The Loss Path Multiplicity Problem in Multicast Congestion Control *

Supratik Bhattacharyya †
Don Towsley
Jim Kurose
Department of Computer Science
University of Masschusetts, Amherst.
Amherst MA 01003 USA
emails : [bhattach,towsley,kurose]@cs.umass.edu

CMPSCI Technical Report TR 98-76

August 12, 1998

Abstract

An important concern for source-based multicast congestion control algorithms is the **loss path multiplicity (LPM)** problem that arises because a transmitted packet can be lost on one or more of the many end-to-end paths in a multicast tree. Consequently, if a multicast source's transmission rate is regulated according to loss indications from receivers, the rate may be completely throttled as the number of loss paths increases. In this paper, we analyze a family of additive increase multiplicative decrease congestion control algorithms and show that, unless careful attention is paid to the LPM problem, the average session bandwidth of a multicast session may be reduced drastically as the size of the multicast group increases. This makes it impossible to share bandwidth in a *max-min fair* manner among unicast and multicast sessions. We show that max-min fairness can be acheived however, if every multicast session regulates its rate according to the most congested end-to-end path in its multicast tree. We present an idealized protocol for tracking the most congested path under changing network conditions, and use simulations to illustrate tnat tracking the most congested path is indeed a promising approach.

Keywords : multicast, congestion control, multiple loss paths, max-min fairness.

*This work was supported by the National Science Foundation under grant NCR 9508274 and by TASC under subcontract J08899-S97114. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

†This author was supported for this work in part by a Lucent Technologies Fellowship.

1 Introduction

The deployment of multicast services in wide-area networks is expected to lead to a proliferation of point-to-multipoint applications in the near future. Such applications will constitute a significant portion of the overall network traffic and will compete with existing point-to-point applications for network bandwidth. Hence it is necessary to control and regulate their bandwidth consumption in order to prevent network congestion.

One possible approach towards multicast congestion control is **source-based** rate control, in which a multicast source regulates its transmission rate in response to loss indications (eg., NAKs) from receivers. A number of specific source-based rate control schemes have been proposed [12, 5, 14]; these represent important first solutions in a very large solution space. However, a number of fundamental issues remain open and have to be addressed by any source-based approach towards multicast congestion control. In this paper, we identify and examine two such issues.

First, the loss indications received by a multicast source from multiple receivers reflect diverse congestion conditions in various parts of the network, and have to be appropriately combined when making a single rate control decision. A transmitted packet may be lost on one or more of the many end-to-end paths in a multicast distribution tree. The number of such loss paths is likely to increase with an increase in the number of receivers; hence the probability that the source receives *at least* one loss indication for every transmitted packet becomes high. If the source reduces its rate in response to every loss indication that it receives, its transmission rate will be severely throttled.

The second important issue concerns fairness in bandwidth sharing among unicast and multicast sessions. Multicast connections should not be allowed to usurp a large share of bandwidth since that may starve unicast connections. On the other hand, a multicast session's rate should not be throttled to the extent that its bandwidth share is drastically reduced, since that will discourage the widespread deployment and use of multicast technology.

The central issue of interest in this paper is how a multicast source combines loss indications from multiple receivers in its multicast distribution tree for rate regulation, and how such combinations affects fairness in bandwidth sharing. We analyze a family of additive increase multiplicative decrease congestion control algorithms [4], and show that unless careful attention is paid to the existence of multiple loss paths in a multicast tree, the average bandwidth share of a multicast session may be reduced drastically as the size of the multicast group grows. Our results also indicate that it is impossible to share bandwidth in a max-min fair manner unless the problem of multiple loss paths is addressed. Our definition of max-min fairness is based on allocating bandwidth to a multicast session according to the most congested path in its multicast tree. We show that it is possible to ensure max-min fairness according to this definition only if every multicast source regulates its rate according to the most congested path in its tree, or equivalently, according to loss indications from only the "lossiest" receiver in the multicast group. We present simulation results that show tracking the worst receiver is indeed a promising approach.

