Comparison of Single-Rate Multicast Congestion Control Protocols vs. ASMP

Christos Bouras
Research Academic Computer Technology Institute and University of Patras
bouras@cti.gr

Apostolos Gkamas
Research Academic Computer Technology Institute
gkamas@cti.gr

Georgios Kioumourtzis
Universal Dispensing Authority University of Patras
gkioumou@ceid.upatras.gr

Abstract

We present in this paper a simulation-based comparison of two single-rate multicast congestion control schemes (TFMCC and PGMCC) against our proposed Adaptive Smooth Multicast Protocol (ASMP). ASMP consists of a single-rate multicast congestion control, which takes advantage of RTCP Sender (SR) and Receiver Reports (RR). The innovation in ASMP lays in the "smooth" transmission rate, which is TCP-friendly and prevent oscillations. The smooth behavior is naturally well suited to multimedia applications as high oscillations of the sending rate may create distortions of Audio-Video (AV) encoders and decoders. Simulation results, which are conducted with the network simulator ns2 software, showed that ASMP can be regarded as a serious competitor of TFMCC and PGMCC. In many cases, ASMP outperforms TFMCC in terms of TCP-friendliness and smooth transmission rates, while PGMCC presents lower scalability than ASMP.

1. Introduction

Congestion control is an important attribute for multicast transport protocols as it is described in RFC 2357 [1]. The existing complexity of congestion control for multicast transmission increases when one tries to accommodate such control mechanisms for multimedia data over best-effort networks. Multimedia applications pose their own requirements and restrictions to congestion control mechanisms, as high oscillations of the transmission rate may lead to infeasible adaptations of the Audio-Video (AV) coders and decoders.

Although congestion control for multimedia data transmission involves various contradictory requirements, we believe that at least the three following requirements should be satisfied:

- Any proposed congestion control should prevent oscillations, as much as possible, in order to minimize the Audio-Video (AV) encoding and decoding distortion.
- Inter-arrival jitter delay should be small in order to meet the multimedia application's requirements.
- Packet losses should be minimized and when exist they should have minimal negative results in end user's perception.

Adaptive Sooth Multicast Protocol (ASMP) stands for our proposed solution that constitutes a single-rate congestion control for multimedia data transmission, which runs on top of RTP/RTCP protocols. The key attributes of ASMP are: a) TCP-friendly behavior, b) adaptive scalability to large sets of receivers, c) high responsiveness to network changes, and finally d) smooth transmission rates, which are suitable for multimedia applications. In our proposal each receiver calculates a TCP-friendly bandwidth share based on the TCP analytical model presented in [2]. It is worth mentioning that ASMP does not require any additional support from the routers or the underlying IP-multicast protocols. This property allows easy deployment over unmanaged networks, like the Internet.

We present in this work a simulation-based comparison of ASMP against two well-known congestion control schemes; TFMCC ([3], [4]) and PGMCC [5]. Simulations conducted with the network simulation software ns2 [6]. As for the evaluation criteria we considered the metrics described in [7].

The rest of this work is organized as follows: Next section discusses related work in the field of single-rate multicast congestion control schemes. We briefly discuss ASMP in section 3. Simulation scenarios and results are discussed in section 4. We conclude our paper in section 5.

2. Related Work

Multicast of multimedia data is an appealing approach especially for the transmission of popular live events in large group of receivers. Recent work has offered new proposals in the field of multimedia multicast protocols. We can broadly classify these approaches into three main categories: (a) Single Layer Design: The sender transmits a single layer and the transmission rate is defined by the lowest receiver, which is the one with the lowest bandwidth capacity. (b) Layered transmission: Multimedia data is transmitted by different streams and each individual receiver joins the multicast stream that is closer to its bandwidth capabilities. (c) Replicated transmission: Under this approach, receivers form different multicast groups and each receiver joins the multicast group that is closer to its bandwidth capabilities.

Single-rate multicast congestion control algorithms cannot be seen only as independent schemes, but also as building blocks of multi-rate congestion control mechanisms. SMCC [8] and GMCC [9] are good examples of multi-rate schemes that are based on single-rate congestion control mechanisms. In this paper we focus on the field of single-rate multicast congestion control. The discussion of the limitations of single-rate multicast protocols versus multi-rate protocols is out of the scope of this paper.

