[RSSt03] *Ivica Rimac, Jens Schmitt, Ralf Steinmetz;* **Observations on Equation-based Estimation of TCP-compatible Rate for Multi-rate Multicast Scenarios;** International Workshop on Multimedia Interactive Protocols and Systems MIPS 2003, Neapel, November 2003;

Observations on Equation-Based Estimation of TCP-Compatible Rate for Multi-rate Multicast Scenarios

Ivica Rimac, Jens Schmitt, and Ralf Steinmetz

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

Abstract. In this paper, we investigate the applicability of the equation-based approach to TCP-fair rate estimation for multi-rate multicast rate adaptation and congestion control in principle. The benefit of such an approach is two-fold: on the one hand, participating receivers might base their join and leave decisions on the calculated rate, instead of probing for the appropriate subscription level; on the other hand, the sender side might adapt and optimize the data rates according to receivers' feedback.

We analyze the characteristics of the underlying rate calculation model, instead of investigating one particular protocol. Therefore, we adopt an already existing algorithm utilized for closed-loop control, which has been validated in several published studies. Through extensive simulations we evaluate the behavior of the algorithm for different multiplexing levels at different transmission rates and timescales.

1 Introduction

1.1 Motivation

Video streaming over the best-effort Internet using IP multicast is a challenging and active research area. One of the significant remaining hurdles to widespread adoption of multicast transport for streaming media is the lack of suitable congestion control mechanisms. In particular, a multi-rate multicast congestion control mechanism mature for wide area deployment is still yet to emerge, despite considerable research efforts and numerous advances [2][3][4]. One obvious reason appears to be the inherent complexity in the development, as well as the testing and validation of such protocols.

The current Internet provides only best-effort service and routers do not exert active control over their bandwidth resources. Thus, different sessions compete for network resources, which demands for appropriate congestion control mechanisms. As a result, bandwidth allocation is a function of the control mechanisms used by the end-systems, which are expected to adopt the "social" rules implied by TCP and be cooperative by reacting to congestion signals and adapting their transmission rates properly and promptly. This paradigm of passive routers and active hosts has been very successful in today's Internet, where TCP-based traffic dominates. Its congestion management mechanisms are primarily responsible for the stability of the Internet despite rapid growth in traffic with respect to volume and diversity. TCP serves very well for reliable transfer of elastic traffic, but due to the way it is probing for available capaeity, it is producing rapidly varying transmission rates (sawtooth behavior). Thus, it cannot meet the requirements of streaming applications, which are better served by more slowly-responsible protocols producing smoother transmission rates instead of mimicking TCP behavior. Furthermore, TCP has been developed to provide a connection-oriented unicast service, which makes it unemployable for multicast. Hence, alternatives have to be developed.

Since TCP traffic is dominating the Internet, alternative congestion control mechanisms should behave in a TCP-compatible manner [5]. That is, on timescales of several round-trip times, a TCP-compatible flow obtains roughly the same bandwidth allocation as a TCP flow in steady-state. For the latter, an analytical model has been developed and widely accepted [6]. TCP-Friendly Rate Control (TFRC) is a well explored and mature representative of the class of unicast protocols, which apply this model for congestion control. Recently, this approach has been adopted by multicast protocols both, single-rate and multi-rate, designed for streaming media [7][2][3][4]. The expected benefit of applying equation-based rate estimation to multi-rate multicast schemes[1] is two-fold. First, a receiver participating in a session might be able to avoid probing and base its join and leave decisions on the calculated rate. Second, the sender might adapt and optimize the data rates according to the calculated rates fed back from the receivers. However, the nature of unicast and multicast communication is fundamentally different; in the latter case, a fine-grained closed-loop control is impossible, and transmission rates cannot be adapted to the reported conditions of each and every receiver. Naively adopting the equation-based approach to multicast might not result in the expected behavior. Nevertheless, this problem has not been addressed sufficiently in literature, instead it has often tacitly been assumed that the equation-based approach would be easily transferable to multicast solutions [8][2][3].

1.2 Contribution

The contribution of this paper is to provide insight into the quantitative performance of the equation-based approach to rate estimation, under diverse conditions in a multicast scenario. In contrast to the unicast case, the speed of response to variations is on a larger timescale and a mismatch between receiving rate and network path conditions of a receiver is quite common. Other research marginalizes this problem and assumes the basic approach to be precise enough under certain assumptions made in the context of the respective work [8][2][3].

