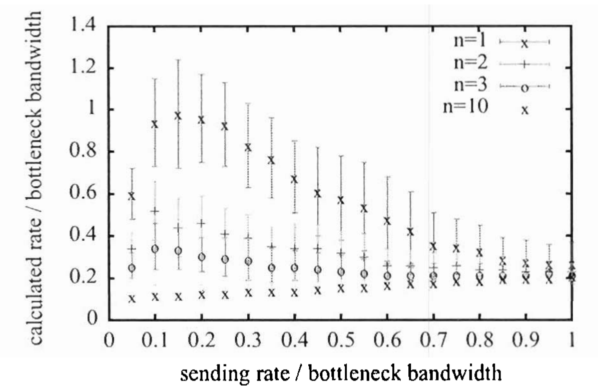

Figure 4: Impact of the sending rate on the calculated rate

Figure 4 quantifies the already noticed impact of the sending on the calculated rate. It shows that as the number of concurrent cross traffic flows increases, the effect reduces. Furtermore, in our simulations we observed that the impact of the sending rate on the loss event rate and its variance are much more pronounced than the impact on the round-trip time and its variance. For this reason, in our further investigations we focus on the loss event rate neglecting the round-trip time.

4. Improved approach

To prevent the described join and leave actions due to rate over- and underestimation, we propose to enhance the estimation algorithm by means of a correction term. As the difference of estimated rate and the fair share is mainly the result of a loss event rate, which does not correspond to the actual calculated rate, we investigate a correction of the loss event rate based on linear interpolation. The key idea is to increase or decrease the measured loss event rate as the calculated rate exceeds or undershoots the sending rate, respectively. The two fix points for the interpolation are the loss event rates measured at L_i and L_{i+1} where $L_i \le r_{calc} < L_{i+1}$.

Let L_i and L_{i+1} denote the transmission rates such that $L_i \le r_{fair} < L_{i+1}$. While receiving layers 1..i, due to the overestimation a join for layer (i+1) will be triggered. We store the loss event rate p_i at this moment. After a certain time, which is a function of the difference $(L_{i+1} - r_{fair})$, the increasing loss event rate forces a leave action from layer (i+1). The measured loss event rate at this moment p_{i+1} corresponds to the loss event rate for L_{i+1}. Both values, p_i and p_{i+1}, are used in the following correction term to estimate the new loss event rates for calculated rates between L_i and L_{i+1}:

$$p_{corrected} = p + \frac{r_{calc} - L_i}{L_{i+1} - L_i}(p_{i+1} - p_i) \quad , \qquad (2)$$

where r_{calc} denotes the last calculated rate, p the current loss event rate as estimated with the original algorithm, and $p_{corrected}$ its corrected value. The value of $p_{corrected}$ is then used to calculate the TCP-compatible rate r_{calc} with the TCP throughput formula.

This approach mimics a linear increase of the loss rate from p_i to p_{i+1} for a calculated rate augmenting from L_i to L_{i+1}. As the loss event rate directly influences the value of the calculated rate, the overestimation should be prevented. We note that in environments, where the loss rate is independent of the sending rate, the term $(p_{i+1} - p_i)$ equals zero and thus the correction has no influence on the original TFRC loss estimation.

4.1 Simulations

In the following simulations we keep the configuration as already introduced in Section 3.2. As with the basic approach, in the first simulation we set the sizes of the three layers such that $r_{send} \in \{L_i | L_i = 0{,}2 \cdot 2^{i-1} \cdot B, i = 1..3\}$. The simulation starts at $t = 0$ with one $n = 1$ competing flow, changes to $n = 2$ at $t = 200$, and back to $n = 1$ t = 400 s. In order to reduce level leaps due to the high variance, we applied a simple smoothing filter for the calculated rate.