Up to now there are promising approaches in this field in literature. TFMCC [4] extends the basic mechanisms of TFRC [10] to support single layer multicast congestion control. The most important attribute of TFMCC is the suppression of feedback receiver reports. TFMCC uses the receiver with the lowest receiving capacity to act as the representative of the multicast group. PGMCC is a window-based TCP scheme, which is based on positive ACKs between the sender and the group representative (the acker). Only the acker is allowed to send positive ACKs to sender, mimicking the "classic" TCP receiver. All other receivers in the multicast session send NACKs. TBRCA [11] targets at maximizing the overall amount of multimedia data to the whole set of receivers. With the use of a bandwidth rate control algorithm it dynamically controls the output rate of the video coder. This is different philosophy from TFMCC and PGMCC. LDA+ [12] employs a TCP equation based congestion control for measuring a TCP friendly bandwidth share. LDA+ uses the RTP [13] protocol for collecting loss and delay statistics from the receivers. ERMCC [14] implements a congestion control scheme that is based on a new metric named TRAC (Throughput Rate At Congestion). The feedback suppression philosophy is similar to that of TFMCC, although it seems to offer higher scalability.

3. ASMP Overview

ASMP consists of a single-rate multicast congestion control, which takes advantage of RTCP Sender (SR) and Receiver Reports (RR). The innovation in this work is the calculation of smooth transmission rate, which is performed by receivers and is based on RTCP reports. Our main objective is to adjust the sender's transmission rate in such way that oscillations are reduced, following a smooth fashion. Another important attribute is the long term TCP-friendliness, meaning that the multimedia stream consumes no more bandwidth than a TCP connection, which is traversing the same path with the multimedia stream. Moreover, with the use of RTCP feedback reports we provide better scalability, as the amount of these feedback reports are controlled by RTCP protocol and they cannot exceed a specified threshold, as percentage of the total available bandwidth. Without disseminating any additional feedback reports (ACKs or NACKs) than those of RTCP sender and receiver reports, we increase bandwidth utilization for user data. Lastly, we develop ASMP on top of RTP/RTCP protocol [16]. This approach has several advantages because we move the complexity up to the application layer, leaving untouched the operating system and network elements, as well. RTP with TCP-friendly control [17] is also built on this approach. The only visible drawback we can see is related to high time intervals between two consecutive RTCP feedback reports in a large multicast group. As a result, the sender cannot react quickly when network conditions change very rapidly. However, we increase responsiveness with the use of network statistical information. This information is based on what we call Congestion Indicators (CI), which provides the warnings of upcoming congestion so that the "smoothness factor" is tuned to a proper value. The smoothness factor regulates the behavior of the congestion control mechanism, making it less or more aggressive, in respect with the level of congestion in the bottleneck link. In this way, we succeed to have a congestion control that is "smooth", without suffering from high oscillations, and at the same time very responsive to network changes. We will observe in the simulation results how the proposed scheme reacts to different events that cause changes to network conditions.

A high level overview of the functionality of ASMP is presented below: (a) The receiver measures the loss event rate and the jitter delay based on RTP packet sequence numbers and timestamps. (b) The receiver measures the RTT between itself and the sender based on receiver's one-way time measurements and the sender RTCP feedback reports. (c) The receiver measures a TCP-friendly bandwidth share with the use of the analytical model of TCP. (d) The receiver is notified by Congestion Indicators (CI) in order to adjust the "smoothness factor", and calculates a new smooth TCP-friendly transmission rate. (e) The receiver sends the calculated smooth TCP-friendly rate to the sender using the extension mechanisms of RTP/RTCP. (f) The sender adjusts its transmission rate based on RTCP feedback receiver reports. More details on the smooth congestion control that is implemented in ASMP can be found in [18].

4. Simulations

4.1. Comparison with TFMCC

We implement TFMCC and ASMP in ns2 (source codes can be found in [19] and [20], respectively) to evaluate its performance under a controlled environment. We run several simulations to investigate: (a) The TCP-friendly behavior, when multicast receivers share the same bottleneck link with multiple TCP connections. (b) The behavior of each congestion control when a “slow” receiver late joins the multicast session. (c) The responsiveness to rapid changes of the network conditions due to packet losses in the link, and finally (d) The responsiveness to dynamics of other competing data.