In this paper, we take a step back and isolate a rate estimation algorithm based on the model in [6] in order to drive conclusions about its basic behavior under varying conditions. Possible pitfalls in applying the equation-baed approach are identified and pointed out. We are not aware of any comparable work, and hope to motivate further and intensive research on this interesting and important topic.

[1]Layered multicast is currently the most widely accepted solution to multi-rate multicast. Thus, in the rest of the paper we will focus on layered multicast, but the results are transferable to simulcast as well [1].

1.3 Outline

The paper is structured as follows. In Section 2, related work is surveyed and the trend in multicast congestion control research is briefly sketched, in order to underline the significance of our work. Thereafter, we give a description of the adopted TCP throughput model and the rate estimation algorithm in Section 3, which constitute the basis of our study. We explain the simulation configuration and analyze the results of our experiments in Section 4, and conclude in Section 5.

2 Related Work

There is a considerable body of work on multicast congestion control. The state-of-the-art on this topic has been covered in [9], except recently proposed work. From the classification of existing work it is obvious that a „one-size-fits-all" protocol is not feasible. Thus, in this paper we focus on multicast congestion control for streaming multimedia. We do not attempt to survey all existing work here in detail; we point out briefly to the in our opinion most significant approaches to the topic in order to show the current trends, and to underline the significance of this paper.

2.1 Single-rate Multicast Congestion Control

Single-rate multicast congestion control can be performed by the sender communicating to the limiting receiver and adjusting the sending window or rate to this receiver's feedback. Prominent representatives of this class of protocols are PGMCC [10] and TFMCC [7]. The former one uses feedback of the limiting receiver to adjust a congestion window similar to TCP, producing rate variations that resemble TCP's sawtooth behavior. On the other hand, TFMCC adopts the equation-based approach from TFRC [11], which makes it much more suitable for streaming applications. However, the single-rate approach suffers a major drawback, since it cannot scale to large, heterogeneous audience sizes. Receivers with heterogeneous capabilities and network conditions cannot be satisfied simultaneously; that is, receivers with lower capacities may suffer congestion while others may have their capacities underutilized.

2.2 Multi-rate Multicast Congestion Control

Rubenstein et al. showed that in theory multi-rate sessions can achieve several desirable fairness properties that cannot be obtained in general networks using single-rate multicast [12]. In a more pragmatic way, by means of simulation experiments, we showed in [13] how the number of possible rate alternatives impacts the aggregate receiver satisfaction in a multicast session. It is our belief that multi-rate multicast is a strong requirement to scale to large, heterogeneous audiences.

Standard approaches to multi-rate multicast employ layered multicast from a single source, which relies on hierarchical coding. One of the first working examples developed by McCanne et al. is Receiver-driven Layered Multicast—RLM [14]. However, the use of RLM to control congestion is problematic, since RLM's mechanisms of adding or dropping a 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 [15], which is based on the generation of periodic bursts for bandwidth inference. To emulate the behavior of TCP, layer sizes are dimensioned exponentially as well as the time interval to pass without a loss before trying to join a layer. Despite the observed limitations of both protocols, they stimulated brisk interest in the research community for the area of layered multicast transmission for streaming media.

2.3 Equation-Based Multi-rate Multicast Congestion Control

With the development of the TCP throughput model and the introduction of the TCP-compatible paradigm—we introduce these concepts in Section 3.1—research started to develop alternatives to probing-based congestion control, in order to overcome the inherent drawbacks like unfairness to TCP and oscillations due to inappropriate join and leave decisions.

