Congestion Control for Fair Resource Allocation in Networks with Multicast Flows *

Supratim Deb† and R. Srikant‡
University of Illinois at Urbana-Champaign
{deb,rsrikant}@uiuc.edu

Abstract

We consider the problem of congestion control in networks which support both multirate multicast sessions and unicast sessions. We present a decentralized algorithm which enables the different rate-adaptive receivers in different multicast sessions to adjust their rates to satisfy some fairness criterion. A one-bit ECN marking strategy to be used at the nodes is also proposed. The congestion control mechanism does not require any per-flow state information for unicast flows at the nodes. At junctions nodes of each multicast tree, some state information about the rates along the branches at the node may be required. The congestion control mechanism takes into account the diverse user requirements when different receivers within a multicast session have different utility functions, but does not require the network to have any knowledge about the receiver utility functions.

Keywords: Multicasting, Congestion Control, Utility Function, ECN marking

1 Introduction

In the modern day Internet there is a demand for multicast, especially in applications where communication is required in a group. Multicast traffic can cause more congestion related damage than unicast traffic due to the fact that a single multicast flow can be distributed along a large multicast tree reaching throughout the entire Internet [15]. Further, many multicasting applications

*Research supported by DARPA Grant F30602-00-2-0542 and NSF Grants ANI-9813710 and NCR-9701525

†Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, Urbana, IL 61801

‡Coordinated Science Laboratory and the Department of General Engineering, 1308 W. Main Street, Urbana, IL 61801; Ph: 217-333-257; Fax: 217-244-1642

like video-conferencing need a large amount of bandwidth. An important issue in the deployment of multicast is the impact of multicast traffic on the global Internet when there is congestion. For unicast traffic, congestion control is implemented within the transport layer protocol TCP. Hence, there is a growing need for a unified congestion control mechanism for networks with unicast and multicast traffic, which is easy to implement and can coexist in a fair and friendly manner with TCP based congestion control for unicast sessions.

Due to the diverse characteristics and requirements of the different receivers within a multicast group, it is sometimes desirable to have multicast sessions in which different receivers receive data at different rates. If the different receivers in a multicast group can have different rates, it is called multirate-multicast. If all the receivers of a multicast group have the same rate it is called unirate-multicast. In unirate multicast, the sender transmits data at a rate requested by the slowest receiver in the multicast group. In this paper we concern ourselves with multirate-multicast. In multirate multicast, a receiver can adapt its rate based on the congestion level solely on the path. Layered video [13] multicast is one such application.

We address the issue of congestion control in networks where resources are shared by unicast and multirate multicast sessions. In networks with only unicast sessions, the total flow rate through a link is the sum of the rate of the unicast sessions. However, if there are multicast sessions passing through a link, the multicast session rate is the maximum of the rate of all downstream receivers of the multicast session. Each multicast session is thus characterized by a number of virtual sessions [16], where each receiver of a multicast session can be thought as a virtual session. As a result, unless a virtual session has a rate equal to the maximum of the rates of all the virtual sessions of a session through that link, that virtual session does not directly contribute to the congestion in that link. Thus, any congestion control scheme which treats each of the virtual sessions like separate unicast sessions is not appropriate for such a framework. Such a scheme may be unfair to the virtual sessions with rate less than the multicast session rate. Further, since there may be receivers with contrasting requirements, source based congestion control where the source updates the rates based on the congestion signals (losses or ECN marks) received may not be appropriate for networks with multicast sessions. While fairness to competing users is a desirable issue in the design of a congestion control algorithm, we also have to keep in mind the scalability of the scheme. The TCP flow control mechanism, which is the dominant rate control mechanism for unicast sessions in the Internet, works well with minimal information from the network and the routers do not need to keep track of per flow information. Such a scalable solution is desirable with the rapidly growing size of Internet. Our goal in this paper is to provide a framework to design a congestion control scheme with multirate-multicast and unicast sessions present in the network which has the desirable features as mentioned.

Clearly, fairness among competing users is an important issue in designing resource allocation schemes. Two notions of fairness that are commonly used are *max-min fairness* and *proportional fairness*. These intuitive notions of fairness can be generalized in an elegant manner using the idea of utility functions [9, 17]. Recently, congestion control in networks with only unicast sessions based on utilities of different users has gained considerable interest [8]. Such a framework encompasses the diverse requirements of the users. The utility function of a flow denotes the utility of a given amount of bandwidth to the user. The utility function is increasing and concave to capture the fact that an additional amount of bandwidth is more useful if the bandwidth already available is less. The TCP flow control mechanism has been shown to be a special case with an appropriate utility function [10]. We use the notion of utility functions for different virtual sessions to devise a congestion control mechanism which is fair to the requirements of all the virtual sessions and the unicast sessions.

1.1 Related Work: A Brief Survey

Considerable research has been devoted to the multicast congestion control problem. There are multirate reception protocols proposed for the reception of layered video [16]. The rate control mechanisms for such systems work by adding or dropping layers depending on the level of congestion. In this context there are two categories of layered multicast that were proposed for congestion control, the *static* layering scheme and the *dynamic* layering scheme. In the static layering scheme, the sending rate of any layer remains fixed with time [16, 12]. Several dynamic layering schemes have been proposed for the purpose of congestion control where rates of different layers can be modified depending on the congestion level [23, 14]. Many of these do not deal with the issue of a unified congestion control mechanism which is fair to all the unicast and the multicast sessions. In [5], the authors have considered the problem of designing decentralized algorithms to achieve max-min fairness in multicast networks. They have shown that this can be achieved by a simple one bit marking strategy. Similarly, in [20], an algorithm has been proposed for computing max-min fair rates in a multicast network. In [21], the authors have extended the algorithm for the computation of max-min fair rates when discrete bandwidth layers are available as in the case of layered video. In [6], an optimization based rate control mechanism, based on a utility maximizing framework, has been proposed for multirate multicast sessions, which tries to solve the dual of a convex program formulation of the problem. Even though this takes into account the heterogeneous requirements of the multicast receivers within and across different sessions, this requires per-flow state information at the nodes. It is also not clear how this can co-exist with TCP based unicast congestion control, which can be viewed as a solution to the primal problem [10, 8]. Congestion control schemes for unirate-multicast has also been proposed in the literature. In [4], the author has proposed a multicast congestion control scheme for the Internet. It has also been argued that a naive unicast window based congestion control scheme which treats each virtual session like a unicast session does not perform well. More recently, a rate control mechanism for unirate-multicast using an optimization framework has been proposed in [22].

1.2 Main Contribution and the Organization of the Paper

Our main contributions are as follows. We propose a class of congestion control algorithm for rate adaptive multirate multicast sessions, keeping in mind the heterogeneity of the multicast receivers. Our algorithms require a packet marking capability by the network without keeping per-flow state information about the unicast sessions and requires minimal per flow information about the virtual sessions (which may be required for the successful implementation of multirate-multicast). Such a scheme is well suited for Internet. Explicit Congestion Notification for flow control was first proposed in [3] and such a mechanism is highly desirable if we want the network to operate at low loss. We propose an easy-to-implement one-bit ECN marking scheme to implement the congestion control algorithm. The receivers adapt their rates accordingly based on the marks received. For layered video, rate adaptation can be achieved by adding or dropping more layers. We also prove that the class of schemes we proposed is stable under a mild set of assumptions. We also show through simulations that our scheme can co-exist in a friendly manner with TCP flow control mechanism for unicast sessions.

The rest of the paper is organized as follows. In Section 2 we discuss our model and state the assumptions and notations used throughout the paper. In Section 3, we develop a heuristic for a set of rate control equations for network with multicast sessions. We propose a class of congestion control algorithms and discuss an easily implementable one-bit marking scheme in Section 4. We also discuss in this section how the algorithms can co-exist with TCP based congestion control for unicast sessions. In Section 5 , we prove the stability of the class of multicast rate control schemes. We discuss some of the implementation related issues, in the context of a layered multicast scenario and a distributed implementation in Section 6. We provide various numerical and simulation results in Section 7 to show the effectiveness of the congestion control mechanism proposed in the paper. Conclusions and future research directions are provided in Section 8.

2 Model and Assumptions

Consider a network with set L of links and let C_l be the capacity of link l. Let S be the set of sessions and R_s be the set of receivers corresponding to any session $s \in S$. R_s is a singleton set for the unicast sessions and so we include the unicast sessions within the same framework. The receivers of a multicast group can be viewed as virtual sessions corresponding to that particular multicast session. Henceforth we use the terms virtual session and receiver interchangeably. We use the notation (s, r) to denote a virtual session corresponding to session s. Let L_{sr} be the set of all links in the route of a virtual session (s, r). Let S_l be the set of all sessions passing through link l and let V_{sl} be the set of all virtual sessions of the session s using link l. We associate a utility function to each virtual session. Let $U_{sr}(x)$ be the utility function of the virtual session (s, r). Also, associated with each link, let there be a function $p_l(x)$ which denotes the fraction of packets marked whenever there is congestion in that link when the total flow rate in that link is x. The details of how the marks are distributed among the virtual sessions sharing the link will be provided later. Let $x_{sr}(t)$ denote the rate at which the virtual session (s, r) sends data at time t. The total flow in link l at time t is given by $\sum_{s\in S_l}(\max_{(s,r)\in V_{sl}}(x_{sr}(t)))$. We make the following assumptions.