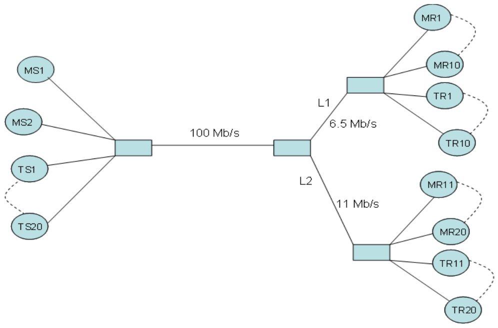

Figure 1. TCP-fairness topology

4.1.1. TCP Fairness. In this simulation we evaluate the fairness of TFMCC and ASMP towards competing TCP traffic.

We use a bottleneck scenario in which two multicast senders share multiple bottleneck links with twenty TCP Agents (figure 1). MS and MR represent the multicast sender and receiver, whereas TS and TR stand for the TCP sender and receiver respectively. We set the initial rate of both the multicast and TCP sender to 150 Kb/s. The bottleneck links should be equally shared by multicast flow and TCP traffic, which means that the multicast receiver must not consume more than 9.09% of the bottleneck bandwidth (each bottleneck link, L1 and L2, is shared by 10 TCP connections and 1 multicast flow). As a result we expect each flow to receive $100\% / 11 = 9.09\%$ of the bottleneck bandwidth. Therefore, multicast receivers in the low capacity link (L1=6.5 Mb/s) must not consume more than 590.85 Kb/s, whereas in the higher link (L2=11 Mb/s) must not consume more than 1 Mb/s. The rest of the available bandwidth should be consumed by TCP connections and it is expected that each TCP connection will receive at least the same bandwidth share with the multicast flow.

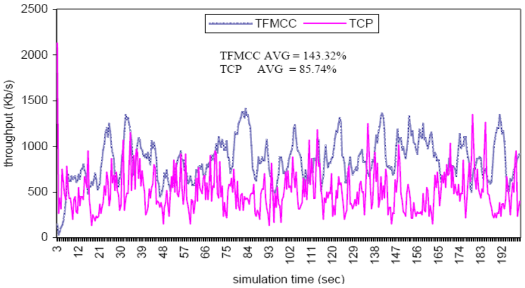

Figure 2. TFMCC vs TCP traffic in L1

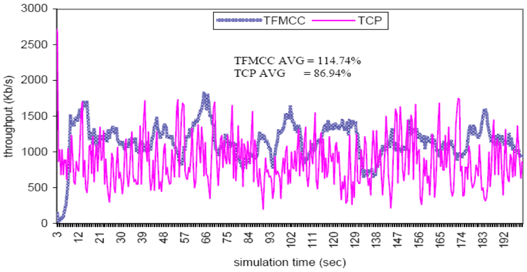

Figure 3. TFMCC vs TCP traffic in L2

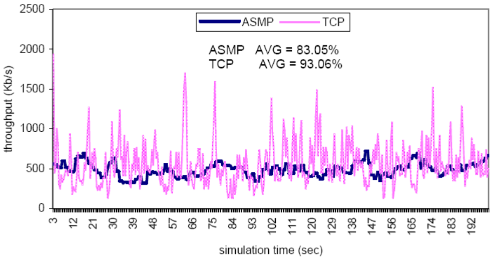

Figure 4. ASMP vs TCP traffic in L1

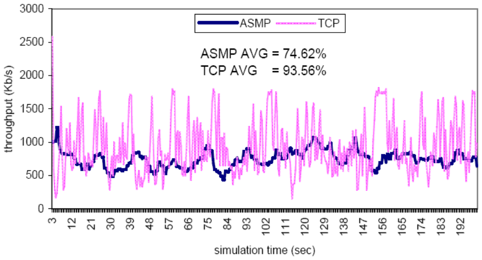

Figure 5. ASMP vs TCP traffic in L2

Figures 2 and 3 present the achieved throughput of TFMCC receivers versus TCP receivers in links L1 and L2, respectively. We observe that TFMCC in the low capacity bottleneck link (L1) consumes on average 143% of the available bandwidth while TCP is close to 85% of its share. In the higher capacity link (L2), TFMCC enjoys again higher bandwidth share than TCP connections. It is our assessment that TCP will suffer from starvation in the light of multiple TFMCC flows in Internet as TFMCC consumes much more bandwidth than a TCP flow when sharing the same bottleneck links.