In [8], Turletti et al. utilize the simple TCP model for receiver-driven layered audio streaming with focus on the loss estimator. With the development of fine-grained scalable codecs as recently adopted in MPEG-4 [16], hybrid schemes which involve sender-side adaptation of transmission rates became feasible. Sisalem and Wolisz in [2] presented a hybrid framework called MLDA for achieving TCP-friendly congestion control, which estimates the fair rate of a receiver as minimally the output of the extended TCP model. While expecting the equation to provide sufficiently good estimates, the authors put their main focus on the development of a scalable round-trip time measurement and feedback suppression. The latter is optimized under the assumption, that the cumulated rates are equally distributed between the minimum and maximum reported receiver rate. Liu et al. follow a similar approach in [3], trying to optimize the rate allocation in the Hybrid Adaptation protocol for TCP-friendly Layered Multicast—HALM. The authors propose solutions to the data distribution and the feedback suppression problem. Recently, Kwon and Byers proposed Smooth Multirate Multicast Congestion Control—SMCC [4]. SMCC utilizes the congestion control mechanism of TFMCC on each of the multicast groups comprising a session, enhanced with a method for additive increase join attempts. While protocol complexity is lower compared to MLDA and HALM, layer bounds are predetermined and cannot be adapted during a session.

2.4 Conclusion

Equation-based layered multicast is currently a trend that the research community follows to develop TCP-compatible multicast congestion control mechanisms for streaming media applications. Originally, solutions were receiver-driven but with recently proposed fine-grained coding schemes, hybrid approaches, where the sender actively adapts the transmission rates, become feasible.

While published work deals with several and important issues, such as scalable round-trip time estimation and feedback suppression, the possible limitations of the basic rate estimation algorithm have in our opinion not yet been addressed sufficiently.

3 Rate Estimation Approach

In this section, we review the underlying model, which is widely accepted for calculating the throughput a TCP flow is expected to reach in steady-state. Based on the throughput model, we describe the algorithms we use to estimate the parameters and the TCP-compatible rate.

3.1 TCP-Throughput Model

The TCP-compatible paradigm [5] transforms the requirement that all congestion control mechanisms must behave like TCP into a looser requirement that all congestion control schemes must be TCP-compatible. The cornerstone of this approach is the observation made by several research works, that the bandwidth allocation of a TCP flow in steady-state can be characterized well by an analytical model. Thus, a TCP-compatible flow is defined as a flow that, in steady-state and a timescale of several round-trip times, uses no more bandwidth than a conforming TCP flow running under comparable conditions.

A simplified analytical model, which does not take into account TCP timeouts, has been presented in [17] among others. The model formulates the throughput of a TCP flow as a function of the packet size s, a constant c (commonly approximated as $\sqrt{3/2}$), the round-trip time t_{RTT}, and the steady-state loss rate p:

$$T = \frac{s \cdot c}{t_{RTT} \cdot \sqrt{p}} \qquad (1)$$

More accurate results, especially in higher loss environments, are provided with a more complex approximation, derived in [6]:

$$T = \frac{s}{t_{RTT}\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right) \cdot p(1 + 32p^2)} \qquad (2)$$

Equation (2) models the throughput of TCP as a function of the packet size s, the round-trip time t_{RTT}, the steady-state loss rate p, and the TCP retransmit timeout value t_{RTO}.

Both models do not characterize the TCP throughput exactly, but they provide a good approximation. As already mentioned in [9], both models assume that the round-trip time and the loss rate are independent of the estimated rate. They are expected to give a good approximation in environments with a high level of statistical multiplexing such as Internet backbone links, where losses might be assumed to be randomized. But care has to be taken, when they are used to estimate the TCP-compatible rate under less ideal conditions, where the sending rate might impact the steady-state loss rate and render the results invalid.

3.2 Rate Estimation Algorithm

To prevent oscillations, it is necessary to accurately measure and smooth loss and round-trip time values. Likewise most of the existing work, we adopt the algorithms presented in [11] for the purpose of our investigations. These have been subject to a number of performance studies such as [18] and are de facto standards. Thus, we calculate the TCP-compatible rate according to Equation (2) and estimate the parameters using filters as presented and evaluated in [11]:

Retransmit timeout t_{RTO}. Instead of deriving t_{RTO} from the usual TCP algorithm, its value is set to $t_{RTO} = 4t_{RTT}$, since it is reported that, this simple heuristics works reasonably well in practice to provide fairness against TCP [11].

Round-trip time t_{RTT}. To prevent a spurious t_{RTT} value from having an excessive effect on the output of the rate estimation algorithm, an exponentially weighted moving average is used, similar to TCP.

Loss event rate p. The obvious way to measure the loss rate is as a loss ratio 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, in TFRC and TFMCC a receiver aggregates the packet losses into loss events, defined as one or more packets lost during a round-trip time t_{RTT}. If l denotes the loss interval, that is, the number of packets transmitted in the time τ between two consecutive loss events, the loss event rate can be calculated as $p = 1/l$.