The rest of the paper is organized as follows. Section 2 presents an discussion on the problem arising out of the existence of multiple loss paths in a multicast tree, and its effect on fair bandwidth sharing. In section 3, we describe a family of additive increase multiplicative decrease congestion algorithms and express the average session bandwidth as a function of the observed loss probability at the source for these algorithms. Section 4 describes a method for analytically deriving the loss indication probability at a multicast source for some of these algorithms. In doing so, we uncover some differences between applications where the source retransmits lost data packets and ones where it does not do so. In section 5, we present a number of case studies where we compute the average session bandwidth for some of these algorithms for a number

of network scenarios. The results illustrate the severe degradation in a multicast session's bandwidth share when the source responds to loss indications from all receivers in the group. Given the recent interest in the PGM protocol [15], we evaluate the performance of the the congestion control algorithm proposed for it in [15] and observe similar degradation in multicast session bandwidth. Section 6 presents simulation results that indicate that it is indeed possible to eliminate the problem of multiple loss paths, and to achieve max-min fair sharing of bandwidth by regulating a source's rate according to loss indications from only the worst receiver in the multicast group. Section 7 concludes the paper.

2 The Multicast Loss Path Multiplicity Problem and Fairness

An Internet-like datagram network suffers from limited observability and controllability due to lack of sufficient support at the network (IP) layer. Hence the most widely used unicast congestion control approach is end-to-end control of user traffic at the transport level. In this approach, each traffic source regulates its rate based on loss (and/or delay) feedback from its receiver. The most popular approach towards rate control is the additive increase and multiplicative decrease of a rate (or window, as in the case of TCP [9]) parameter. Rate is decreased multiplicatively every time a congestion feedback (e.g., loss indication) is received, and increased additively otherwise.

Let us now consider extending this approach to a multicast source, with the source adjusting its rate in response to **loss indications (LI)** from receivers in its multicast group. This gives rise to two problems. The first is the problem of spatial loss correlation - a single packet loss may affect multiple receivers; hence the source may receive more than one LI for the loss. If the source reduces its rate in response to each such LI, it will have overcompensated for the single loss. One possible way of countering this problem is to have the source reduce its rate less aggressively for each individual LI. Alternatively, assuming all LIs for the same packet loss reach the source within a certain time window, the source can react to only one of them and ignore the rest [12].

The second problem arises due to the existence of multiple end-to-end paths in a multicast tree. Suppose that a multicast source reduces its rate in response to LIs from all its receivers, but reacts to no more than one LI per transmitted packet. However, a transmitted packet may be lost independently on one or more of the multiple paths in the tree. As the number of such paths increases, the probability that the source receives at least one LI per transmitted packet also increases. We refer to this problem as the **loss path multiplicity (LPM)** problem. In order to gain an intuitive understanding of the problem and its effect, let us consider a multicast group with n receivers, each independently experiencing a loss probability of p. Then the probability that the source receives at least one LI per transmitted packet is given by $Q = 1 - (1 - p)^n$. As $n \to \infty, Q \to 1$. Therefore the multicast source regulates its rate as if it were observing a single network path with loss probability Q, and the average session bandwidth is very low.

If the LPM problem reduces the bandwidth share of multicast sessions, competing unicast sessions will receive most of the available network bandwidth, resulting in unfairness in bandwidth sharing. In order to evaluate the extent of this unfairness, we introduce the following fairness criterion. First, neither unicast sessions nor multicast sessions are to be given preferential treatment when allocating bandwidth. Second, bandwidth is allocated to each multicast session according to the most congested path in its tree. Such a policy has already been proposed both for multicast ABR services [16] and for the Internet [14] and is conformant with the widely popular notion of **max-min** fairness [2]. For example, suppose there is an amount of bandwidth B, available on the most bandwidth-constrained path in a multicast tree and there is one more unicast session that traverses this path. Then under the fairness definition, the multicast and

the unicast session will each be allocated a share $B/2$. Consequently the multicast session would also be allocated bandwidth $B/2$ on every other path in its tree, even if there is excess capacity on those paths. The excess bandwidth on those paths is then available to other sessions that traverse those paths.

Note, that we consider this specific criterion in order to provide the context in which to study the effect of the LPM problem. Choosing an appropriate fairness criterion is a policy issue. Any such policy has to address two key issues. The first is whether unicast sessions are to be given preferential treatment vis-a-vis multicast sessions, or vice-versa. Our policy makes no assumptions about the kind of preferential treatment to be given to any session. However, multicast sessions may be given incentives, in terms of a larger share of bandwidth, since they make more efficient use of network resources. The second issue is which end-to-end paths in a multicast are to be considered for bandwidth allocation. Under our chosen fairness criterion, multicast bandwidth is allocated based on the notion that a multicast session can only use as much bandwidth as is available on the most bandwidth-constrained path. This does not take account inter-receiver fairness [10, 14], i.e. whether it is fair to constrain all receivers in a multicast group to receive at the rate allowed for the most congested path. A different policy may require that multicast bandwidth availability on some or all of the end-to-end paths be taken into account.