Assumption 1. *(Marking functions, Utility functions, Multicast Tree)*

1. *The functions $p_l(x)$ are monotonically non-decreasing functions of the total flow in the link x.*
2. *The functions $U_{sr}(x)$'s are strictly concave, continuously differentiable increasing functions of x.*
3. *We also assume that the multicast tree associated with each multicast session does not change with time.* □

The objective is to find rates for which the total utility is maximized. Thus, the problem can be posed as the following convex program.

$$\max \sum_{s\in S} \sum_{r\in R_s} \Delta_{sr} U_{sr}(x_{sr})$$

subject to

$$\begin{aligned} \sum_{s\in S_l} \max_{(s,r)\in V_{sl}} x_{sr} &\le C_l, && \forall l \in L, \\ x_{sr} &\ge 0, && \forall s \in S,\ r \in R_s. \end{aligned} \tag{1}$$

Here $\Delta_{sr} > 0\ \forall (s, r)$, and can be viewed as weights associated with each of the sessions. The above maximization problem has a concave objective function and a convex constraint set and thus can be shown to have a unique optimal solution. For a network consisting of only unicast sessions, it has been shown by Kelly *et. al* [8] that, if user s adapts the rate according to

$$\frac{dx_s(t)}{dt} = \kappa(\Delta_s - \frac{\beta}{U_s'(x_s(t))}\sum_{l\in L_j} p_l(\sum_{s\in S_l} x_s(t))), \quad (2)$$

then the rates $x_s(t)$'s converge to a unique stable point which maximizes the following revised expression.

$$V_u(\underline{x}) = \sum_{s\in S} \Delta_s U_s(x_s) - \beta\sum_{l\in L}\int_0^{\sum_{m\in S_l} x_m(t)} p_l(u)du \quad (3)$$

Thus this algorithm solves a relaxation of the optimization problem (1) and hence can be viewed as a penalty function formulation of the non-linear program (1) with only unicast sessions. It has been shown in [10] and [7] that the TCP based congestion control mechanism can be thought as a special case of (2).

3 Multicast Rate Control: Issues and Solution Approach

The max functions in the total rate of a multicast session passing through a link poses a challenge in studying the rate control equations and their stability. To understand the problem posed by the max functions, first note that, in the case of only unicast sessions present in the network the rate control equation given by (2) is essentially of the form

$$\frac{dx_s(t)}{dt} = \frac{\kappa}{U_s'(x_s(t))}\frac{\partial V_u(\underline{x}(t))}{x_s}\ , \quad (4)$$

where $V_u(\underline{x})$ is the revised objective given by (3) which is also a Lyapunov function for the system with only unicast sessions present. However, such an approach does not apply when there are multicast sessions in the network, since the partial derivatives of $V_u(\underline{x})$ with respect to the rates of the virtual sessions are not well-defined due to the non-differentiability of the max function.

One possible approach could be to split the constraints into multiple constraints to make the constraints linear. To see this, suppose a link A has two receivers of a multicast session passing through it with rates x_{01} and x_{02}. Let link A also have a unicast session with rate x_1 passing through it. Note that the link constraint for link A,

$$\max(x_{01}, x_{02}) + x_1 \leq C_A\ ,$$

can be equivalently written as

$$x_{01} + x_1 \leq C_A \quad \text{and} \quad x_{02} + x_1 \leq C_A\ .$$

Here C_A is the capacity of the link. We can thus split the link rate constraint into multiple linear constraints depending on the number of virtual sessions using the link. Suppose that we define a penalty function in a similar vein as in the solution to the unicast problem and define the rate control mechanism of each of the virtual sessions using the partial derivative of the penalty function. It is easy to see that this would require each link to keep track of the rates of the individual virtual sessions (since each virtual session passing through a link will impose an additional constraint for that link) and thus, this approach is not scalable.

In the following, we discuss a possible approach to overcome this problem. We start by approximating the multicast rate control problem so that it is easier to handle and find a stable, decentralized solution for the approximate problem. We will then derive an easy to implement decentralized solution for the exact problem from the solution of the approximate problem.

3.1 Multicast Rate Control Based on a System of Approximate Rate Control Equations

Consider the functions of the form $(\sum_i x_i^n)^{\frac{1}{n}}$. Observe that the function $\max_i(x_i)$ can be approximated by the function $(\sum_i x_i^n)^{\frac{1}{n}}$ for large enough n, when all x_i's are non-negative real numbers. (*i.e.* $(\sum_i x_i^n)^{\frac{1}{n}}$ uniformly converges to $\max_i(x_i)$ as n becomes large). Motivated by this observation, we now consider the following optimization problem, where the max functions in (1) are replaced by their differentiable approximations.

$$\max \ \sum_{s\in S}\sum_{r\in R_s}\Delta_{sr}U_{sr}(x_{sr})$$

subject to

$$\begin{aligned} \sum_{s\in S_l}\Big(\sum_{(s,r)\in V_{sl}} x_{sr}^n\Big)^{\frac{1}{n}} &\le C_l, \quad \forall l\in L, \\ x_{sr} &\ge 0, \qquad \forall s\in S,\ r\in R_s. \end{aligned} \tag{5}$$

We present a set of rate control equations for the convex program (5) and based on this, we will develop a heuristic for the rate control algorithm for a network with multirate multicast sessions.

Towards this end, we consider a penalty function formulation of the convex program (5), in which we try to minimize the following revised expression.

$$V_n(\underline{x}) \;=\; -\sum_{s\in S}\sum_{r\in R_s}\Delta_{sr}U_{sr}(x_{sr}) + \beta\sum_{l\in L}\int_0^{(\sum_{m\in S_l}(\sum_{(m,j)\in V_{ml}} x_{mj}^n)^{\frac{1}{n}})} p_l(u)du\ , \tag{6}$$

where β is a non-zero positive number. Further, consider the following congestion control algorithm to minimize the cost function given by (6).

$$\frac{dx_{sr}(t)}{dt} = \Delta_{sr} - \frac{\beta}{U'_{sr}(x_{sr}(t))}\sum_{l\in L_{sr}}(p_l\left(\sum_{m\in S_l}(\sum_{(m,j)\in V_{ml}} x_{mj}^n(t))^{\frac{1}{n}}\right))G_{l,(sr)}^{(n)}(\underline{x}) \quad (7)$$

$$G_{l,(sr)}^{(n)}(\underline{x}) = x_{sr}^{n-1}\sum_{(s,j)\in V_{s,l}} x_{sj}^n(t))^{\frac{1}{n}-1}, \quad (8)$$

Next we show that the rate control differential equation given by (7) is stable for large but finite n, and the function $V_n(\underline{x})$ provides a Lyapunov function for the system of differential equations described in (7). Here $\underline{x}$ denotes the column vector of rates of all the virtual/unicast sessions.

Theorem 3.1. *The function $V_n(.)$ is a valid Lyapunov function for the system described in (7) and this system is globally asymptotically stable. The stable point is the unique $\underline{x}$ minimizing the strictly convex function $V_n(.)$.*

Proof. If we show $V_n(.)$ is a convex function, then the result follows along the lines of the proof in [8]. The functions $-U_{sr}(.)$'s are strictly convex functions due to the assumption that the utility functions are strictly concave. Thus to show the convexity of $V_n(.)$ it suffices to show the convexity of the second term on the right hand side of the expression of $V_n(.)$. Let

$$h_l(\underline{x}) = \sum_{m\in S_l}(\sum_{(m,j)\in V_{ml}} x_{mj}^n)^{\frac{1}{n}}$$

It is easy to observe that the function $h_l(.)$ is convex. Also, since $p_l(.)$ is a monotonously non-decreasing function, the function $\int_0^x(p_l(u)du)$ is convex in x. So, we can write the following.

$$\begin{aligned}\int_0^{h_l(\underline{y})}(p_l(u)du) - \int_0^{h_l(\underline{x})}(p_l(u)du) &\geq (h_l(\underline{y}) - h_l(\underline{x}))(p_l(h_l(\underline{x})) \\ &\geq (p_l(h_l(\underline{x}))(\bigtriangledown h_l(\underline{x}))^T(\underline{y}-\underline{x})\end{aligned}$$

The second inequality follows from the convexity of $h_l(.)$. From this we can see that the function $\int_0^{h_l(\underline{x})}(p_l(u)du)$ is convex in $\underline{x}$ and so $V_n(.)$ is a convex function. The remaining part of the proof to show that $\dot{V}_n(\underline{x}) \leq 0$ is as given in [8]. □