Different methods for calculating the loss event rate have been extensively tested in [11]. As a result, the weighted average method outperformed both, the dynamic history window and the exponentially weighted moving average method. Thus, the average loss interval size can be computed as the average of the m most recent loss intervals $l_k, \ldots, l_{k-m+1}$:

$$l_{avg}(k) = \frac{\sum_{i=0}^{m-1} w_i l_{k-i}}{\sum_{i=0}^{m-1} w_i}$$

The weights w_i are chosen so that very recent loss intervals receive the same high weights, while the weights gradually decrease to 0 for older loss intervals. The eurrent interval since the most recent loss event is incomplete; we do not know when it ends. Thus, it is conservatively included in the estimation of the loss event rate if it increases the average loss interval:

$$p = \frac{1}{\max(l_{avg}(k), l_{avg}(k-1))}$$

A reader familiar with multicast protocols probably recognizes that the accurate round-trip time measurement in large-scale multicast is an important issue. In a comprehensive approach to protocol design, appropriate algorithms have to be integrated, for example, such as proposed in [2]. In the context of the present work, this is not an issue since it is our intention to investigate the basic behavior of the isolated algorithm. In particular, we study the dependency of the calculated rate on the actual transmission rate. As a consequence, we use the closed-loop measurement so that other effects do not interfere with the response of the basic rate estimation algorithm.

4 Experiments

Since analytical tractability is out of scope for the given complexity of the problem, we use the methodology of experimental analysis for our investigation. This section comprises the description of the experimental design and the discussion of the results.

We conducted our experiments using the ns-2 network simulator. As already stated, it is not our intention to investigate the behavior of a certain protocol; we are rather interested in the precision and limits of the isolated equation-based rate estimation algorithm. For this purpose, we implemented the algorithms to estimate the theoretical TCP-fair rate on top of a constant-bit-rate (CBR) agent. This serves our needs very well since it simulates the situation a receiver participating in a multicast session would experience. That is, the received (cumulative) rate is constant for a given time interval, and is very likely not to match the fair share. In the rest of the paper, we will refer to our implementation as CBRmod.

Our topology is the well-known single bottleneck („dumbbell"). All access links have a delay of 2 ms, and are sufficiently provisioned to ensure that packet drops due to congestion only occur at the bottleneck link. The bottleneck link is configured to have a propagation delay of 20 ms and a bandwidth of $B = n * 500$ kbps, where n denotes the number of concurrent flows (multiplexing level) on the bottleneck link. We run the simulations with both, droptail and RED queues, since we expect the drop policy to severely impact the behavior. The parameters of the queues are scaled as in [18].

In each simulation, a single instance of CBRmod is sharing the bottleneck link with (n-1) TCPSack instances. To avoid synchronization effects, the sources start transmission randomly within the first 3 seconds of the simulation. The packet size of all flows is 500 Bytes. During a single simulation run, the sending rate r_{snd} of the CBRmod flow is incrementally increased from 100 kbps to 900 kbps at a granularity of 50 kbps. Each sending rate is kept for 50 s, and we monitor the loss event rate p, round-trip time t_{RTT}, and the output of Equation (2) for T. We repeat each simulation 20 times, unless we state otherwise, and calculate the two-sided 95% confidence interval. In each plot, we normalize throughput, sending rate, and calculated rate to the "theoretic" fair share of 500 kbps per flow.

We recall that ideally the algorithm should

- give approximately the same results independent of the sending rate, and
- the calculated rate should exhibit as little variations as possible.

Given the above requirements are fulfilled, the algorithm would be applicable to a rceeiver-driven or hybrid multicast congestion control protocol as is.

4.1 Droptail Gateway

In order to draw conclusions about the behavior in a droptail environment, in Figure 1 we plotted the loss event rate p and the round-trip time t_{RTT} a CBRmod receiver estimates over a timescale $\tau = 0.5$ seconds.

The round-trip time is relatively constant and independent of the sending rate for all multiplexing levels, as depicted in Figure 1a). Furthermore, the narrow confidence interval on the short timescale indicates that the variation of the underlying time series is marginal. As a consequence, the round-trip time estimator performs adequately and does not influence the behavior of the algorithm as a function of the sending rate.

