MULTICAST FLOW CONTROL IN PRIORITY-BASED IP NETWORKS *

Ashraf Matrawy Ioannis Lambadaris
Broadband Networks Laboratory
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
{amatrawy,ioannis}@sce.carleton.ca

Abstract

In this paper, we present simulation results of our research work in the area of multicast congestion control for video applications. This work is based on our proposal [1, 2] of the use of a new form of network support to multicast congestion control. Our approach is to bring together simple, loosely-coupled, router mechanisms and adopt them for the multicast case. We use a variant of the Explicit Congestion Notification (ECN) mechanism to notify the sender of a multicast session of potential network congestion. We develop an end-to-end multicast system using this congestion control scheme. This included the development and tuning of an elaborate rate adaptation algorithm that operates at the sender. We build this system on top of a network that applies packet priority-dropping to insure providing minimum video quality during persistent congestion. In particular, the work is targeted at the IETF Assured Forwarding (AF) services networks. We show that the synergy of these mechanisms deals with the heterogeneity of receivers in a scalable manner and avoids the major problems of earlier approaches. We believe that this work is of great value to multicasting applications in future QoS-aware networks.

Keywords: *Multicast, Congestion, Flow, Assured Forwarding*

1 Network model

We start by briefly describing the network model we consider for our work. The algorithm is targeted at real-time multicast applications. These applications are assumed to operate in IP networks that support priority-dropping as a means of providing different classes of services to its users. Priority-dropping is the process of packet dropping during congestion by the routers based on a priority level assigned to the packet by its sender. Routers achieve that by employing active queue management techniques that recognizes packet priorities and enqueue, dequeue, and drop packets based on this priority. Random Early Detection (RED) is a widely used active queue management technique [3]. Our network models assumes that routers support RED with In/Out bits (RIO) [4] for providing service differentiation. In our work, we compare different options for selecting the queue management technique. We will discuss that later. Our selection of RIO makes the algorithm suitable for the proposed *Diffserv* Assured Forwarding (AF) service [5]. RIO queues maintain a different set of parameters for each priority level and treat each of these levels as a different virtual queue. We also assume that routers can send messages upstream to the sender with information about the router's congestion status.

2 End-to-end architecture

We build an end-to-end architecture on top of the network model we describe in Section 1. The results we present in this letter are based on testing the algorithm in the context of multicasting MPEG4 encoded real-time video. We send MPEG4 packets as *one* multicast group. These packets are marked with different priority levels by the rate adaptation algorithm at the sender. The algorithm decides how much is the total sending rate and the percentage of the packets marked with each priority level. These decisions are based on the congestion status reported to the sender by the different routers in the network. The algorithm tries to always match the rate for the high priority (most important) packets with the capacity of the slowest receiver. At lower priority levels, rates can be higher than receivers capacities as the packets will be dropped by the routers when they are not needed.

*This research was funded in part by grants from: Communications and Information Technology Ontario (CITO), Natural Sciences and Engineering Research of Canada (NSERC), Mathematics of Information Technology and Complex Systems and (MITACS)

CCECE 2003 - CCGEI 2003, Montreal, May/mai 2003
0-7803-7781-8/03/$17.00 ©2003 IEEE

3 The rate adaptation algorithm

3.1 The rate adaptation equation

Assume that MPEG4 traffic is generated at the source and divided into L layers marked with L different priorities.[1] Also we assume that this is the number of different priorities (and hence virtual queues) that RIO can recognize at the routers. Let $R_i(t)$, $1 \leq i \leq L$, be the rate (in packet/sec) of layer i at the source at time t. We also consider

$$P_i(t) = P_i^{Max}(t) + P_i^{Send}(t)\ P_i^{MinMax}(t)$$

where $P_i(t)$ is the probability that virtual queue i will generate a feedback message at time t, and at time t

$$\begin{aligned} P_i^{Max}(t) &= \text{Prob}\{\text{QueueSize}(i) \geq max\} \\ P_i^{MinMax}(t) &= \text{Prob}\{min \leq \text{QueueSize}(i) \leq max\} \\ P_i^{Send}(t) &= \text{Prob}\{\text{Send feedback message} \\ &\qquad \mid min \leq \text{QueueSize}(i) \leq max\} \end{aligned}$$