Figure 5 and Figure 6 demonstrate the calculated rate and the resulting behavior of the basic L-TFRC without correction and the enhanced version with correction term, respectively. In both cases the L-TFRC instances follow the changing condition. The basic approach again shows frequent oscillations between the two levels. But the improved protocol needs only one leap to the higher level and back to get the parameters for the correction term and then acts quite smooth. The mechanism at every sending level works quite the same as previously described such that in each time interval the corrected value is close to the fair share rate

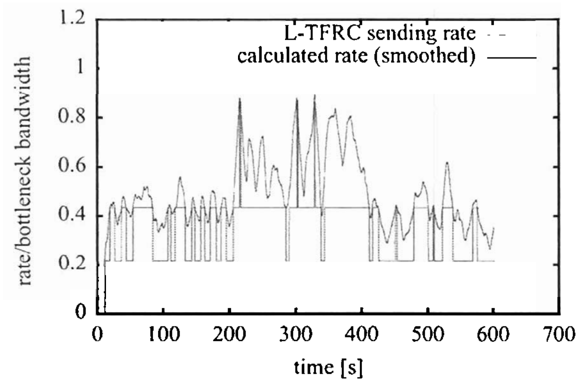

Figure 5: Calculated and sending rate of a L-TFRC without correction.

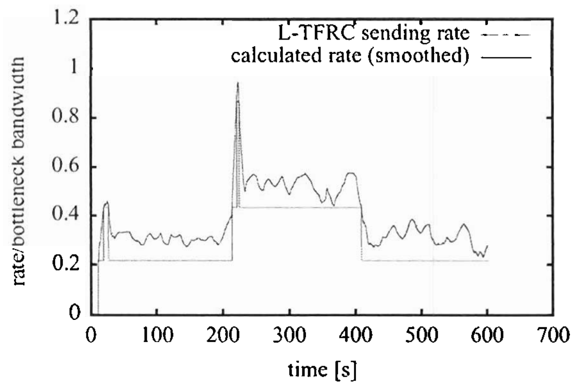

Figure 6: Calculated and sending rate of a L-TFRC with correction.

5. Conclusion and outlook

In simulations we have demonstrated that, although the use of an equation-based approach to a TCP-compatible layered transmission is an elegant solution, it shows pathological behavior in an environment with a low degree of statistical multiplexing, which leads to frequent and annoying oscillations between the discrete sending rates. To improve the performance in such environments, we presented a simple, yet effective heuristic, which implemented in a layered multicast protocol leads to a lower degree of erroneous join and leave actions.

In future work, we will confirm our heuristic approach with an analytical model, which might result in a better approximation of the behavior and possibly an improved solution. Finally, we will incorporate the obtained results in the development of an equation-based multi-rate multicast congestion control protocol.

6. References

[1] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floxd, V. Jacobson, S. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on queue management and cogestion avoidance in the Internet. RFC2309, April 1998.

[2] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM 2000*, pages 43–56, Stockholm, Sweden, August 2000.

[3] Arnaud Legout and Ernst W. Biersack. Pathological behaviors for RLM and RLC. In *Proceedings of the 10th International Workshop on Networking and Operating Systems Support for Digital Audio and Video (NOSSDAV 2000)*, Chapel Hill, NC, June 2000.

[4] Jiangchuan Liu, Bo Li, and Ya-Qin Zhang. A hybrid adaptation protocol for TCP-friendly layered multicast and its optimal rate allocation. In *Proceedings of IEEE Infocom 2002*, New York, NY, June 2002.

[5] Steven McCanne, Van Jacobson, and Martin Vetterli. Receiver-driven layered multicast. In *Proceedings of ACM SIGCOMM '96*, pages 117–130, Stanford, CA, 1996.

[6] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM '98*, Vancouver, BC, September 1998.

[7] Ivica Rimac, Jens Schmitt, and Ralf Steinmetz. Is dynamic layering worthwhile the effort? In *Proceedings of the 28th EUROMICRO Conference 2002 (Multimedia and Communications Track)*, Dortmund, Germany, September 2002.

[8] Dorgham Sisalem and Adam Wolisz. MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments. In *Proceedings of the 8th International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, June 2000.

[9] Wai-Tian Tan and Avideh Zakhor. Video multicast using layered FEC and scalable compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):373–386, March 2001.

[10] Thierry Turletti, Sacha Fosse-Parisis, and Jean Bolot. Experiments with a layered transmission scheme over the Internet. Research Report No. 3296, INRIA, France, Novemeber 1997.

[11] Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of IEEE INFOCOM '98*, San Francisco, CA, March 1998.

[12] Jörg Widmer, Robert Denda, and Martin Mauve. A survey on TCP-friendly congestion control. *Special Issues of the IEEE Network Magazin*, 2001.

[13] Jörg Widmer and Mark Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of ACM SIGCOMM 2001*, pages 275–286, San Diego, CA, August 2001.