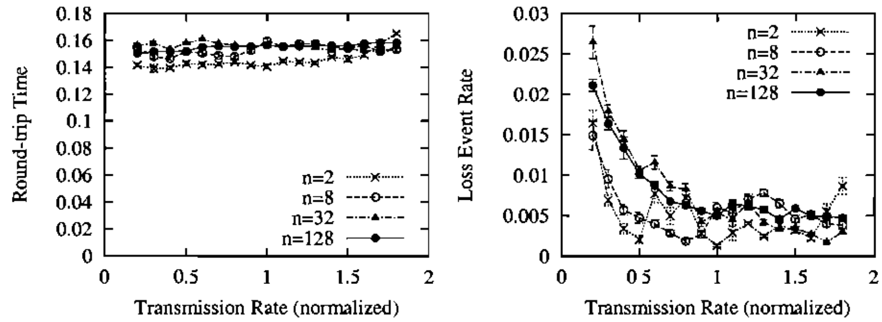

Figure 1. a) Round-trip time and b) loss event rate of a CBRmod flow competing with a varying number of (n-1) TCP flows on a droptail gateway.

Figure 1b) demonstrates, that the loss event rate is relatively smooth, but exhibits a pronounced dependency on the sending rate in the region where the latter is below the theoretic fair share. As a result, the calculated rate is underestimated in that region and its value increases with the sending rate approaching the theoretic fair level, as depicted in Figure 2.

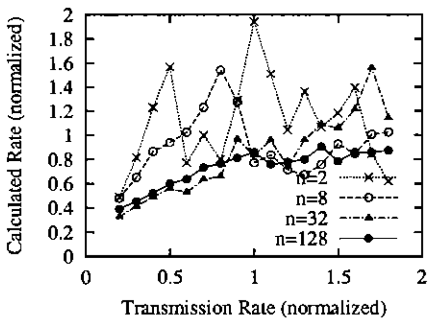

Figure 2. Calculated rate of a CBRmod flow competing with a varying number of (n-1) TCP flows on a droptail gateway.

The reason for the observed behavior can be attributed to the periodic cycles of TCP's control algorithm and CBRmod's sending behavior. Control theory suggests that this periodicity can resonate with the deterministic drop policy iu the gateway. As a result, the CBRmod flow, which periodically sends its packets, is penalized with a decreasing sending rate. This holds for higher multiplexing environments, where the influence of the CBRmod on the cycle of the TCP flows is negligible. For a very low level of multiplexiug Figure 2 shows that the CBRmod flow has larger influence on the periodicity of the competing TCP flows, resulting in a more complex dependency. To illustrate our observations, in Figure 3 we plot the actual receiving rate and the calculated rate of the CBRmod receiver competing with 3 and 127 concurrent TCP flows, respectively.

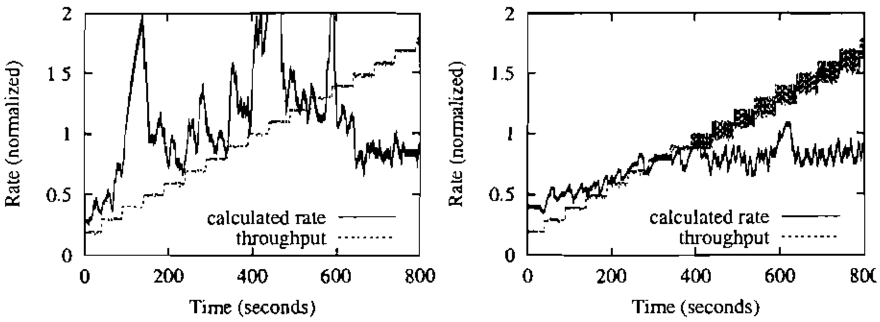

Figure 3. Actual receiving rate and calculated TCP-compatible rate of a CBRmod flow competing with a) 3 TCP flows and b) 127 TCP flows on a droptail gateway.

Remark. The intersection of the calculated rate and the sending rate, which an adaptive flow would converge to, does not reach the expected value. We verified this behavior in simulations with the TFRC implementation. As a result, this is a limitation of the underlying model, not specific to our work.