We derived $P_i(t)$ from the specification of Backward ECN (BECN) [6]. Considering the changes from *old* to *new* values of $R_i(t)$, and $P_i(t)$ in a small interval Δt, we use the following equation to update the rate $R_i(t)$:

$$R_i{}^{new} = R_i{}^{old}(1 - \alpha_i \Delta P_i), \quad 0 < \alpha_i < 1 \qquad (1)$$

where

$$\Delta P_i = P_i{}^{new} - P_i{}^{old}$$

The rationale behind using this equation is to be always changing $R_i(t)$ in the opposite direction of change of $P_i(t)$ with a step α_i.

We change α_i to control how much $R_i(t)$ changes in reaction to changing network conditions. $|\Delta P_i|$ can assume values between 0 and 1. At these extreme values, changes in R_i^{new} can be either very high (100%) or no change at all (0%). We select $\alpha_i = C_i\sqrt{|\Delta P_i|}$ where C_i is constant at layer i. This sets the maximum rate change to C_i. The choice of square root function was motivated by our design goal of being able to react to very small changes of network conditions ($|\Delta P_i| \leq 0.1$) as the square root of these small values is greater than the actual value ($\sqrt{x} > x$; $|x| \leq 1$). This helps to react to congestion while it is developing. Through simulations, we found that operating with values of C_i ranging from 0.05 to 0.25 keeps the system stable. A value of 0.1 at the high priority layer gave the best performance based on the criteria of 1) minimum packet loss ratio in the high priority layer and 2) matching of the source's rate to receivers'bandwidth capacities.

[1] We use the term layers to describe the different priority levels of packets but we still send all of them in one multicast stream

Equation (1) is subject to the constraints:

$$\begin{aligned} R_i^{min} \leq &\quad R_i &\leq R_i^{max} \\ R^{min} \leq &\quad \textstyle\sum_{i=1}^{L} R_i &\leq R^{max} \end{aligned}$$

The values of R_i^{min}, R_i^{max}, R^{min}, and R^{max} depend on the limitations imposed by the video encoder and on the outgoing link speed.

3.2 Round-Trip Time (RTT)

Routers send feedback messages to the sender with values of P_i^{new} that indicate the congestion status of the routers. The sender will evaluate the feedback from all routers every Δt and decide on a new rate R_i^{new}. The value of Δt will depend of the sender's estimation of the Round-Trip Time (RTT) from the routers that send the feedback information. We select the RTT value that corresponds to the router that has the worst situation at the high priority layer. That is the router with $Max(P_i^{new})$ in its feedback message.

3.3 Feedback suppression

The value of $P_i(t)$ will be estimated by routers and sent back to the sender. To reduce feedback, routers will send feedback messages with a probability instead of sending a feedback message for every packet that causes a problem. From simulations, sending 2% to 5% of the feedback messages kept feedback volume reasonable and in the same time kept the sender responsive to changes in network conditions.

3.4 Calculation of probabilities

The quantities $P_i^{Max}(t)$ and $P_i^{MinMax}(t)$ are calculated using real-time measurements from the network rather than being based on an analytical model. The reason for this is that in the general case where all kinds of traffic flows are coming into the routers queues, it is very hard to assume a certain model for the input traffic.