From the above theorem, the rate control algorithm given by (7) solves a relaxation of the optimization problem (5). The stability of (7) for arbitrarily large n thus motivates us to suggest a set of rate control equations for multicast networks. First, observe the following.

$$\lim_{n\to\infty} G^{(n)}_{l,(sr)}(\underline{x}) = \begin{cases} 0, & \text{if } (s,r) \in V_{sl} \text{ and, } x_{sr} < \max_{(s,j)\in V_{sl}}(x_{sj}) \\ 1, & \text{if } (s,r) \in V_{sl},\ x_{sr} = \max_{(s,j)\in V_{sl}}(x_{sj}), \\ & \text{and } \forall\ (j \neq r, (s,j) \in V_{sl})\ x_{sr} > x_{sj} \\ (\sum_{(s,j)\in V_{sl}} I_{(x_{sr}=x_{sj})})^{-1} & \text{if } (s,r) \in V_{sl},\ x_{sr} = \max_{(s,j)\in V_{sl}}(x_{sj}), \\ & \exists\ (j \neq r, (s,j) \in V_{sl}) \text{ st } x_{sr} = x_{sj} \end{cases}$$

So, the rate control equation for the virtual sessions in a multicast network can be written as follows.

$$\frac{dx_{sr}(t)}{dt} = \Delta_{sr} - \frac{\beta}{U'_{sr}(x_{sr}(t))} \sum_{l\in L_{sr}} p_l \left(\sum_{m\in S_l} \max_{(m,j)\in V_{ml}} x_{mj}(t) \right) I_{(x_{sr}=\max_{(s,j)\in V_{sl}}(x_{sj}))} \frac{1}{\sum_{(s,j)\in V_{sl}} I_{(x_{sr}=x_{sj})}} \tag{9}$$

Clearly, the rate control equation suggests that the ECN marks be sent by a link to a receiver only if that receiver has a maximum rate among all the receivers of its multicast session passing through that link. When more than one multicast receiver has a rate equal to the multicast session rate through the link, the fraction of ECN marks should be split equally among all the multicast receivers with rate equal to the multicast session rate. So, from the point of view of a virtual session, it never needs to react to congestion on those links in which its rate is less than the multicast session rate.

Remark 1. *The rate control mechanism of the virtual session (s,r) can be of the general form*

$$\dot{x}_{sr} = H(x_{sr}) \left[\Delta_{sr} - \beta(U'_{sr}(x_{sr}))^{-1} \sum_{l\in L_{sr}} q_{l(sr)} \left(\sum_{m\in S_l} \max_{(m,j)\in V_{ml}} x_{mj} \right) \right] \tag{10}$$

where $H(.)$ is a continuous and positive function of x_{sr}. In describing our algorithm we have chosen $H(x_{sr}) = 1$. For a unicast session, if $U(x) = -1/x$ and $H(x) = 1$, this approximates the TCP congestion avoidance phase [10].

Remark 2. *Since the function $G^{(n)}_{l,(sr)}(\underline{x})$ does not converge uniformly, the global asymptotic stability of the system of (7) is not sufficient to deduce the stability of the system (9) (see [19]). Nevertheless, (7) helps us to develop a heuristic for the rate control in multicast networks. In Section 5 we prove the stability of the whole class of multicast congestion algorithms. We note that the resource allocation for the various sessions in the steady state is unaffected by the choice of $H(.)$.*

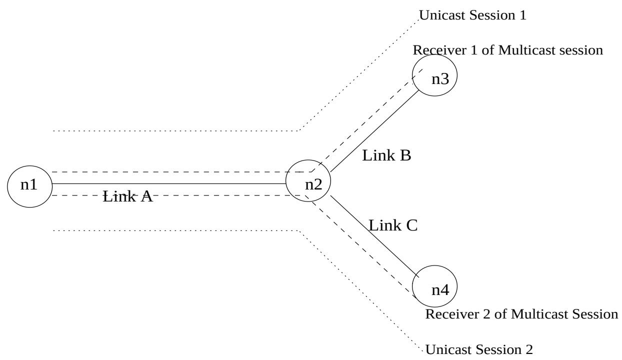


Figure 1: The Y network with two unicast sessions and one multicast session with two virtual sessions.

4 Algorithm for Multicast Congestion Control

From the preceding discussion, the rate control mechanism of the virtual session (s, r) can be written as follows.

$$\dot{x}_{sr} = \Delta_{sr} - \beta(U'_{sr}(x_{sr}))^{-1} \sum_{l \in L_{sr}} q_{l(sr)} \left(\sum_{m \in S_l} \max_{(m,j) \in V_{ml}} x_{mj} \right) \quad (11)$$

Here the function $q_{l(sr)}(.)$ is link l's marking function for the virtual session (s, r). For all the unicast sessions this function is the same as the original marking function $p_l(.)$. For all the virtual sessions belonging to a multicast group the function $q_{l(sr)}(.)$ is same as $p_l(.)$ if that virtual session alone has the highest rate among all the virtual sessions of that multicast group through link l. The function $q_{l(sr)}(.)$ is equal to zero for any virtual session going through that link but having rate less than the multicast session rate through that link. If more than one virtual session has a rate equal to the multicast session rate through the link, then for all those virtual sessions $q_{l(sr)}(.)$ is equal to $\frac{p_l(.)}{K}$, where K is the number of virtual sessions with rate equal to the multicast session rate. In practice, this can be viewed as being equal to $p_l(.)$ with probability $\frac{1}{K}$ and 0 with probability $(1 - \frac{1}{K})$.

A multicast network is comprised of nodes that replicate copies of each packet (depending on the rate requirement of the receiver) of a multicast session before sending to different receivers whose paths are different from that node onwards. These nodes are referred to as "junction nodes" henceforth. For example, in Figure 1, the node common to the links A, B and C is a junction node for the multicast session. Clearly, we want the different multicast receivers of a multicast flow to receive ECN marks at different rates. However, the replication of copies of packets of a multicast flow will mean that if a packet is marked, the mark reaches all the receivers receiving a copy of the packet. To overcome this difficulty we introduce a unmarking scheme at the junction nodes. Before describing the exact algorithm at the links, nodes and the receivers, we first take a look at the discrete time version of (11). Let t_k denote the time at which the k^{th} update takes place. Then discretizing (11) and allowing the possibility that every unicast/virtual session has a minimum rate below which the rates would not be dropped to, we obtain the following.

$$\begin{aligned} x_{sr}(t_{k+1}) &= \max(x_{sr}^{min},\ x_{sr}(t_k) + \\ &\delta(\Delta_{sr} - \frac{\beta}{x_{sr}(t_k)U'_{sr}(x_{sr})}\left(x_{sr}(t_k)\sum_{l \in L_{sr}} q_{l(sr)}(\sum_{m \in S_l} \max_{(m,j)\in V_{ml}} x_{mj}(t_k))\right))) \end{aligned} \quad (12)$$

Here δ is a sufficiently small positive real number. This shows that the updates of the rates of the different multicast sessions has to take into account the sum of the marking probabilities in all the links along the path. Consider a one-bit marking strategy used by the network to convey $\sum_{l \in L_{sr}} q_{l(sr)}$ to the virtual sessions. Suppose link l sets the ECN bit with a probability of $q_{l(sr)}(.)$ for virtual session (s,r). Then, $x_{sr}\sum_{l\in L_{sr}} q_{l(sr)}$ can be viewed as the rate at which the virtual session (s,r) receives marked packets, provided the marking probabilities $q_{l(sr)}$'s are small. This can be seen by noting that a packet of the virtual session (s,r) is marked with probability $1-\prod_{l\in L_{sr}}(1-q_{l(sr)})$, which is approximately $\sum_{l\in L_{sr}} q_{l(sr)}$ if the q_l's are small. Alternatively, $p_l(.)$ can be thought of as the link "price" and q_l's can be conveyed to the end hosts using a scheme like REM [1]. In this paper, we will directly treat p_l's marking probabilities. So the multicast congestion control algorithm becomes the following.