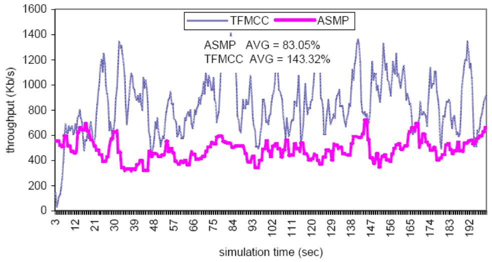

Figure 6. TFMCC vs ASMP bandwidth share in L1

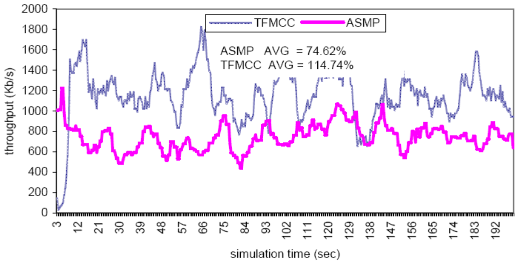

Figure 7. TFMCC vs ASMP bandwidth share in L2

On the other hand, ASMP is more TCP-friendly than TFMCC. We observe in figures 4 and 5 that TCP enjoys a bandwidth share of 93% in both cases in L1 and L2,, which is very close to absolute value of a TCP-friendly bandwidth share (100%). ASMP consumes less than its share but still high enough and close to 83% in the low capacity link.

However, except for TCP-friendliness we observe that ASMP is smoother than TFMCC when we directly compare the transmission rates of each congestion control scheme in L1 and L2 (figures 6 and 7). It is fair to say that ASMP presents much smoother transmission rates than TFMCC, which seems to suffer sometimes from high oscillations. A summary of the achieved throughputs is also presented in tables 1 and 2.

With this experiment we verify that not only ASMP is more TCP-friendly than TFMCC but it is also smoother, which is a desired attribute for multimedia applications.

Table 1. TFMCC vs TCP Achieved Throughput

	Link utilization	Average transmission rates
TFMCC	L1: 143.32%	846,800.6 Kb/s
	L2: 114.74%	1.147 Mb/s
TCP	L1: 85.74%	506.59 Kb/s
	L2: 86.94%	869.40 Kb/s

Table 2. ASMP vs TCP Achieved Throughput

	Link utilization	Average transmission rates
ASMP	L1: 83.05%	490 Kb/s
	L2: 74.62%	746.20 Kb/s
TCP	L1: 95.06%	561.66 Kb/s
	L2: 93.56%	935.60 Kb/s

4.1.2. Late Join of Low-rate Receiver. In this simulation scenario a bottleneck link (L1) with capacity of 1.5 Mb/s is shared by one multicast sender and four TCP connections (figure 8).

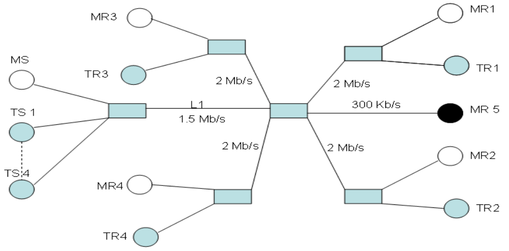

Figure 8. Late-join scenario topology

In the beginning of the simulation one multicast sender transmits at a rate of 500 Kb/s and four receivers join the session. At the 50th simulation second a late receiver (MR5) with lower receiving capacity (300 Kb/s) joins the multicast group. We observe that TFMCC needs few seconds to realize the presence of a low-capacity receiver before adjusting its transmission rate. In addition, TFMCC reacts aggressively by dropping immediately the transmission rate in the light of the first packet losses (figure 9). In contrast, ASMP does not make abrupt changes in the transmission rate and adjusts its transmission rate as it is expected. At the 100th simulation second the low capacity receiver leaves the session and we observe that TFMCC increases again its transmission rate in such way that creates high oscillations. ASMP starts gradually regaining its previous high transmission rates (figure 9).

Our conclusion from this experiment is that not only ASMP keeps a smooth transmission rate but reacts also rapidly to network topology changes.

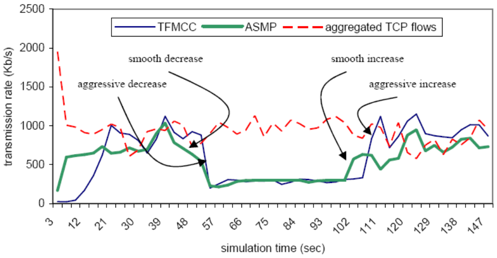

Figure 9. TFMCC vs ASMP transmission rates

4.1.3. Responsiveness to Changes in the Loss Rate.

In this simulation scenario we investigate how TFMCC and ASMP respond to changes in the loss rate and evaluate its performance.

