A Rate Adaptation Algorithm for Multicast Sources in Priority-Based IP Networks

Ashraf Matrawy, *Associate Member, IEEE*, and Ioannis Lambadaris

***Abstract*—This letter presents a new rate adaptation algorithm for multicast sources that operate in priority-based IP networks. The algorithm represents the flow control component of our work on an architecture for video multicasting in priority-based IP networks. We show (through simulation results) that the algorithm meets our design goal of satisfying the quality-of-service (QoS) requirements of different video multicast receivers which have different networking capabilities.**

***Index Terms*—Assured forwarding, , congestion, multicast.**

I. Network Model

WE START by briefly describing the network model we consider for our work. The algorithm is targeted at real-time multicast applications that operate in IP networks that support priority-dropping as a means of providing different classes of services to its users. Our network models assume that routers support RED with in/out bits (RIO) [1] for providing service differentiation using priority dropping. Our selection of RIO makes the algorithm suitable for the proposed *Diffserv* Assured Forwarding (AF) service [2]. 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 ECN (Explicit Congestion Notification) messages upstream to the sender with information about the router's congestion status.

II. End-To-End Architecture

We build an end-to-end architecture on top of the network model described in Section I. The results presented 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. This congestion status is represented by the probability of the router sending a feedback message as described by $P_i(t)$ in Section III. The algorithm always tries to set the rate for the high priority (most important) packets to accommodate the router with the worst congestion. That is the router with $\mathrm{Max}\{P_1(t)\}$, where $i = 1$ refers to the highest priority layer. 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.

III. The Rate Adaptation Algorithm

A. 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) recognized at the routers. Let $R_i(t)$, $1 \leq i \leq L$, be the rate (in packets/s) of layer i at the source at time t.

We also consider

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

where $P_i(t)$ is the probability that virtual queue i will generate a feedback message at time t. Also at time t we have

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

We derived $P_i(t)$ from the specification of BECN (Backward ECN) [3]. 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 $R_i(t)$:

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

where $\Delta P_i = P_i^{\mathrm{new}} - P_i^{\mathrm{old}}$. The rationale behind using this equation is to always change $R_i(t)$ 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 no change at all (0%) or very high (100%). We select $\alpha_i = C_i \sqrt{|\Delta P_i|}$ where C_i is a constant for layer i and $0 < C_i < 1$. This sets the maximum rate change to $C_i R_i$ at layer 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 (when $|\Delta P_i| \leq 0.1$) as the square root of these small values is greater than the actual value ($\sqrt{x} > x$; $|x| \leq 1$). This ensures reacting to congestion while it is developing. Equation (1) is subject to the following constraints:

$$R_i^{\min} \leq R_i \leq R_i^{\max}$$

$$R^{\min} \leq \sum_{i=1}^{L} R_i \leq R^{\max}$$

Manuscript received August 21, 2002. The associate editor coordinating the review of this paper and approving it for publication was Prof. N. Ghani.

The authors are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: amatrawy@sce.carleton.ca).

Digital Object Identifier 10.1109/LCOMM.2002.808378

[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

where $R_i^{\min}$ and $R_i^{\max}$ are the value limits of the rate at layer i, respectively, and $R^{\min}$ and $R^{\max}$ are the limits for the total source rate. These values depend on the limitations imposed by the video encoder and on the outgoing link speed.

B. 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 congestion at the high priority layer. That is the router with $\text{Max}(P_i^{\text{new}})$ in its feedback message.

C. Feedback Suppression

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%–5% of the feedback messages kept feedback volume reasonable.

D. Calculation of Probabilities

The quantities $P_i^{\text{Max}}(t)$ and $P_i^{\text{MinMax}}(t)$ 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 used the scheme presented in [4] for measuring loss intervals. The probability is observed at each virtual queue i in k subsequent intervals and give these intervals different weights w_i, $1 \leq i \leq k$. To calculate $P_i(t)$ (whether $P_i^{\text{Max}}(t)$ or $P_i^{\text{MinMax}}(t)$) at the end of an interval m we use

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

We experimented with the values of k and w and for most or our simulations, we used $k = 10$, and $w = \{4,4,4,4,4,2,2,2,1,1\}$. These values result in smooth changes in values of P_i^{new}. $P_i^{\text{Send}}(t)$ is calculated using the method in [5]. It depends on the average queue size and on the RIO parameters.

E. Changing the Equation Parameters

The value of P_i^{new} will be changed every Δt.

- *At the highest priority layer* $(i = 1)$, take $\text{Max}(P_1^{\text{new}})$. This will result in accommodation of the router with the worst congestion situation.
- *At lower priority layers* $(i > 1)$, take $\text{Min}(P_i^{\text{new}})$ This results in maximizing $R_i(t)$ at layer i.

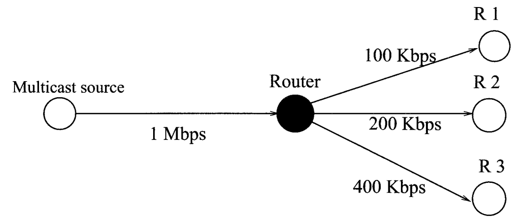

Fig. 1. Simulation setup.