Algorithm 4.1. *(Multicast Congestion Control Algorithm)*

1. ***Link's Algorithm***: *Suppose link l has a marking function $p_l(.)$ which depends on the total flow rate through the link. Then it sets the ECN bit of a packet to 1 with probability $p_l(.)$, if the ECN bit is 0; if the ECN bit is 1 it is left as it is.*

2. ***Node's Algorithm***: *For packets of unicast sessions and packets of multicast sessions for which the node is not a junction node, it transmits each packet along the desired path. If the node happens to be a "junction node" of a multicast tree, then it sends the marks to any of the receivers whose rate is same as the multicast session rate. For all other receivers, the junction node unmarks any packet that is marked, before sending any copy of the packet. This effectively sets the ECN bit to 0 for all but one of the receivers (one of the fastest), for which it retains the mark. The receiver that receives the ECN bit is chosen at random among the receivers with rate equal to the multicast session rate.*

3. ***Receiver Node's Algorithm**: From the marked packets received, each virtual session estimates the rate at which marked packets are received. Specifically, let $m(t_k)$ be the number of marked packets received in interval $(t_k, t_{k+1}]$. Then $m(t_k)/(t_{k+1} - t_k)$ gives an estimate of the rate at which marked packets are received. Then the rate update at the $(k+1)$-th update epoch is thus given by,*

$$x_{sr}(t_{k+1}) = \max(x_{sr}^{min},\ x_{sr}(t_k) + \delta(\Delta_{sr} - \frac{\beta}{x_{sr}(t_k)U'_{sr}(x_{sr}(t_k))}\frac{m(t_k)}{\delta}))\ ,$$

where $\delta = t_{k+1} - t_k$ is the rate update interval. □

In the above congestion control mechanism, the only information required about the multicast flows at the nodes (the junction nodes) is the identification of receivers whose rates are maximum. So, it may look as if the algorithm needs per receiver state information at the junction nodes. However, given a multicast tree with the source at the root and the receivers at the leaves, it is sufficient for a junction node to keep track of its children (in the multicast tree) which are in the path of the receivers with maximum rate. We discuss this in more detail in Section 6.2.

Further, no information about the per flow state of the unicast sessions is available to the network and is also not required by the algorithm. Since in future networks, unicast sessions will still contribute to a very large fraction of the network traffic, the unified congestion control mechanism will not need much additional per flow state information, apart from that required for multicasting.

4.1 TCP friendliness

In the context of modern day Internet, the above rate based congestion control algorithm has to co-exist with the usual window based TCP flow control mechanism for unicast sessions. Thus, we have to ensure that, through an appropriate choice of the utility function for each multicast virtual session, we can design a *TCP-friendly* congestion control algorithm. For a unicast session, a simplified definition of TCP-friendliness is as follows: if a unicast session achieves the same throughput as a TCP session that traverses the same route, then the unicast session is TCP-friendly. This can be achieved by choosing the utility function as $A \arctan(Ax)$ ($A = RTT \times \sqrt{\beta}$) [7] or $-1/x$ [10]. In addition, β has to be chosen as $2/3$ [10] and Δ has to be chosen as $1/r^2$ where r is the RTT (round-trip time).

We use the same parameter and utility function choices for multicast congestion control to ensure the*TCP-friendliness* of our protocol. However, for a multirate-multicast congestion control protocol, some virtual sessions do not cause congestion on all the links on its route. For example, consider a virtual session $(1,1)$ belonging to multicast session 1 that traverses links A, B and C. Suppose another virtual session $(1,2)$ (belonging to the same multicast session 1) traverses links A and B and the steady-state throughput of $(1,2)$ is higher than that of $(1,1)$. Then, virtual session $(1,1)$ does not directly contribute to congestion on links A and B. Thus, from the point of view of TCP-friendliness, it is more meaningful to compare the throughput of $(1,1)$ with that of a TCP session that only traverses link C. With a proper choice of parameters, when no two multicast receivers have same rates, a multicast virtual session will have long run throughput equal to a TCP session traversing through the same set of bottleneck links.

However, the situation is somewhat different when there are multiple virtual sessions of a multicast group having rates equal to the multicast session rate. Note that, in this scenario the marks from a congested link will be shared by all the virtual sessions whose rates are same as the multicast session rate through the link. Thus the comparison of a multicast receiver with that of a TCP session going through the same set of congested links has to be done a bit more carefully. Since any multicast receiver which shares a congested link with other multicast receivers of the same multicast session, only receives a fraction of the marks from that link meant for that multicast session, the multicast receiver rate will be more than that of a TCP session going through the same set of congested links. In the following proposition we give worst case bounds on the total rate of the TCP sessions going through the same set of congested links as the multicast receivers.

Proposition 4.1. *Consider a multicast session with N virtual sessions. Let the steady state rates of the virtual sessions be $x_{01},\ x_{02},...,x_{0N}$. Let $x_1^{(TCP)}, x_2^{(TCP)}, ..., x_N^{(TCP)}$ be the steady state rates of TCP sessions going through the same set of congested links corresponding to the virtual sessions $(0,1)$, $(0,2)$,..., $(0,N)$ respectively. Also, let K be the number of virtual sessions having rates equal to some other virtual session of the same multicast session. Without loss of generality let these K virtual sessions be $(0,1)$, $(0,2)$,..., $(0,K)$. Then,*

$$\frac{1}{\sqrt{K}}\sum_{i=1}^{K}\sqrt{\frac{\Delta_i}{\Delta_{0i}}}x_{0i} + \sum_{i=K+1}^{N}\sqrt{\frac{\Delta_i}{\Delta_{0i}}}x_{0i} \leq \sum_{i=1}^{N}x_i^{(TCP)} \leq (\sum_{i=1}^{N}\sqrt{\frac{\Delta_i}{\Delta_{0i}}})\max_{1\leq i\leq N}x_{0i}\ . \tag{13}$$

Here Δ_{0i} denotes the steady increase of rate for the virtual session $(0,i)$ and $\Delta_i = \frac{1}{RTT_i^2}$ corresponds to TCP session i, and all the multicast sessions have the utility functions of the form $\frac{-1}{x}$ so as to mimic the TCP behavior.

Proof. The proof of this is straight forward and follows from a few simple observations. Let q_i be the steady state fraction of marks received by the virtual session corresponding to multicast receiver $(0,i)$ and also let p_i be the steady state fraction of marks received by TCP session i. The steady state rates are obtained by equating the right hand side of (11) to 0. Thus we have,

$$x_{0i} = \sqrt{\frac{\Delta_{0i}}{\beta q_i}} \quad , \qquad x_i^{TCP} = \sqrt{\frac{\Delta_i}{\beta p_i}} \ . \tag{14}$$

Note that, if the marks from a link are equally distributed among all the virtual session having the maximum rate through the link, then we have the following.

$$\frac{p_i}{K} \leq q_i \leq p_i \ \text{ for } 1 \leq i \leq K \quad \text{and,} \quad p_i = q_i \ \text{ for } K+1 \leq i \leq N \ . \tag{15}$$

The result follows by combining (14) and (15). □

We now try to interpret the inequality given by (13). Firstly observe that, if Δ_{0i} of virtual session $(0, i)$ can be chosen so that $\Delta_{0i} = \Delta_i$, then $\frac{1}{N}\sum_{i=1}^{N} x_i^{(TCP)} \leq \max_{i \leq i \leq N} x_{0i}$. This shows that there is no advantage in splitting a multicast session into N unicast sessions. In other words, if possible there is always an incentive in forming a multicast group so as to share the network resources in a more efficient manner. The lower bound shows that the total throughput of the competing TCP sessions is not significantly different from the competing multicast receivers. This is indeed the case in many real scenarios and we shall see in our simulation results that the multicast receivers and the competing TCP sessions share the network resources quite evenly when the parameters of the multicast rate control equation are chosen appropriately as discussed earlier.

Further, we note that in practice, even when the multicast virtual sessions have different rates, the throughput of any virtual session may be lower than the throughput of a corresponding TCP session sharing the same set of links. Recall that the throughput of the virtual session is dependent on the RTT used to determine the increase parameter of the congestion-control algorithm. However, to make the virtual session's rate equal to the TCP session's rate, we have to estimate the propagation delay on only the links at which the virtual session is congested. But it is much easier to estimate the end-to-end RTT which is an upper bound on the RTT for the congested links only. Thus, our increase parameter using the end-to-end RTT would lead to a smaller increase parameter than if we had used the RTT on the congested links only (in terms of the notations used in Proposition 4.1, $\Delta_{0i} < \Delta_i$). Thus, the corresponding throughput would be smaller than that of the TCP session passing through the congested links only. Thus in this case, our congestion controller can be TCP-like and its throughput can be as high as TCP's throughput if there is a mechanism to estimate RTT only on the links on which it is congested.

5 Stability

We show the stability of the rate control mechanism given in (10). We consider the following general form of the rate control mechanism given by

$$\dot{x}_{sr} = \frac{H_{sr}(x_{sr})}{U'_{sr}(x_{sr})}\left(\Delta_{sr}U'_{sr}(x_{sr}) - \beta\sum_{l\in L_{sr}} q_{l(sr)}\left(\sum_{m\in S_l}\max_{(m,j)\in V_{ml}} x_{mj}\right)\right) , \quad (16)$$

where, $H_{sr}(.)$ is a strictly positive and continuous function. We first define the function $V(\underline{x})$ as follows.

$$V(\underline{x}) = -\sum_{s\in S}\sum_{r\in R_s}\Delta_{sr}U_{sr}(x_{sr}) + \beta\sum_{l\in L}\int_0^{(\sum_{m\in S_l}(\max_{(m,j)\in V_{ml}} x_{mj})} p_l(u)du \quad (17)$$

Lemma 5.1. *The function $V(\underline{x})$ is strictly convex in $\underline{x}$.*

Proof. This follows from the concavity of the utility functions, the convexity of the max(.) functions and the fact that the marking functions are increasing function of the total rates. □

Since the function $V(\underline{x})$ is strictly convex in $\underline{x}$, for large enough β, $\exists$ a unique $\underline{x}^*$ such that $\nabla_s V(\underline{x}^*) = 0$ [18], where ∇_s denotes a subgradient[1].