In the rest of the paper, we examine how the LPM problem may introduce unfairness according to our definition of max-min fairness. In the course of our study, we also identify a promising end-to-end approach for ensuring fairness. This approach is based on having each multicast source identify the most congested path in its distribution tree, by identifying the "lossiest" or "worst" receiver i.e., the one experiencing the highest end-to-end loss probability[1]. The source rate is then regulated in response to LIs from only this receiver. Of course, algorithms for increasing and decreasing source rate have to be chosen such that any two sessions experiencing the same end-to-end loss probabilities[2] will receive equal shares of bandwidth.

We close this section with a brief description of representative schemes [5]. In any such scheme, a subset of receivers in a multicast group is designated as **representatives** such that, the source reduces its rate only in response to LI from a representative and ignores all LIs from non-representatives. Therefore, a representative scheme tracks only some of the many loss paths in a multicast tree, and has the potential of alleviating the effect of the LPM problem. In section 5, we present case studies that illustrate some of the benefits and problems of controlling the source rate using representatives. Note that tracking the worst receiver is equivalent to having a representative scheme with a single representative.

3 A Family of Rate Control Algorithms

In this section we describe a family of additive increase multiplicative decrease algorithms, collectively referred to as FLICA (Filtered Loss Indication-based Congestion Avoidance), for regulating a multicast source's transmission rate. Each algorithm in the class decreases the transmission rate multiplicatively in response to LIs from receivers, and increases it additively in the absence of LIs. From the discussion in Section 2, we observe that every LI from every receiver may not be considered for rate adjustment. The source decides which LIs to use for this purpose and filters out the rest. Let us define a **congestion signal (CS)** as an LI that the source uses for rate adjustment. We can identify two main components for any FLICA algorithm (Figure 1) :

[1] If the worst receiver is not unique, then we can choose any one of the worst receivers.

[2] For a multicast session, consider the worst receiver's end-to-end loss probability.

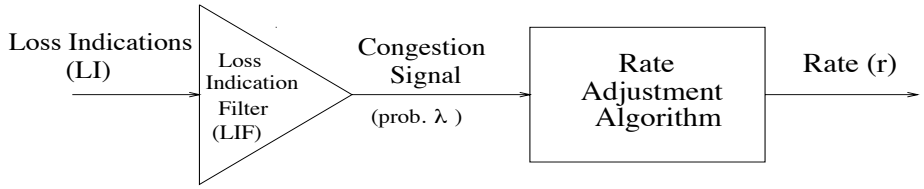

Figure 1: General representation of FLICA algorithms

- **a Loss Indication Filter (LIF)** : this determines which of the LIs received are to be considered as CSs.
- **a Rate Adjustment Algorithm** : an algorithm that determines how to decrease the rate when a CS is received and how to increase the rate in the absence of CSs.

The design of the LIF is policy dependent. For example, an LIF may filter out LIs from non-representatives, as in the case of a representative-based scheme. It may also be timer-driven, letting through no more than one LI within a certain time interval. Such a time-driven LIF corresponds closely to the LTRC scheme in [12]. The LIF filter need not necessarily be located at the source, and may be centralized or distributed. For example, the representative scheme in [5] proposes that non-representative receivers suppress their LIs using backoff timers. For an active network protocol such as the one proposed in [15], filters are actually located at the active nodes inside the network and selectively forward LIs towards the source. Note that Figure 1 applies to unicast sources as well, with the LIF in that case letting through all LIs from the single receiver.

For every FLICA algorithm, the source maintains a variable r that represents the current transmission rate of the source. The value of r is adjusted in response to CSs in the following manner :

$$\begin{array}{ll} \text{On receiving a CS,} & : \ r \leftarrow r - r/C, \\ \text{In the absence of any CS for } S \text{ units of time,} & : \ r \leftarrow r + 1. \end{array}$$

where C and S are adjustable parameters. Therefore the transmission rate is reduced by $1/C$ of its current value on receiving a congestion signal (multiplicative decrease) . In the absence of such signals, r is increased by 1 every S units of time (additive increase). A particular FLICA algorithm is completely defined by specifying its LIF and the values of C and S.