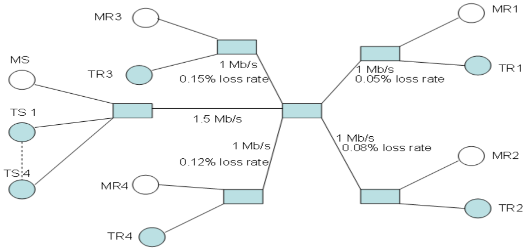

Figure 10. Varying loss rates topology

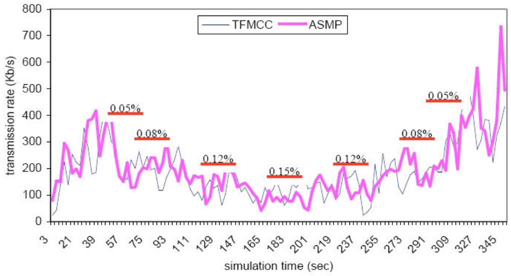

Figure 11. Responsiveness to varying loss rates

Loss rate variances affect the estimated TCP-friendly bandwidth share. We use a star topology with four links, having loss rates of 0.05%, 0.08%, 0.12%, and 0.15%, respectively (Figure 10). At the beginning of the simulation we only have one receiver that joins the session while the rest of receivers join the session every 50-second intervals in the order of their loss rate. Receivers with lower loss rates in their links join first the session. After 200 seconds, receivers leave the session in reverse order; receivers with higher loss rates in their links leave first the session. TCP background traffic is transmitted in the four links along with multicast traffic. Figure 11 depicts the simulation results. We observe that the smooth attribute does not seriously affect the responsiveness of ASMP. The reaction of TFMCC is very close to that of ASMP. We expected this behavior, as one important design goal of TFMCC is its smooth reaction to packet losses, so that it can be used for multimedia data transmission. Our implementation offers even smoother reactions in the light of packet losses than TFMCC.

4.1.4. Responsiveness to Dynamics of Competing Traffic.

In this simulation we compare TFMCC and ASMP in a heterogeneous dynamic network that was used for similar experimentations in [14].

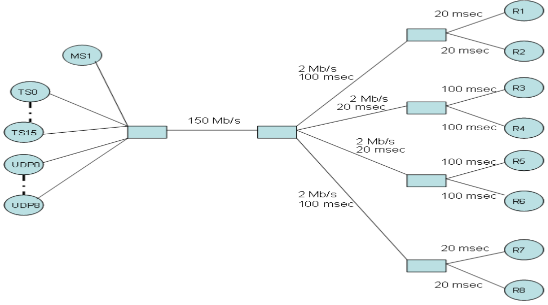

Figure 12. Heterogeneous dynamic network

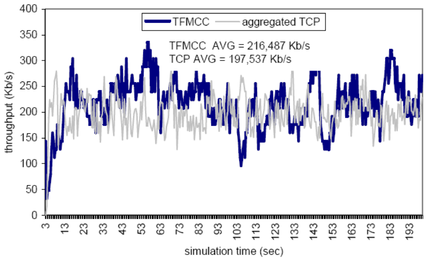

Figure 13. Responsiveness to competing traffic: TCP achieved throughput with competing UDP and TFMCC multicast flows

In figure 12 receivers R1 through R8 are connected with 2 Mb/s links and there is at most one link with 100 msec delay between senders and receivers. On each link two TCP flows are randomly turned on and off according to Pareto distribution with average value of 5 sec. Two additional UDP flows of 400 Kb/s on each link are also randomly turned on and off. These flows dynamically generate bottlenecks and make the network heterogeneous. At least there is a multicast flow from multicast sender (MS1) to receivers. Therefore, at every time instance there are at most five flows on any link: two TCP, two UDP and one multicast. The multicast flow is expected to receive approximately 400 Kb/s bandwidth share. We observe in the simulation results (figure 13) that TFMCC presents higher throughput than ASMP (figure 14), while TCP throughput is almost the same (better when competing with ASMP) in both cases. Another interesting observation is that ASMP has high stability and adaptability in this heterogeneous environment, although we expected that the smooth functions and high time intervals of RTCP reports would degrade its performance. However, it becomes clear that we need to increase the achievable throughput of ASMP in order to reach at least TCP throughput when both flows share the same bottleneck link (figure 15).