We bias the probability estimation by giving more weight to newer values to make the estimate a better representative of the current state of the network. We observe the probability at each virtual queue i in k subsequent intervals and give them different weights w_i, $1 \leq i \leq k$. To calculate $P_i(t)$ (whether $P_i^{Max}(t)$ or $P_i^{MinMax}(t)$) at the end of an interval m we use:

$$P_i(m) = \frac{\sum_{j=1}^{k} w_j\ P_{m-j}}{\sum_{j=1}^{k} w_j} \qquad (2)$$

The choice of a value for k is important. Large value of k will result in a smooth estimation of $P_i(m)$. In the same time, values of w_i should give more weight to more recent measurements so that $P_i(m)$ reflects sudden changes in network conditions. $P_i^{Send}(t)$ is calculated using the method in [3]. It depends on the average queue size and on the RIO parameters.

3.5 The algorithm

REPEAT every RTT
    REPEAT for every layer i
        If Nofeedback
          increase R_i by 1%
        else If $(\Delta P_i > 0)$
          reduce R_i using Eqn. (1)
        else If $(\Delta P_i < 0)$
          If (i NOT highest priority layer)
            increase R_i using Eqn. (1)
        else If $(\Delta P_i = 0)$
          If $P_i > 0.9$
            reduce R_i by 3%
    END REPEAT for every layer i
END REPEAT every RTT

3.6 Changing the equation parameters

The value of P_i^{new} will be changed every Δt. There are different criteria that we consider for changing these values:

1. *At the highest priority layer, take the maximum:* In this case, we set the value of P_1^{new} to the maximum value received during Δt. This will results in accommodation of the router will the worst congestion situation. This is done subject to the constraint $R_1(t) \geq R_1^{min}$ to avoid the problem where a slow portion of the receivers drag the sending rate down dramatically[2].

2. *At lower priority layers, take the minimum* In this case, we set the value of P_i^{new} to the minimum value received during Δt. This results in maximizing $R_i(t)$ at layer i subject to $R_i(t) \leq R_i^{max}$. The reason behind this is to let receivers with extra available bandwidth utilize their links. Slow receivers will have these packets dropped by their routers.

4 Simulation

We carried out simulations using *ns-2* [7]. We simulated the topologies in Fig. 1 with two priority levels. The results we show are for simulations that are 300 seconds long. We use the decoupled version of RIO where the average length of each virtual queue is based on the number of packets belonging to this queue (and hence its priority level). We use $C_i = 0.1$ for both priority levels ($i = 1, 2$).

[2]This is known as the "drop to zero" problem.

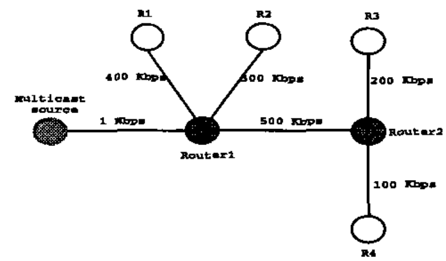


Figure 1: Topology of the bandwidth adaptation test

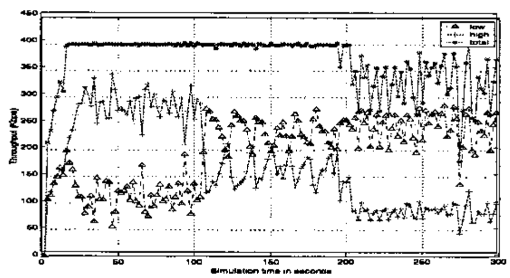
In this experiment, we test how the rate adaptation algorithm adapts to changing network condition which in this case is the changing receiver membership of a multicast session. The topology of Fig. 1 is used to run the following scenario. R1 and R2 join the MPEG4 multicast session that is conducted by the Multicast Source. After a 100 seconds, R3 joins the session, and after 200 seconds, R4 joins the session as well. The bandwidth of each link is as shown in the figure.