Let us define the **congestion signal probability** λ as the probability that the source receives a CS for an arbitrary transmitted packet. If B is the average session bandwidth obtained by the source under a FLICA algorithm, then the functional dependence of B on λ is given by

$$B(\lambda) = \frac{1}{\lambda S \left(0.5 + \sqrt{.25 + \left(\frac{2}{2C-1}\right) \frac{1}{\lambda S}}\right)} \tag{1}$$

The derivation of this result is provided in the appendix.

4 Congestion Signal Probabilities for some LIF Policies

In this section, we consider a few specific LIF policies and describe how to compute the value of λ for these policies and for a given multicast topology. Computing the value of λ is a prerequisite for analytically computing the average session bandwidth (equation (1)) attained by a session for a particular FLICA algorithm. The LIF policies that we consider here are :

- **Pass-All** : Of all the LIs received for a transmitted packet (new or retransmitted), only one is considered as a congestion signal, and this LI may be from *any* receiver in the multicast group.
- **Pass-K-of-N** : Given N receivers in a multicast group, K ($K \leq N$) receivers are designated as representatives. All LIs from the $N - K$ non-representatives are ignored. If one or more LI(s) are received from representatives for a transmitted packet, only one is considered as a CS.
- **Pass-Worst** : The receiver with the highest end-to-end loss probability is identified and all LIs from that receiver are considered as CSs. LIs from all other receivers are ignored.

Note that Pass-All and Pass-Worst are special cases of Pass-K-of-N. However, we introduce them separately for ease of exposition.

Before we proceed with the derivation of λ for these LIFs, we need to make a distinction between two models of data delivery. The first is *reliable delivery*, where the source retransmits a data packet as many times as required, until the packet has been delivered at least once to every receiver in the multicast group. In this case, the probability of generating a LI decreases with repeated retransmissions of a packet. This must be taken into account when deriving an expression for λ. The second model of data delivery is *no-retransmissions delivery* where the source does not perform any retransmissions. Loss indications (LIs) are used in this case solely for rate adjustments. This model is applicable to continuous media applications or to reliable data transfer applications with repair servers providing repairs for lost data packets. Unlike the reliable data delivery case, the probability of generating at least one LI for a packet is now the same for every packet transmitted, since no packet is transmitted more than once by the source. We now present a method for computing the value of λ in each case.

4.1 Reliable Data Delivery

Let us first consider the Pass-All LIF policy. Let $\mathcal{T} = (\mathcal{M}, \mathcal{E})$ be the multicast distribution tree spanning all receivers in a multicast group, where $\mathcal{M}$ is the set of nodes, $\mathcal{E}$ is the set of directed edges in the tree and all receivers are attached to leaf nodes of the tree. Let $S \in \mathcal{M}$ denote the root of the tree (i.e. the node closest to the source) and let $c(n),\ n \in \mathcal{M}$, denote the set of child nodes of node n in $\mathcal{T}$.

Let $R^{\mathcal{T}}(n)$ denote the number of times a packet has to be transmitted to a node $n \in \mathcal{T}$, until it has been received at least once by all receivers downstream from n. Let $F_n^{\mathcal{T}}$ be the probability distribution function for $R^{\mathcal{T}}(n)$ and p_n be the loss probability of a packet at node n. The expression for $F_n^{\mathcal{T}}$ is given in [3] :

$$F_n^{\mathcal{T}}(i) \ = \ P[R^{\mathcal{T}}(n) \leq i] = \begin{cases} 1 - p_n^i, & n \text{ is a leaf node,} \\ \sum_{u=0}^{i-1} \binom{i}{u} p_n^u (1-p_n)^{i-u} \prod\limits_{k \in c(n)} F_k^{\mathcal{T}}(i-u), & \text{otherwise.} \end{cases} \tag{2}$$

Therefore,

$$E[R^{\mathcal{T}}(S)] = \sum_{i=0}^{\infty} (1 - F_S^{\mathcal{T}}(i)) \tag{3}$$

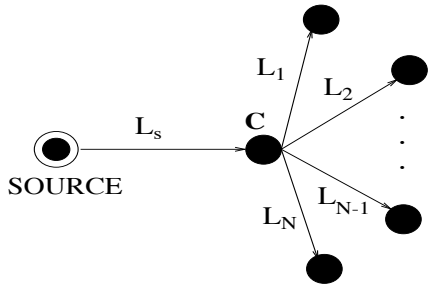

Figure 2: Modified Star Topology, q = loss probability on L_s. p_i = loss probability on L_i.