Remark 3. *: If the utility functions are of the form $U(x) = -1/(nx^n)$, where $n > 0$, then it is easy to see that $\lim_{|\underline{x}|\to\infty} V(\underline{x}) = \infty$. Thus for any positive β, $\exists$ a unique $\underline{x}^*$ such that $\nabla_s V(\underline{x}^*) = 0$.*

The congestion control equation (16) can be written as follows.

$$\dot{x_{sr}}(t) = -\frac{H_{sr}(x_{sr})}{U'_{sr}(x_{sr})}(\nabla_s V(\underline{x}(t)))_{sr} , \quad (18)$$

where $\nabla_s(\underline{x}(t))$ is a particular subgradient and $(\nabla_s V(\underline{x}(t)))_{sr}$ denotes the (s,r)-th component of the sub-gradient. If the point $\underline{x}^*$ lies on the hyperplanes of non-differentiabilities of $V(.)$, then the particular subgradient we are using in the algorithm need not have a zero. For proving the convergence of rate control equation given in (11) we make the following assumptions.

Assumption 2. *The convex function $V(\underline{x})$ attains its minima at a point where its gradient exists.*

[1] The subgradient for a convex function can be viewed as a generalized gradient even when the gradient does not exist. For "well behaved" functions any convex combination of all the gradients around a small neighborhood of the point where the gradient does not exist can serve as a subgradient. Wherever the gradient exists, the subgradient is unique and is same as the gradient [18].

The preceding assumption is a technicality required for the proof of stability. Suppose the minimum of the convex function $V(\underline{x})$ lies on a hyperplane of non-differentiability. Since the gradient of $V(\underline{x})$ does not exist at the minimum point, there exists a subgradient of $V(\underline{x})$ which is zero at the minimum point, but this need not correspond to the subgradient used in the right hand side of the rate control as shown in (16). However, under Assumption 2, $V(\underline{x})$ reaches its minimum at a point where its gradient exists and so the equilibrium point of the rate control mechanism, $\underline{x}^*$, also corresponds to the minima of $V(\underline{x})$.

Theorem 5.1. *The system of multicast rate control equation given by (16) is globally asymptotically stable.*

Proof. First define the following candidate Lyapunov function.

$$W(\underline{x}) \quad = \quad \sum_{s\in S}\sum_{r\in R_s}\int_{x^*_{sr}}^{x_{sr}} \frac{U'_{sr}(u_{sr})}{H_{sr}(u_{sr})}(u_{sr}-x^*_{sr})du_{sr}$$

It is easy to see that $W(\underline{x})$ is a radially unbounded positive function with a unique zero at $\underline{x}^*$. We also denote the gradient of $W(.)$ by $\nabla W(.)$. Now, note the following for $\delta > 0$.

$$\begin{aligned}
&W(\underline{x}(t+\delta)) - W(\underline{x}(t))\\
&= W(\underline{x}(t) + \delta\dot{\underline{x}}(t) + o(\delta)) - W(\underline{x}(t))\\
&= (\delta\dot{\underline{x}}(t) + o(\delta))^t \nabla W(\underline{x}(t)) + o(\delta)\\
&= -\delta \sum_{s\in S}\sum_{r\in R_s}(x_{sr}(t) - x^*_{sr})(\nabla_s V(\underline{x}(t)))_{sr} + o(\delta)\\
&= -\delta(\underline{x}(t) - \underline{x}^*)^t(\nabla_s V(\underline{x}(t))) + o(\delta)
\end{aligned}$$

Since $\nabla_s V$ is the subgradient of a strictly convex function, we have, $V(\underline{x}(t)) - V(\underline{x}^*) < (\underline{x}(t) - \underline{x}^*)^t(\nabla_s V(\underline{x}(t)))$ from which it follows that,

$$W(\underline{x}(t+\delta)) - W(\underline{x}(t)) \quad < \quad -\delta(V(\underline{x}(t)) - V(\underline{x}^*)) + o(\delta)\ .$$

We thus have,

$$\begin{aligned}
D^+(W(\underline{x}(t))) \quad &= \quad \limsup_{\delta\to 0^+} \frac{W(\underline{x}(t+\delta)) - W(\underline{x}(t))}{\delta}\\
&< \quad \limsup_{\delta\to 0^+} \frac{-\delta(V(\underline{x}(t)) - V(\underline{x}^*))\ + o(\delta)}{\delta}\\
&= \quad -(V(\underline{x})) - V(\underline{x}^*)) \leq 0
\end{aligned}$$

The last inequality follows from the fact that, under Assumption 2, $V(.)$ attains its minima at x^*. In the above, D^+ denotes the upper right Dini derivative[2] with respect to the time variable. Hence the system is globally asymptotically stable and all trajectories ultimately converge to $\underline{x}^*$. □

[2]The upper right Dini derivative of a function $f(.)$ at t is defined as $D^+(f(t)) = \limsup_{\delta\to 0^+} \frac{f(t+\delta)-f(t)}{\delta}$

6 Implementation Considerations

In this section, we discuss some implementation issues with the congestion control scheme described earlier. We discuss how the algorithm can be modified for the case of discrete bandwidth layers and also a distributed implementation of the scheme is discussed.

6.1 Implementation with Discrete Bandwidth layers

In [16], the authors have proposed a multirate reception protocol called RLM (receiver driven layered multicast). In layered multicast, discrete bandwidth layers are offered, and so the receivers cannot change their rates to any arbitrary level as desired by an accurate implementation of the congestion control mechanism described in the earlier section.

To overcome this difficulty in implementing the congestion control algorithm for layered multicast, the following can be done in practice. Suppose a multicast group has n different levels of available bandwidths, $b_1,\ b_2,\ \ldots, b_n$, where $b_i < b_{i+1}$. This can correspond to data being transmitted over n multicast groups with bandwidths $b_1,\ (b_2 - b_1),\ \ldots(b_n - b_{n-1})$. Such a framework has been discussed in [23]. The receivers can subscribe to different number of layers based on its requirements. So, the congestion control algorithm would be to drop layers whenever there is congestion in the network, and to add more layers whenever there is enough bandwidth available. More specifically, let the rate of the receiver (s, r) after the k^{th} update be $x_{sr}(t_k)$. Also, let this rate be such that the receiver can subscribe to bandwidth till m th level. $(x_{sr}(t_k) = b_m)$. Now the new rate calculated at the receiver will be done in the following steps.

1. Calculate the new rate according to the procedure discussed in the Receiver Node's algorithm in the previous section. Let this rate be $x_{sr}(t_{k+1})$.

2. If the new desired rate $x_{sr}(t_{k+1})$ is such that, $x_{sr}(t_{k+1}) \geq b_n$, then the receiver subscribes to all the available multicast groups and $x_{sr}(t_{k+1})$ is set to b_n. If $x_{sr}(t_{k+1}) < b_n$, calculate j such that the following holds.

$$b_j \quad \leq \ x_{sr}(t_{k+1}) \quad < b_{j+1} \tag{19}$$

   Here j is clearly the level up to which the receiver can subscribe.

3. If the new desired level of subscription j is more than the previous subscription level m, then the receiver adds more layers and if the new level is less than the previous one then layers are dropped accordingly.

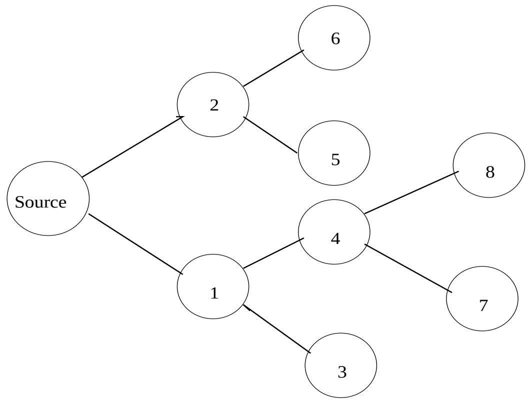


Figure 2: A multicast routing tree with 5 receivers and 3 intermediate junction nodes.

Clearly, if the number of discrete bandwidth layers offered is large, the stable rates with the modification just discussed can be very close to the equilibrium rates of an exact implementation of the congestion control scheme discussed in the previous section.

6.2 Distributed Implementation

In [2], the authors have used the concept of upward consolidation of congestion control in a multicast tree. Since in the receiver driven congestion control scheme, feedback is provided about the updated rates (or subscription levels) to the source, we discuss a similar implementation of how the updated rates in our algorithm can be consolidated along the reverse path from the receiver to the source.