Fig. 2 shows the throughput of each receiver at each priority level as well as the total throughput. In the interval [0,100), R1 and R2 receive between 250Kbps and 300Kbps of high priority to accommodate R1 and R2 gets the rest of its 400Kbps (a little more than 100Kbps) in lower priority. Both R3 and R4 get nothing until R3 joins the session after 100 seconds which lowers the rate at the high priority to an average of 170Kbps to accommodate R3. Both R1 and R2 still utilize their bandwidth with more of the lower priority traffic. After 200 seconds, R4 joins the session, and again the sender adjusts the high priority rate to 80Kbps and allows the other receivers more lower priority to fully utilize their links.

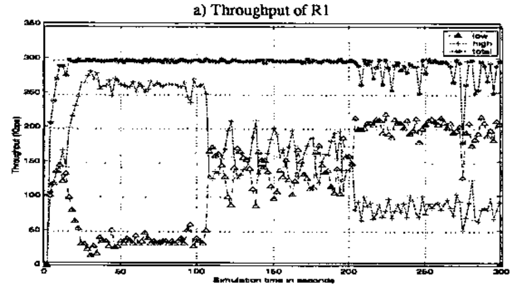
5 Conclusion

We have presented a rate adaptation algorithm for multicast sources in priority-based networks. It is useful for *Diffserv-like* IP networks. It enables users with different bandwidth capabilities to receive the same video multicast in different qualities. This differentiation is based on encoding the video into two levels of priorities. One important basic stream and one (or more) enhancement stream(s). The algorithm tries to match the rate for the important level with that of the slowest receiver. The enhancement level is increased to enable other receivers get better quality.

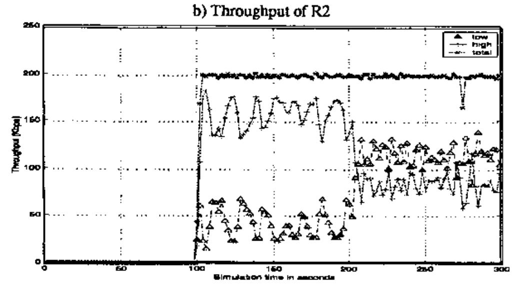
The limitation of this approach is that receivers should at least have their bandwidth grater than R_1^{min} (minimum rate at the high priority layer). We have presented a subset of our results, for a comprehensive simulation study of this approach the reader should refer to [8].



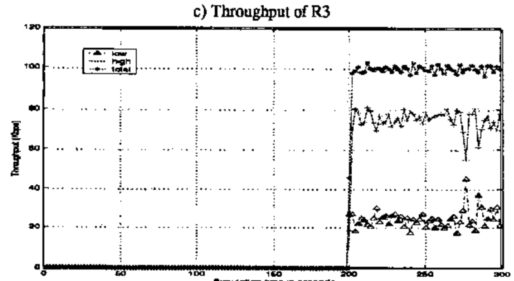
a) Throughput of R1



b) Throughput of R2



c) Throughput of R3



d) Throughput of R4

Figure 2: Bandwidth adaptation to changing session membership

References

[1] A. Matrawy and I. Lambadaris, "A Rate Adaptation Algorithm for Multicast Sources in Priority-based IP Networks," *IEEE Communications Letters*, vol. 7, no. 2, February 2003.

[2] A. Matrawy and I. Lambadaris, "A Real-time Video Multicast Architecture for Assured Forwarding Services," *accepted for publication in IEEE Transactions on Multimedia.*

[3] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. on Networking*, pp. 397–413, August 1993.

[4] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Trans. on Networking*, vol. 6, no. 4, pp. 362–373, August 1998.

[5] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," *RFC 2597*, June 1999.

[6] J. Hadi Salim, B. Nandy, and N. Seddigh, "A Proposal for Backward ECN for the Internet Protocol (IPv4/IPv6)," *Internet Draft, draft-salim-jhsbnns-ecn-00.txt.*

[7] S. McCanne and S. Floyd, "ns Network Simulator," *http://www.isi.edu/nsnam/ns/.*

[8] A. Matrawy and I. Lambadaris, "Simulation Results for Multicast Video for Assured Forwarding Networks," *Carleton University Technical Report SCE-02-11*, October 2002.