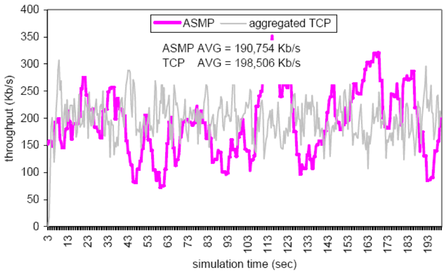

Figure 14. Responsiveness to competing traffic: TCP achieved throughput with competing UDP and ASMP flows

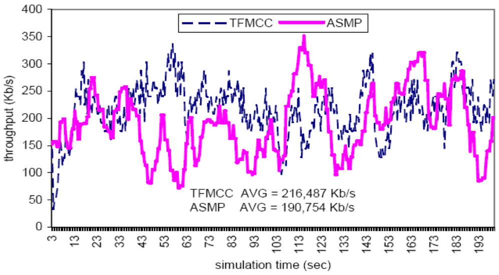

Figure 15. Responsiveness to competing traffic: TFMCC vs ASMP

4.2. Comparison with PGMCC

For the comparison between ASMP and PGMCC we use simulation results from [5] and compare them with traces from identical sumulations that we conducted with ns2.

4.2.1. TCP Fairness. Figure 16 shows a simulation scenario involving two PGMCC receivers (MR1, MR2), which belong to the same flow, and one TCP receiver (TR). MS and TS stand for the multicast and TCP senders, respectively. Links L1 and L2 have capacity of 400Kb/s and 500 Kb/s and the propagation delay for both links was set to 50 msec. In the beginning of the simulation receiver MR2 is started first, followed by MR1 and by TR.

We observe in figure 17 that in the beginning of the simulation PGMCC sender (MS) occupies the available bandwidth and when receiver MR2 joins the session (simulation time 17:04) it reduces its transmission rate to 400 Kb/sec. PGMCC sender further reduces its transmission rate down to approximately 220 Kb/s when the TCP sender (TS) starts its transmission (simulation time 17:05). When TCP sender terminates its transmission (simulation time 17:07) PGMCC sender increases the session rate to 500 Kb/s causing congestion to MR1. The congestion and packet losses result to an acker switch from MR2 to MR1 and the transmission rate is adjusted to a value around 400 Kb/s. It is interesting that PGMCC presents stable behavior throughout the simulation. This mainly happens due to the fact that the acker switch occurs only between two PGMCC receivers. Our comment is that PGMCC transmission rate is indeed TCP-friendly, although in this mild-congested scenario PGMCC has lower throughput than TCP traffic.

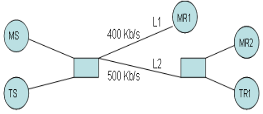

Figure 16. Bottleneck network

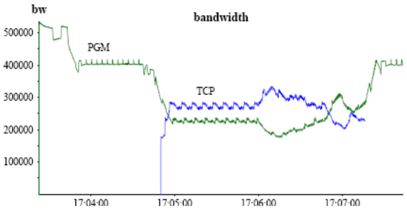

Figure 17. PGMCC vs TCP traffic

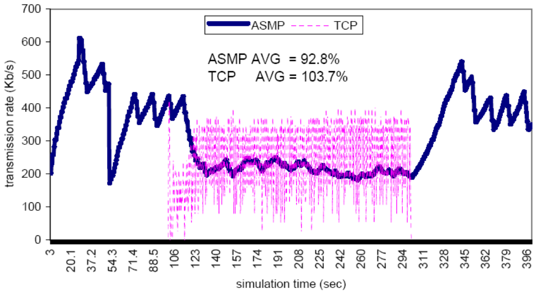

Figure 18. ASMP vs TCP traffic

Figure 18 presents simulation results from the same topology with ASMP. We observe that in the beginning of the simulation ASMP sender tries to occupy all the available bandwidth. At the 50th simulation second ASMP receiver (MR1) joins the session and the sender reduces its transmission rate below 200 Kb/s, based on TCP-friendly estimations of MR1. This happens because when a receiver first joins a session it has a low estimation of the available bandwidth. It takes the algorithm few seconds to converge and indeed at the 57th simulation second receiver MR1 calculates the available bandwidth to be around 400 Kb/s. At the 100th simulation second MS further reduces its transmission rate as a result of congestion that is caused by TCP traffic. We observe that MS transmission rate is stable and smooth when link L2 is shared by multicast and TCP traffic. L2 is almost equally shared by the two sources. MS sender has an average transmission rate of 232,068 Kb/s and TCP 259,401 Kb/s. When TCP sender stops its transmission the multicast sender increases gradually its transmission rate in a smooth fashion, as it was expected.