Consider the multicast tree in Figure 2 with hierarchical organization of multicast nodes as shown. The source is at the top of the tree and there are five receivers, the nodes 3, 5, 6, 7 and 8. The nodes 1, 2, 4 where the multicast tree splits are referred to as junction nodes. Any junction node can aggregate the feedback about the updated rates (or subscription level) before passing it on to the parent node. For example, node 4 can calculate the maximum of the rates (or subscription level) of receiver 7 and 8 and pass that to node 1. This is the rate at which node 1 sends data to node 4. Node 1 updates the source about the maximum of, rate at which it has to send data to node 4 and the update of the rate (subscription level) it receives from node 3. The source sends data to node 1 with this rate (subscription level). Thus, every junction node consolidates the updates of the subscription level or the rates received from its children nodes, before providing a feedback to its parent node about the new desired rate or subscription level. In this implementation, every node in the network which is a junction node of some multicast tree, has to keep track of the desired level of subscription or rates of every node it sends packets to. However, the congestion control for the unicast sessions within the network can still be source driven based on the feedback received from the receivers.

Finally, a few comments on the scalability of the algorithm. In the *Node's Algorithm* in Algorithm 4.1, we need to ensure that marks are only sent to receivers with maximum rate. Consider the junction node 1 in the multicast tree in Figure 2. In order to send the marks to a receiver with maximum rate, all that node 1 needs to know is whether the receiver with maximum rate lies in the path to node 3 (if the receiver at node 3 has rate more than the receivers at node 7 and node 8) or in the path to node 4 (if the receivers at node 7 or node 8 has rate more than the receiver at node 3). Thus it is sufficient to *unmark* packets to be sent to all but one of the paths. This takes care of the fact that marks are eventually sent only to a receiver with maximum rate sharing the link. Thus, the information required at the junction nodes is the identification of suitable child/children in the multicast tree. Thus, the state required to be maintained at any junction node is of size, *number of children of the junction node in the multicast tree.* This number is much smaller than the number of receivers, for most multicast trees.

7 Results and Discussion

In this section we present numerical and simulation results to show the performance of the congestion control algorithm presented before. We solved the differential equation model (also called the fluid model) for rate control with $H(x_{sr}) = 1$ numerically for the Y-network and a general network, and also performed packet based simulations for the Y-network with large number of flows. The purpose of numerical solution of the rate control equation is two-fold. First, we show from our numerical solutions that, with an appropriate utility function the congestion control algorithm can be used to achieve max-min fair rate allocation. Second, we also study the impact of discrete bandwidth layers a general network topology. We present simulation results for a packet level simulation, and show that the multicast rate control equation gives fairness in the presence of unicast sessions using a TCP based congestion control mechanism. We show the effectiveness of the congestion control mechanism with large number of TCP sessions and multicast sessions in the Y-network.

7.1 Numerical Results from the Rate Control Equations

7.1.1 Results for a Y network

There are two virtual sessions of a multicast session and two unicast sessions sharing the three links A, B, and C as shown in Figure 1. Let x_{01} and x_{02} be the rates of the two virtual sessions of the multicast session and x_1 and x_2 be the rates of the two unicast sessions. We first show a justification behind using (11) for multicast congestion control by showing that the stable solution of (7) indeed approaches the unique stable solution of (11) for large n (n is the parameter in (7)). Based on the buffer overflow probability of an $M/M/1/B$ queue, we have chosen the marking probability $p_l(.)$ in link l to be $\frac{(x-c_l)^+}{x}$, where x is the total flow rate through the link and c_l is the capacity of the link. Such a marking function has been considered in [10]. We choose the capacity of link A to be 10 units, the capacity of link B to be 15 units and the capacity of link C to be 5 units. In Table 1 we show the results with the same utility function $\ln(x)$ for all the virtual sessions and the unicast sessions.

Virtual/Unicast Sessions	Links Used	Equilibrium rates of (7) for				Equilibrium rates of (11)
		$n = 2$	$n = 5$	$n = 10$	$n = 20$	
(0,1)	A, B	4.13	4.49	4.40	4.31	4.29
(0,2)	A, C	2.98	3.27	3.36	3.40	3.41
1	A, B	3.35	3.88	4.14	4.27	4.29
2	B, C	2.18	1.99	1.93	1.90	1.90

Table 1: Table showing the convergence of the sequence of equations equations (7) for large n to (11), for the Y-network shown in Figure 1. The parameters used are $\beta = 5$ and the capacities for the different links are $c_A = 10,\ c_B = 15,\ c_C = 5$. The differential equations were discretized with a step size of 0.01 and the value of Δ_{sr} was taken to be 1 for all the sessions.

Next we show that by choosing the utility functions appropriately the congestion control algorithm can be used to obtain max-min fair rates. To this end, we chose utility functions of the form $U_n(x) = \frac{-1}{(n-1)x^{n-1}}$ with large enough n, and marking functions of the form $\frac{(x-c_l)^+}{x}$. We can observe from Table 2 that as n grows large the equilibrium rates with utility functions $U_n(x)$ are close to the max-min fair rates, which can be easily calculated as $x_{01} = 3.75,\ x_{02} = 2.5,\ x_1 = 3.75,\ x_2 = 2.5$, for $c_A = 10, c_B = 15, c_C = 5$ [20]. This is not difficult to prove with our choice of marking function.

Virtual/Unicast Sessions	Links Used	Equilibrium rates of (13) for Different $U(.)$		
		$U(x) = \frac{-1}{x}$	$U(x) = \frac{-1}{3x^3}$	$U(x) = \frac{-1}{10x^{10}}$
(0,1)	A, B	4.08	3.78	3.75
(0,2)	A, C	3.11	2.62	2.50
1	A, B	4.08	3.78	3.75
2	A, C	2.47	2.49	2.50

Table 2: Table showing the equilibrium rates for of the different sessions in the Y network for different utility functions. The same utility function has been assumed for all the sessions and $\beta = 1$.

7.1.2 Results for a more General Network

We now consider a more complex network as shown in Figure 3 to illustrate the impact of discrete bandwidth layers being available. Consider the network shown in Figure 3. The different links

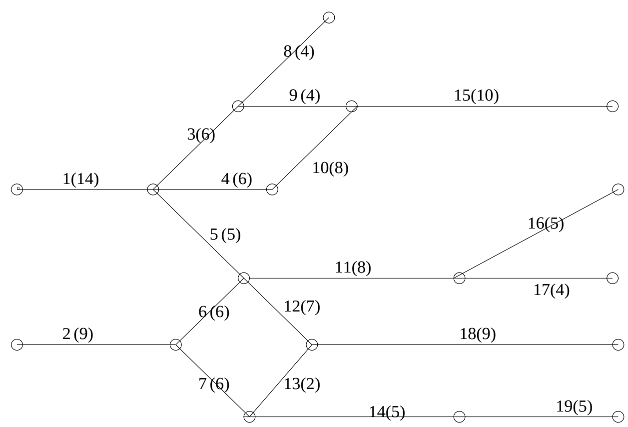


Figure 3: A general network with 19 links. The capacities of the different links are given in parentheses next to the link indices. There are a total of 19 virtual sessions and unicast sessions going through the links as given in Table 3.

are indexed as shown in the figure, and the corresponding capacities are given in the parentheses next to the link indices. Fluid level simulations were performed for this network with 6 multicast sessions and 5 unicast sessions. All the unicast sessions were assumed to have utility functions of the form $\frac{-1}{x}$, since this function is one of the utility functions used to model TCP behavior [10]. The different utility functions for the virtual sessions, the links each virtual session and the unicast sessions pass through are shown in Table 3. The rate control algorithm converges and the

Virtual/Unicast Sessions	Links Used	Utility Functions, U(x)	Equilibrium rates of (13)	Equilibrium rates when the rates of multicast sessions are in discrete steps of:		
				0.15	0.05	0.005
(0,1)	1, 3, 8	$\ln(x)$	2.6437	2.10	2.40	2.515
(0,2)	1, 9, 15	$\ln(x)$	2.6537	2.25	2.70	2.685
1	1, 4, 10, 15	$\frac{-1}{x}$	2.0110	2.9972	2.5764	2.3061
(2,1)	1, 5, 11, 16	$\ln(x)$	1.5172	1.20	1.40	1.455
(2,2)	1, 5, 12, 18	$\frac{-1}{x}$	1.5273	1.20	1.40	1.46
3	1, 5, 11, 17	$\frac{-1}{x}$	1.1434	1.3867	1.2137	1.1807
(4,1)	2, 6, 2, 18	$\ln(x)$	3.0168	2.10	2.35	2.83
(4,2)	2, 7, 14, 19	$\frac{-1}{x}$	1.4818	1.25	1.55	1.53
5	2, 7, 13, 18	$\frac{-1}{x}$	1.5891	1.8854	1.7246	1.6631
6	1, 3, 9, 15	$\frac{-1}{x}$	1.5550	1.9055	1.6036	1.5764
(7,1)	1, 3, 8	$\ln(x)$	1.8702	2.10	2.05	1.97
(7,2)	1, 5, 11, 16	$\frac{-1}{x}$	1.8345	1.50	1.70	1.745
(7,3)	1, 5, 11, 17	$\ln(x)$	1.8450	1.5	1.75	1.76
(8,1)	2, 6, 12, 18	$\ln(x)$	3.0308	2.7	2.95	3.075
(8,2)	2, 6, 11, 16	$\frac{-1}{x}$	2.0700	1.95	1.95	2.04
(8,3)	2, 7, 14, 19	$\ln(x)$	2.1952	1.50	1.85	2.05
9	2, 7, 14, 19	$\frac{-1}{x}$	1.4414	1.8854	1.6327	1.4987
(10,1)	1, 4, 10, 15	$\ln(x)$	4.0240	2.7	3.15	3.56
(10,2)	1, 5, 1, 17	$\frac{-1}{x}$	1.2849	1.35	1.3	1.325