4.2 RED Gateway

In the following, we describe the results we obtained by substituting the droptail with a RED queue.

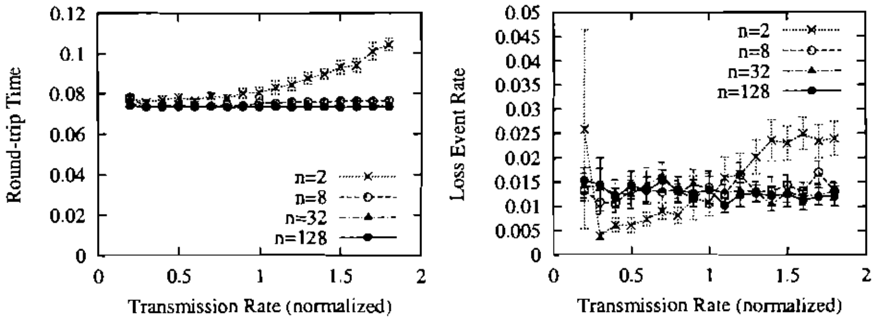

Figure 4. a) Round-trip time and b) loss event rate of a CBRmod flow competing with a varying number of (n-1) TCP flows on a RED gateway.

Figure 4a) and b) demonstrate, that at a very low degree of multiplexing the sending rate impacts both, the round-trip time and loss event rate. However, already at a slight increase of competing flows, this effect is not noticeable any more.

Another important observation we made is, that the loss event rate and consequently the calculated rate as depicted in Figure 5 have a large confidence interval. This implies, that they are subject to relatively high variations. Thus, we calculated the coefficient of variation (CoV) of the calculated rate, which reaches a value of up to 25 percent.

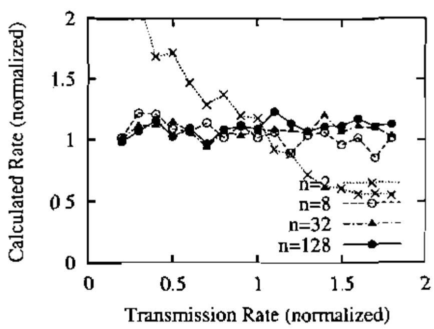

Figure 5. Calculated rate of a CBRmod flow competing with a varying number of (n-1) TCP flows on a RED gateway.

The observed behavior is attributed to the properties of the RED gateway. The latter conducts a Bernoulli experiment for each packet, whereby the probability of a drop is increasing with the average of the queue size. As a result, the loss process is randomized, and on average the loss event rate does not depend on the induced traffic of the CBRmod flow. This behavior would fit the needs of a receiver in our scenario well. However, since the average queue length of the RED gateway is varying due to characteristics of the TCP traffic, so does the loss probability. Thus, we observe a high variation of the calculated rate in Figure 6.

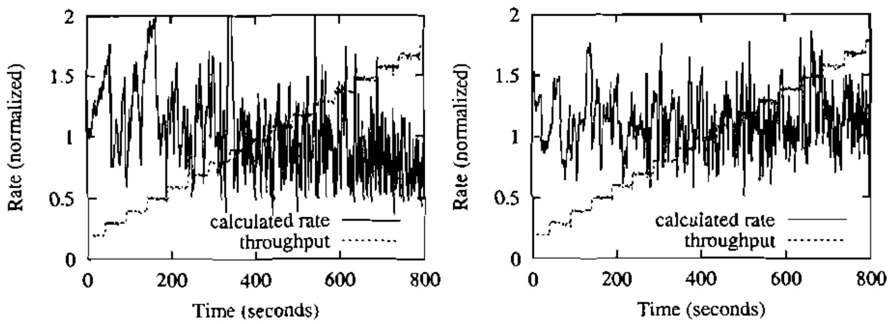

Figure 6. Actual receiving rate and calculated TCP-compatible rate of a CBRmod flow competing with a) 3 TCP flows and b) 127 TCP flows on a RED gateway.

4.3 Summary

Our results show, that the TCP-compatible rate estimator as currently adopted in multicast scenarios has limitations which would have to be—but currently are not—addressed sufficiently in a comprehensive protocol design.