Since this is the expected number of times that a packet will be transmitted, the expected number of times that at least one receiver will lose the packet is $E[R^{\mathcal{T}}(S)] - 1$. For each of these times, the source will receive a CS. Hence the expected number of CSs generated per $E[R^{\mathcal{T}}(S)]$ packets is $E[R^{\mathcal{T}}(S)] - 1$, and

$$\lambda = 1 - \frac{1}{E[R^{\mathcal{T}}(S)]} \tag{4}$$

Now consider the Pass-K-of-N LIF. Let $\mathcal{G} = (\mathcal{M}', \mathcal{E}')$ be the multicast distribution tree for only the representatives , where $\mathcal{M}' \subset \mathcal{M}$ and $\mathcal{E}' \subset \mathcal{E}$. Note that for the Pass-Worst LIF, the tree $\mathcal{G}$ consists of the single end-to-end path from the source to the lossiest receiver. Then the expected number of times a packet has to be transmitted in order to be delivered at least once to each of the representatives is

$$E[R^{\mathcal{G}}(S)] = \sum_{i=0}^{\infty}(1 - F_S^{\mathcal{G}}(i)) \tag{5}$$

where $F_S^{\mathcal{G}}(i)$ is defined by equation (2). Hence λ is given by

$$\lambda = \frac{E[R^{\mathcal{G}}(S)] - 1}{E[R^{\mathcal{T}}(S)]} \tag{6}$$

Note that in the case that $K = N, \mathcal{G} = \mathcal{T}$, the Pass-K-of-N LIF reduces to Pass-All and (6) reduces to (4). We now describe how to apply the above technique to compute λ for two simple topologies – a "modified star" (Figure 2) and a complete binary tree. The multicast source is connected to the center C of the star by a link L_s and each receiver i is connected to C by a link $L_i, i = 1, \cdots, N$. A packet loss on L_s affects all receivers, while a packet lost on L_i affects only receiver i. Let q be the probability of packet loss on L_s and let p_i be the loss probability on L_i. Then from equation (2) we can derive

$$F_S^{\mathcal{T}}(i) = \sum_{u=0}^{i-1} \binom{i}{u} q^u (1-q)^{i-u}(1-p^{i-u})^N$$

For the Pass-All LIF, λ is given by (3) and (4). For the Pass-K-of-N LIF, λ is given by (3), (5) and (6).

Let us next consider a complete binary tree having a uniform loss probability p at each node and with all receivers attached to the leaf nodes. Let the height of the tree be H and let us associate a level number

with each node such that the root is at level H, the leaf nodes are at level 0 and each node at level h has two children at level $h-1, h = 1, 2, \cdots, H$.

Every node in the tree has the same packet loss probability, hence for any two nodes $n1$ and $n2$ at the same level of the tree, F_{n1} and F_{n2} are identical. Let us therefore denote by $R(h)$, the number of times a packet has to be transmitted transmitted to a node at level h, till it has been received at least once by all downstream receivers. Let $F_h^{\mathcal{T}}$ be the probability distribution function for $R(h)$. Then from equation (2), we can write,

$$F_h^{\mathcal{T}}(i) \;=\; P[R^{\mathcal{T}}(h) \leq i] = \begin{cases} 1-p^i, & h \;=\; 0, \\ \sum_{u=0}^{i-1} \binom{i}{u} p^u (1-p)^{i-u} (F_{h-1}^{\mathcal{T}}(i-u))^2, & h \;=\; 1, \cdots, H. \end{cases} \tag{7}$$

For the Pass-K-of-N LIF,

$$E[R^{\mathcal{T}}(H)] = \sum_{i=0}^{\infty} (1 - F_H^{\mathcal{T}}(i))$$

and λ is given by

$$\lambda = \frac{E[R^{\mathcal{G}}(H) - 1}{E[R^{\mathcal{T}}(H)]} \tag{8}$$

For a Pass-All LIF, λ can be obtained by replacing $\mathcal{G}$ by $\mathcal{T}$ in equation (8).