Table 3: The equilibrium rates for the different virtual sessions and the unicast sessions for the network shown in Figure 3. The different parameters used are $\beta = 3$, discretization step size, $\delta = 0.001$.

equilibrium rates are shown in the Table 3. This is when no discrete layering of bandwidth is there. In the case of layered multicast, the bandwidth that can be requested by the receivers is discrete. We show the effect of the discrete bandwidth layers on the equilibrium rates with simple illustrations. First we consider a scenario when bandwidth is available in steps of 0.15 (in other words, each of the multicast groups requires a bandwidth 0.15), so the allowable rates for a session are in integer multiples of 0.15. The corresponding equilibrium rates are shown in Table 3. Also shown are the equilibrium rates when the discrete bandwidths offered are integer multiples 0.05 and 0.005. The rate control equations still converge to a equilibrium rate with this discretization of rates for the multicast sessions. Further, it can be observed that the rates get closer to the equilibrium rates with no discretization of rates when the number of multicast groups are large. We also note that, with layered multicasting for the multicast sessions, the unicast sessions get rates larger than the case with no layering. This is seen by the equilibrium rates of the unicast sessions in Table 3.

7.2 Results with Packet level Simulation

We present simulation results using a packet model (in contrast to the earlier fluid model) for transmission of data. We chose a network with topology as the Y-network as in Figure 1 to perform packet level simulations. The network has four nodes n1, n2, n3 and n4. The nodes n1 and n2 are connected by an 80 Mbps link and has a one-way propagation delay of 20 ms, the link between nodes n2 and n3 is 60 Mbps and has a delay of 20 ms and the link between n2 and n4 is of 40 Mbps and has a delay of 10 ms.

There are 3 classes of users, Class 1 consists of multicast sessions with source at node n1 and two receivers for each of the multicast sessions in this class at node n3 and n4. Class 2 has unicast sessions from node n1 to n3 and Class 3 sessions are unicast sessions from node n1 to n4. For the set of simulation results shown in this section, we have considered 40 sessions corresponding to each of the classes. So link A is shared among 120 sessions, link B is shared among 80 sessions and link C is shared among 80 sessions. The packet sizes are assumed to be 1000 *bytes* and so the capacity of link A, link B and link C are equivalent to 10,000, 7,500 and 5,000 packets per second respectively.

In these set of simulations we have taken the case when the users (sources for unicast sessions and receivers for any multicast receiver) respond based on the ECN marks received. Since the purpose of the simulation experiment is to demonstrate the effectiveness of the congestion control algorithm when unicast and multicast sessions coexist in a network, we have chosen the marking probability in each link to be $\frac{(\hat{\lambda}_l - \tilde{C}_l)^+}{\hat{\lambda}_l}$, where $\hat{\lambda}_l$ is an estimate of the total arrival rate in a link and $\tilde{C}_l$ can be thought as a marking level of the link l and we have chosen $\tilde{C}_l = 0.98 \times C$ in our simulations. Such a scheme can be implemented using a virtual queue with capacity $\tilde{C}_l$ [11]. However, the observations in our simulations should hold good for any other marking function used by the links.

All the multicast sessions were made to adapt their rates based on (11). The values of the parameters used in the algorithm were discussed in 4.1. We used two kinds of rate control mechanisms for the unicast sessions. In one scenario we assumed a TCP-based congestion control mechanism for the unicast sessions and in the other, rate-based congestion mechanism were used for the unicast sessions. We present and discuss the results for the former scenario, as the observations for the latter one are very much similar. For the purposes of our simulations we neglected the slow start phase of the TCP sessions and assumed all of them to be in the congestion avoidance phase. Since our simulations used an ECN marking based mechanism at the links, which can be tuned to result in negligible packet losses [11], we also neglected the effect of retransmission time out. Every received ACK increased the TCP current window size W by $\frac{1}{\lfloor W \rfloor}$ and every marked packet echoed back by the receiver halved the current window size. We have taken persistent sources (popularly known as *web elephants*) in our simulations, thus neglecting the effect of any short flows (also known as *web mice*). However, simulations in [11] show that the effect of short flows is to primarily change the throughput seen by each users, but they do not affect the relative fairness among the long flows.

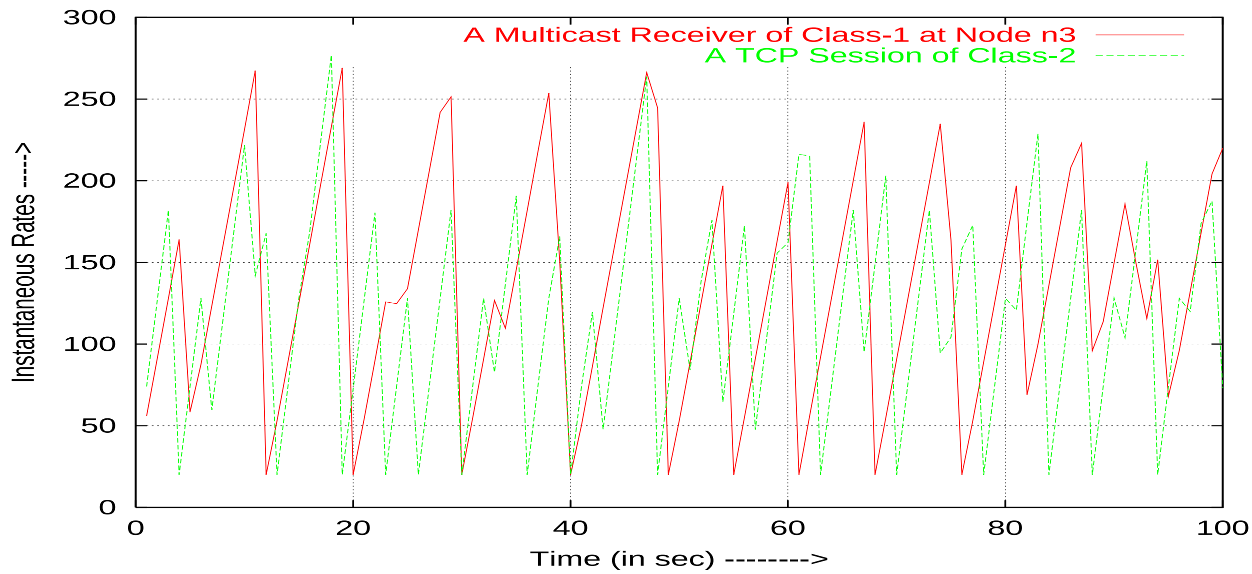
Since we want to show that the congestion control algorithm ensures fairness to the multicast sessions as well as to the TCP sessions, we choose the multicast congestion control parameters so that the rate control mechanism for the multicast sessions can mimic the TCP behavior. So the utility functions for all the multicast sessions were chosen as $A \arctan(Ax)$ ($A = RTT \times \sqrt{\beta}$) and the value of β was chosen as $\frac{2}{3}$ (this model for TCP has been discussed in [7]). Let S be the length of the intervals at which the multicast receivers update their rates based on the fraction of marked packets received over a duration of S. Let Δ be the steady increase rate of a multicast session. To make the rate control behavior close to that of TCP, we choose $\Delta = \frac{1}{r^2}$, where r is the RTT which is twice the propagation delay from the multicast source to that receiver. This is clear from the fact that in the absence of any congestion TCP increases its window size (rate × RTT) by 1 every RTT. In accordance with the above discussion, for the Y-network we simulated, we chose $\Delta = 150$ for the multicast receivers at node n3 and, $\Delta = 270$ for the multicast receivers at node n4. The rate update interval S was chosen as 0.1 sec to allow the reception of sufficient number of packets

Session Classes	Average of throughputs of the 40 sessions over a duration of 500 sec	Variance of throughputs of the 40 sessions over a duration of 500 sec	Equilibrium rates from the corresponding fluid model
Multicast Receivers at node n3 of Class 1 sessions	90.38	1.86	94.30
Class 2 TCP Sessions	86.89	1.96	94.30
Multicast Receivers at node n4 of Class 1 sessions	68.77	0.49	68.39
Class 3 TCP Sessions	53.11	0.42	59.79

Table 4: Table showing the Average of throughput of the sessions in packets per second using the proposed rate based congestion control algorithm for the multicast sessions, and TCP based congestion control for the unicast sessions. The corresponding equilibrium rates for a fluid model with the same set of parameters are also shown.

at the receivers before updating the rate based on the fraction of marked packets received.