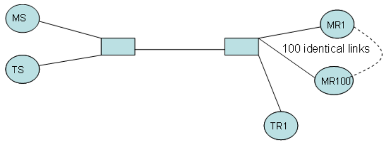

Figure 19. Multicast Receivers, one TCP

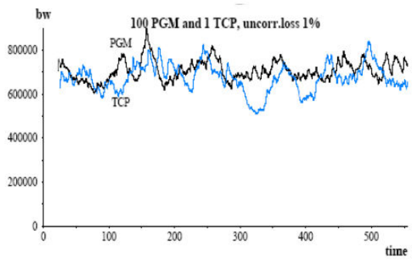

Figure 20. PGMCC responsiveness to mild losses

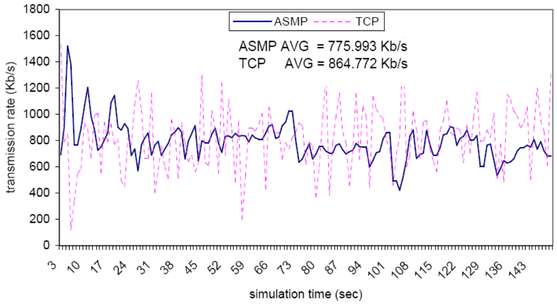

Figure 21. ASMP responsiveness to mild losses

We verify with this simulation scenario that ASMP presents almost the same results with PGMCC, although this scenario favors PGMCC in terms of number of receivers in the group and the lack of dynamic competing traffic. It is our assessment that the frequent acker selection, as a result of a dynamically changed network, would reduce PGMCC's performance.

4.2.2. Responsiveness to losses. In this simulation we investigate PGMCC behavior and compare it with ASMP in a network topology with uncorrelated losses.

We use a simulation scenario with 100 multicast receivers behind independent links with random loss ratio of 1% (figure 19). An additional link with same features is used for TCP traffic. In the beginning of the simulation 10 multicast receivers join the session followed by additional 90 receivers at the 300th simulation second. Figure 20 shows the simulation results of PGMCC. We observe that PGMCC achieves lower throughput than TCP, as PGMCC rate control is based on acker selection (one of the 100 PGMCC receivers). The presence of the additional 90 receivers does not seem, however, to affect PGMCC's performance.

Figure 21 shows simulation results from ASMP. TCP throughput is again higher than ASMP throughput, as TCP has higher responsiveness than the smooth congestion control in ASMP. The multicast sender's transmission rate drops at time 300 when the additional 90 receivers join the session. However, the transmission rate is adjusted again in few seconds to previous rates. The additional 90 receivers do not affect ASMP's performance. We conclude from this medium-scale simulation that PGMCC and ASMP have similar behavior in terms of responsiveness to uncorrelated losses. Much larger scale simulations/experiments are needed to further investigate the impact of losses and the scalability of ASMP

5. Conclusions-Future Work

We have presented a simulation-based comparison of two well-known single-rate multicast congestion control schemes against our proposal. Table 3 summarizes the comparison of ASMP with TFMCC and PGMCC. As the table shows, ASMP has very good performance towards competing TCP traffic, while TFMCC seems to starve the TCP traffic. The TCP-friendliness of PGMCC is very close to ASMP. TFMCC presents high fluctuations of the transmission rate, while ASMP has smooth and steady transmission rates in almost all cases. PGMCC's transmission rate is similar to ASMP in simulations involving mild losses and small number of receivers. ASMP presents high responsiveness to packet losses and adapts very rapidly the changes in the network topology, although the convergence time is higher than that of TFMCC and PGMCC.