F. The Algorithm for Changing R_i

```
REPEAT every RTT
 REPEAT for every layer i
  If Nofeedback
   increase R_i by 1%
  else If (ΔP_i > 0)
   reduce R_i using (1)
  else If (ΔP_i < 0)
   If (i NOT highest priority layer)
    increase R_i using (1)
  else If (ΔP_i = 0)
   If P_i^new > 0.9
    reduce R_i by 3%
 END REPEAT for every layer i
END REPEAT every RTT
```

Note that the values of 1% and 3% are chosen to conservatively increase/decrease the rate as at this point we can not be sure exactly which direction ΔP_i is moving. Note also that the rate of the highest is increased only when there is no feedback to make sure that it is not increased beyond slow receivers capacities.

IV. Simulation

We carried out simulations using *ns-2* [6]. We simulated the topology in Fig. 1 with two priority levels. The MPEG4 traffic used in the simulation is based on a traffic model we developed for MPEG4 [7]. The results we show are for simulations that are 300 seconds long. We use the de-coupled 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 $(i = 1, 2)$. The goal of these simulation is to check whether the algorithm will match the rate of the highest priority layer with that of the slowest receiver and whether it will allow other receivers to get more traffic in the lower priority layer.

From the results in Fig. 2 we can see that the rate of the high priority packet quickly settles at a value slightly less that 100 Kbps which is the rate for R1 (part R1-A). Both R2 and R3 get the same rate but they get higher rate at lower priority layer (parts R2-A and R3-A). In parts R1-B, R2-B, and R3-B of the figure we note the difference in the loss ratio of the three receivers at each of the priority levels. This results in different qualities of received video.

We simulated more complex topologies [8]. The results from these simulations show a performance that is consistent with the results presented in this letter. A modification that we apply in

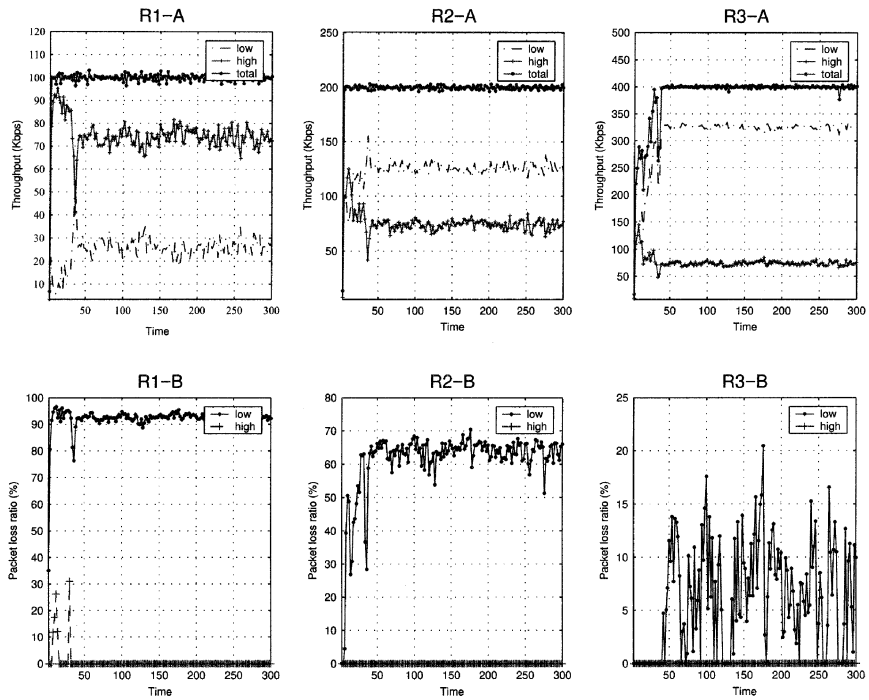

Fig. 2. Simulation results.

[8] is to drop *all* lower priority packets that is going to a specific congested receiver in the case of continuous high loss rate. The MPEG4 decoder might not be able to benefit from this lower priority stream with this continuous high loss.

V. 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. 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).

REFERENCES

[1] D. Clark and W. Fang, “Explicit allocation of best effort packet delivery service,” *IEEE/ACM Trans. Networking*, vol. 6, pp. 362–373, August 1998.

[2] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB Group,”, RFC 2597, 1999.

[3] J. H. Salim, B. Nandy, and N. Seddigh, “A proposal for Backward ECN for the Internet Protocol (IPv4/IPv6), Internet Draft,”, draft-salim-jhs-bnns-ecn-00.txt.

[4] J. Widmer and M. Handley, “Extending equation-based congestion control to multicast applications,” in *Proc. SIGCOMM*, Aug. 2001.

[5] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. Networking*, pp. 397–413, Aug. 1993.

[6] S. McCanne and S. Floyd. ns Network Simulator. [Online]. Available: http://www.isi.edu/nsnam/ns/

[7] A. Matrawy, I. Lambadaris, and C. Huang, “MPEG4 traffic modeling using the transform expand sample methodology,” in *Proc. 4th IEEE Int.Workshop on Networked Appliances*, Jan. 2002, pp. 249–256.

[8] A. Matrawy and I. Lambadaris, “Simulation results for multicast video for assured forwarding networks,”, Carleton Univ.Tech. Rep. SCE-02-11, 2002.