In low-multiplexing droptail environments, the algorithm hardly gives any reasonable estimate. Under high-multiplexing conditions, when the sending rate is below the TFRC-equivalent rate, the target rate is underestimated. Depending on the arrangement of the layers of a multicast session, a participating receiver basing its join decision on the algorithm as is, has a good chance of keeping the current subscription level though being far from the appropriate TCP-fair bandwidth allocation.

When using RED gateways, the average value of the calculated rate is close to the theoretic fair level. But due to the relatively high coefficient of variation, the calculated rate converges to the operating point on a timescale of at least several seconds. Obviously, if smoothing over this timescale is performed, it makes the protocol quite unresponsive. On the other hand, measuring on a smaller timescale would allow faster action, but could lead to quite oscillatory behavior.

Consequently, increasing the granularity of layers might alleviate the reported effects and result in a more closed-loop like adaptation behavior. However, this is also not so attractive since it necessitates a large number of layers, which would result in a high management and routing overhead. Thus, in future work we will investigate other means such as optimized probing for improvement of the basic scheme.

5 Conclusion

In this paper, we presented our observations made with an algorithm for equation-based fair rate estimation in the context of multi-rate multicast streaming sessions. We focused on identifying and understanding the limits of the approach rather than on designing another multicast congestion control protocol with heuristic solutions. For the purpose of our work, we implemented a rate estimation algorithm on top of a CBR source in the network simulator ns-2, based on the common TCP-throughput model.

Through extensive simulations we showed, that under certain conditions, the algorithm widely used for rate control in unicast sessions when adopted as is to the multicast scenario will lead to undesired behavior. In a droptail environment, due to the deterministic loss process, receivers might underestimate their fair share and refrain from joining an additional layer although appropriate. While this is circumvented by RED gateways through randomization of the losses, the inherent fluctuation of the loss rate results in heavy variations of the calculated rate. Depending on the arrangement of the layers, this might lead to annoying oscillatory behavior due to inappropriate join and leave decisions.

The insight presented in this paper should aid in the development and testing of equation-based multi-rate multicast rate and congestion control protocols. As future work, we perceive combining the equation-based approach with probing mechanisms.

6 References

[1] B. Li and J. Liu. Multirate video multicast over the internet: An overview. *IEEE Network*, 17(1):24–29, January/February 2003.

[2] D. Sisalem and A. 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.

[3] J. Liu, B. Li, and Y.-Q. 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. IEEE.

[4] G.-I. Kwon and J. W. Byers. Smooth multirate multicast congestion control. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, April 2003.

[5] 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 congestion avoidance in the Internet. RFC2309, April 1998.

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

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

[8] T. Turletti, S. Fosse-Parisis, and J. Bolot. Experiments with a layered transmission scheme over the internet. Research Report No. 3296, INRIA, France, Novemeber 1997.

[9] J. Widmer, R. Denda, and M. Mauve. A survey on TCP-friendly congestion control. *Special Issues of the IEEE Network Magazin*, 2001.

[10] L. Rizzo. Pgmcc: A TCP-friendly single-rate multicast congestion control scheme. In *Proceedings of ACM SIGCOMM 2000*, pages 17–28, Stockholm, Sweden, August 2000. ACM.

[11] S. Floyd, M. Handley, J. Padye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM 2000*, pages 43–56, Stockholm, Sweden, August 2000. ACM.

[12] D. Rubenstein, J. Kurose, and D. Towsley. The impact of multicast layering on network fairness. In *Proceedings of ACM SIGCOMM '99*, Cambridge, MA, August 1999. ACM.

[13] I. Rimac, J. Schmitt, and R. Steinmetz. Is dynamic layering worthwhile the effort? In *Proceedings of the 28th EUROMICRO Conference 2002 (Multimedia and Telecommunications Track)*, pages 233–239, Dortmund, Germany, September 2002. IEEE.

[14] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of ACM SIGCOMM '96*, pages 117–130, Stanford, CA, 1996. ACM, ACM Press.

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

[16] F. Pereira and T. Ebrahimi, editors. *The MPEG-4 Book*. Prentice Hall PTR, Upper Saddle River, NJ, 2002.

[17] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999.

[18] D. Bansal, H. Balakrishnan, S. Floxd, and S. Shenker. Dynamie behavior of slowly-responsive congestion control algorithms. In *Proceedings of ACM SIGCOMM 2001*, San Diego, CA, August 2001.