4.2 No-retransmission Data Delivery

For the Pass-All LIF, let us again start with an arbitrary multicast tree $\mathcal{T} = (\mathcal{M}, \mathcal{E})$ spanning all receivers in the multicast group. Let p_n be the probability of packet loss at node $n, n \in \mathcal{M}$. Let $Q_n^{\mathcal{T}}$ be the probability that a packet transmitted to node n is lost by at least one receiver downstream from n. $Q_n^{\mathcal{T}}$ is computed recursively according to the following equation :

$$Q_n^{\mathcal{T}} = \begin{cases} p_n, & n \text{ is a leaf node}, \\ 1 - (1-p_n) \prod_{m \in child(n)} (1 - Q_m^{\mathcal{T}}), & \text{otherwise}. \end{cases} \tag{9}$$

Then,

$$\lambda = Q_S^{\mathcal{T}} \tag{10}$$

where S is the root of T.

For a Pass-K-of-N LIF, λ is given as

$$\lambda = Q_S^{\mathcal{G}} \tag{11}$$

For the modified star topology λ can be derived for a Pass-K-of-N LIF using (9) and (11) as

$$\lambda = 1 - (1-q)(1-p)^K \tag{12}$$

For the Pass-All LIF, $\mathcal{T} = \mathcal{G}$ and λ is obtained from (12) by replacing K with N.

For a complete binary tree $\mathcal{T}$ with uniform node loss probability p, we follow the same approach as before and define $Q_h^{\mathcal{T}}$ as probability that a packet transmitted from any node at level h of the tree is lost by at least one downstream receiver. From (9), we have

$$Q_h^{\mathcal{T}} = \begin{cases} p, & h \;=\; 0, \\ 1 - (1-p)(1 - Q_{h-1}^{\mathcal{T}})^2, & h = 1, \cdots, H. \end{cases} \tag{13}$$

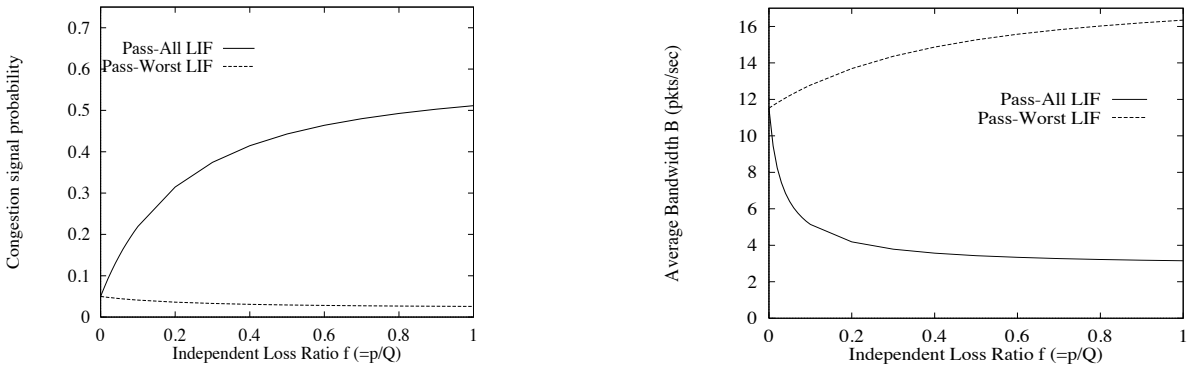

Figure 3: Congestion signal probability (λ) and average session bandwidth(B) vs. independent loss ratio (f) for reliable data delivery with a modified star topology having $N = 50$. $Q_i = Q = 0.05, i = 1, \cdots, 50$. Algorithm : FLICA with $C = 2$, S = 0.2 sec.

For the Pass-K-of-N LIF,

$$\lambda = Q_H^{\mathcal{G}} \tag{14}$$

For the Pass-All LIF, λ is obtained by replacing $\mathcal{G}$ with $\mathcal{T}$ in equation 14.

5 Case Studies

In this section, we study the behavior of some specific FLICA algorithms for the modified star and complete binary tree topologies by considering different network loss scenarios. The metric used for evaluating the performance of the algorithms is the average session bandwidth B. B is computed using equation (1), with the value of λ computed according to the method described in Section 4. The purpose of this study is to gain insights into the effect of the LPM problem and into its possible solutions. We also study the performance of the congestion control algorithm proposed for the PGM protocol with the goal of understanding whether, and to what extent, it is affected by the LPM problem.

5.1 FLICA algorithms

The FLICA algorithms studied here use different loss indication filters (LIFs) but the same rate adjustment algorithm with $C = 2.0$ and $S = 0.2$ sec.