In Table 4, we have shown the average of the throughput (in packets per second) for the all the sessions and the equilibrium rates from a fluid model using the same set of Δ's, β and utility functions. We have also shown the variance of throughput of all the sessions in a class to show that the throughput as seen by different users within a class are close. It can be observed that the scheme can perform quite well in the presence of TCP sessions and is fair to all the multicast receivers and the TCP sessions. The average throughput of the virtual and the unicast sessions along the same path are seen to be close, and close to their corresponding fluid limits. In Figure 4 we show the variation of instantaneous throughput of a typical multicast session and a typical TCP session along the two paths n1-n2-n3 and n1-n2-n4. For the multicast sessions the interpretation of instantaneous throughput is just the rate and for the TCP sessions it is the number of packets transmitted over a period of 0.1 sec divided by 0.1 sec. This is chosen to be consistent with the fact that the rate update intervals of the multicast receivers were chosen to be 0.1 sec in our simulations. Clearly, even within short intervals, the TCP sessions and the multicast receivers compete quite evenly. It has however been observed in the simulations that the rate update intervals for the



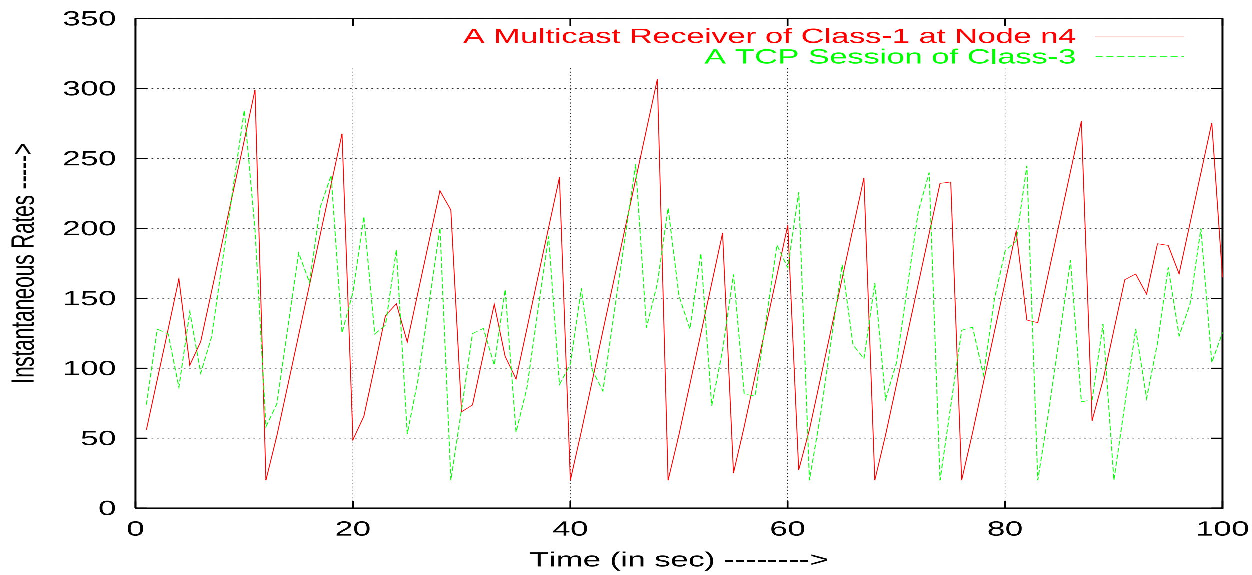


Figure 4: Plots showing the variation of rates of a Multicast Receiver and a TCP session sharing the path n1-n2-n3 (for the Y-network) in the first plot, and the path n1-n2-n4 (for the Y-network) in the second plot.

multicast sessions have to be chosen carefully to ensure good performance of the scheme. In our simulations with various parameters we found that an update interval which is 2 or 3 times the one way propagation delay of the particular multicast receiver is sufficient.

8 Conclusions

In this paper, we have proposed a class of congestion control algorithms for networks with unicast and multicast sessions. The congestion control mechanism can be implemented in a decentralized manner and with a simple one-bit marking scheme. The marking scheme is simple: *among all receivers whose rate equals to the rate of the multicast session, randomly select a receiver to send the ECN mark.* We have also proved the stability of the class of congestion control algorithms. Implementation issues of the scheme in a layered multicast scenario have been discussed. Fluid level simulation results were provided to show how max-min rates can be obtained using the algorithm. It was also shown that layered multicasting with the congestion control mechanism proposed can have stable rates close to the case when the available bandwidth is not discrete. We also performed a packet level simulation in the presence of multiple TCP sessions along with multiple multicast sessions. Our results suggest that the scheme can be used efficiently in the presence of TCP sessions and with a judicious choice of parameters it can ensure fair long run throughput among different TCP sessions and the multicast receivers.

The algorithm proposed in this paper assumes that each router can identify the virtual session that receives the maximum rate within a multicast session passing through it. This can be done by ensuring that the routers can identify an appropriate path in the multicast tree where the receiver with maximum rate lies and so the state size that may be needed to be maintained at a router size does not grow with the number of receivers. However, certain implementations of layered multicasting assume that each layer is transmitted as a separate multicast group. The implementation of the algorithm proposed in this paper in such a context is an open issue.

Acknowledgment

It is a pleasure to thank Prof. Frank Kelly for comments on an earlier version of this paper.

References

[1] S. Athuraliya, D. E. Lapsley, and S. H. Low. Random early marking for Internet congestion control. In *Proceedings of IEEE GLOBECOM*, 1999.

[2] A. Basu and S. J. Golestani. Estimation of receiver round trip times in multicast communications. In *Proceedings of IEEE INFOCOM'98*, March 1998.

[3] S. Floyd. TCP and explicit congestion notification. *ACM Computer Communication Review*, 24:10–23, October 1994.

[4] S. J. Golestani. Fundamental observations of multicast congestion control in Internet. In *Proceedings of the IEEE INFOCOM*, 1999.

[5] E. Graves, R. Srikant, and D. Towsley. Decentralized computation of weighted max-min fair bandwidth allocation in networks with multicast flows. In *Proceedings Tyrrhenian International Workshop on Digital Communications 2001 (IWDC '01), Taormina, Italy*, Sept. 2001.

[6] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multirate multicast sessions. In *Proceedings of the IEEE INFOCOM*, 2001.

[7] F. P. Kelly. Mathematical modeling of the Internet. In *Fourth International teletraffic Conference on Industrial and Applied Maths*, July 1999.

[8] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[9] F.P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1998.

[10] S. Kunniyur and R. Srikant. End-to-end congestion control: utility functions, random losses and ECN marks. In *Proceedings of INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[11] S. Kunniyur and R. Srikant. A time-scale decomposition approach to adaptive ECN marking. *IEEE Transactions on Automatic Control*, June 2002.

[12] A. Legout and E. Biersack. PLM: Fast convergence of cumulative layered multicast transmission schemes. In *Proceedings of ACM SIGMETRICS*, 2000.

[13] X. Li, S. Paul, and M. Ammar. Layered video multicast with retransmission. In *Proceedings of IEEE INFOCOM*, New York, NY, June 1998.

[14] M. Luby, V. Goyal, S. Skaria, and G. Horn. Wave and equation based rate control using multicast round trip time. In *Proceedings of ACM SIGCOMM*, 2002.

[15] A. Mankin, A. Romanow, S. Bradner, and V. Paxon. IETF criteria for evaluating reliable multicast transport and application protocols. *Internet Draft, Network Working Group.*

[16] S. Mccanne, V. Jacobson, and M. Vetterli. Evaluation of hierarchical rate control. In *Proceedings of ACM SIGCOMM*, September 1996.

[17] J Mo and J. Walrand. Fair end-to-end window based congestion control. *IEEE/ACM Transactions on Networking*, pages 556–567, October 2000.

[18] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[19] N. Rouche, P. Habets, and M. Laloy. *Stability Theory by Liapunov's Direct Method*. Springer Verlag, 1977.

[20] S. Sarkar and L. Tassiulas. Fair allocation of discrete bandwidth layers in multicast networks. In *Proceedings of INFOCOM 2000*, 1999.

[21] S. Sarkar and L. Tassiulas. Fair distributed congestion control in multirate multicast networks. In *Proceedings of INFOCOM 2000*, 1999.

[22] J. K. Shapiro, D. Towsley, and J. Kurose. Opitimization-based congestion control for multicast communications. In *Proceedings of Networking 2002*, 2002.

[23] L. Vicisano, L. Rizzo, and J. Crowcroft. Tcp like congestion control for layered multicast data transfer. In *Proceedings of RMRG Meeting*, December 1998.