ASMP scalability is based on RTCP protocol's limitations in which only a small amount of the session bandwidth (5%) can be used for the dissemination of RTCP sender and receiver reports. In groups with large number of receivers the RTCP report intervals tend to increase so that ASMP cannot function properly. In such cases, additional feedback mechanisms have to be in place to ensure that the sender will receive feedback reports, especially from "slow" receivers, in order to adjust its transmission rate. TFMCC presents high scalability with the use of suppression algorithms that select a group representative. PGMCC's scalability depends on network components. In any case PGMCC requires that all receivers in the group should send NACs so that the PGMCC sender can be able to select the group representative (acker). Apart from the observations made during simulations concerning ASMP behavior, that will lead us to correct identified design flaws, we still need to investigate ASMP's scalability with a large number of receivers. Moreover we will evaluate the effect of ASMP in video quality due to the smooth transmission rate. It is also interesting to implement ASMP in real world experiments and investigate its behavior with real multimedia traffic. Another challenging area is to investigate ASMP's performance in wireless networks.

Finally, all sources, simulation scripts and results are available in [20].

Table 3. Comparison of ASMP and Other Single-Rate Congestion Control Mechanisms

	ASMP	TFMCC	PGMCC
TCP-friendliness	very good	poor	good
Stable transmission rate	very good	no	good
Convergence time	average	good	good
Scalability	average (RTCP based)	very good (group representative)	average (requires NACs from all receivers)
Limitations	no	no	requires support from network devices

6. Acknowledgement

We thank the anonymous MASCOTS 2008 reviewers for their helpful comments.

References

[1] RFC 2357: A. Mankin, A. Romanow, S. Bradner, V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols", June 1998.

[2] J. Pandhye, J. Kurose, D. Towsley, R. Koodli, "A model based TCP-friendly rate control protocol", Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, June 1999.

[3] J. Widmer and M. Handley, "Extending equation-based congestion control to multicast applications," in Proc. of ACM SIGCOMM '01, 2001.

[4] RFC 4654, "On TCP-friendly Multicast Congestion Control (TFMCC)", J. Widmer, M. Handley.

[5] L. Rizzo, "pgmcc: A TCP-friendly single-rate multicast congestion control scheme," in Proc. of ACM SIGCOMM '00, 2000.

[6] http://www.isi.edu/nsnam/ns/

[7] S. Floyd, "Metrics for the Evaluation of Congestion Control Mechanisms", Internet draft-irtf-tmrg-metrics-11, 5 Oct. 2007.

[8] G.-I. Kwon, J. Byers, Smooth multirate multicast congestion control, in: Proceedings of IEEE INFOCOM, 2003.

[9] J. Li, S. Kalyanaraman, Generalized multicast congestion control, in: Proceedings of 5th COST 264 International Workshop on Network Group Communications (NGC 2003) and ICQT, 2003.

[10] RFC 3448, M. Handley, S. Floyd, J. Padhye, J. Widmer, "TCP Friendly Rate Control (TFRC)", Network Working Group, January 2003.

[11] Smith, H., Mutka, M., Rover, D. A Feedback based Rate Control Algorithm for Multicast Transmitted Video Conferencing, Accepted for publication in the Journal of High Speed Networks.

[12] Sisalem D., Wolisz A., "LDA + TCP - Friendly Adaptation: A Measurement and Comparison Study," in the 10th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'2000), June 25 - 28, 2000, Chapel Hill, NC, USA.

[13] RFC 3550, "RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, July 2003.

[14] L. Jiang, M.Yuksel, S. Kalyanaraman, "Explicit rate multicast congestion control", Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 50, Issue 15, (October 2006), Pages: 2614 – 2640, 2006.

[15] RFC 3208, "PGM Reliable Transport Protocol Specification", T. Speakman, et. al., December 2001.

[16] C. Bouras, A. Gkamas, G. Kioumourtzis, "Extending the Functionality of RTP/RTCP Implementation in Network Simulator (ns – 2)" First International Conference on Simulation Tools and Techniques for Communications, Networks, and Systems, Marseille, France, 3 - 7 March 2008, (to appear).

[17] Gharai, L., "RTP with TCP Friendly Rate Control", draft-ietf-avt-tfrc-profile-07 (work in progress), March 2007.

[18] C. Bouras, A. Gkamas, G. Kioumourtzis, "Smooth Multicast Congestion Control for Adaptive Multimedia Transmission", NGI 2008, Krakow, Poland, 28030 April 2008.

[19] http://www.informatik.uni-mannheim.de/pi4/projects/congctrl/tfmcc/installation.html (accessed on 5 January 2008).

[20] http://ru6.cti.gr/ru6/ns_rtp_home.php.