Let the modified star (Figure 2) have $N = 50$ receivers. Let Q_i be the end-to-end loss probability for receiver i. With p_i and q as defined earlier, we then have $p_i = (Q_i - q)/(1 - q)$. Let us define the **independent loss ratio** as $f_i = p_i/Q_i$. This is a measure of the fraction of independent (i.e. not spatially correlated with any other receiver) loss for receiver i.

Let us first consider identical loss probabilities for all receivers, i.e. $Q_i = Q = 0.05, i = 1, \cdots, 50$. This implies that the independent loss ratio is also the same for all receivers. Let $f_i = f, i = 1, \cdots, 50$. Figure 3 illustrates the dependence of λ and B on f in the case of applications requiring reliable data delivery. Figure 4 shows the same for applications using no-retransmission data delivery.

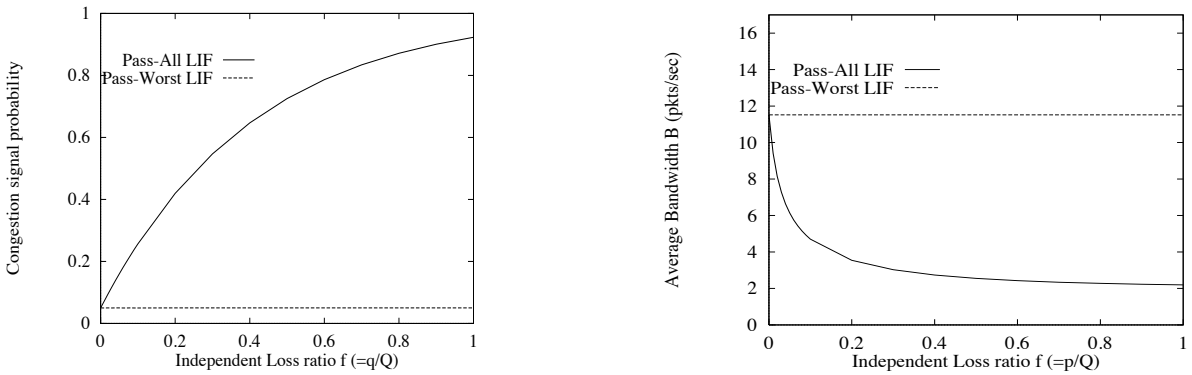

Figure 4: Congestion signal probability (λ) and average session bandwidth(B) vs. independent loss ratio (f) for no-retransmission data delivery with a modified star topology having $N = 50$. and $Q = Q = 0.05$, $i = 1, \cdots, 50$. Algorithm : FLICA with $C = 2$, S = 0.2 sec.

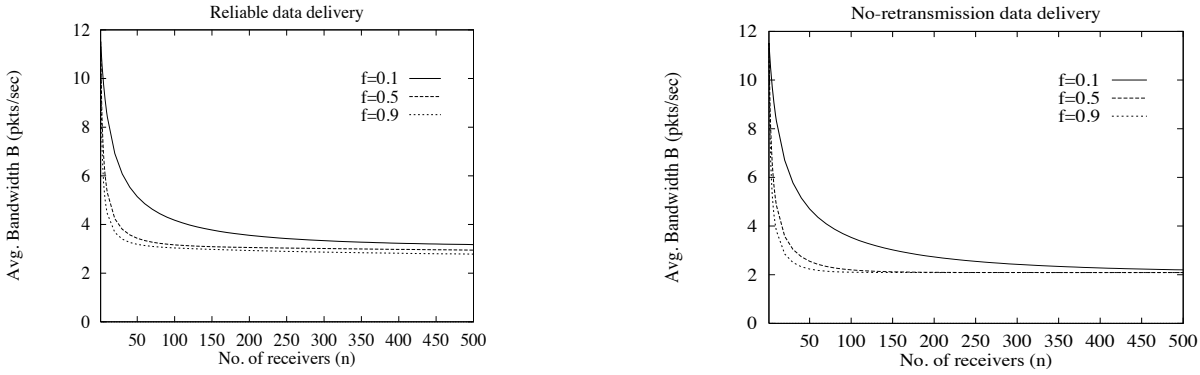

Figure 5: Average session bandwidth(B) vs. no. of receivers (N) for (A) reliable data delivery and (B) no-retransmission data delivery with a modified star topology. $Q_i = Q = 0.05$, $i = 1, \cdots, N$. Algorithm : FLICA with $C = 2$, $S = 0.2$ sec, Pass-All LIF

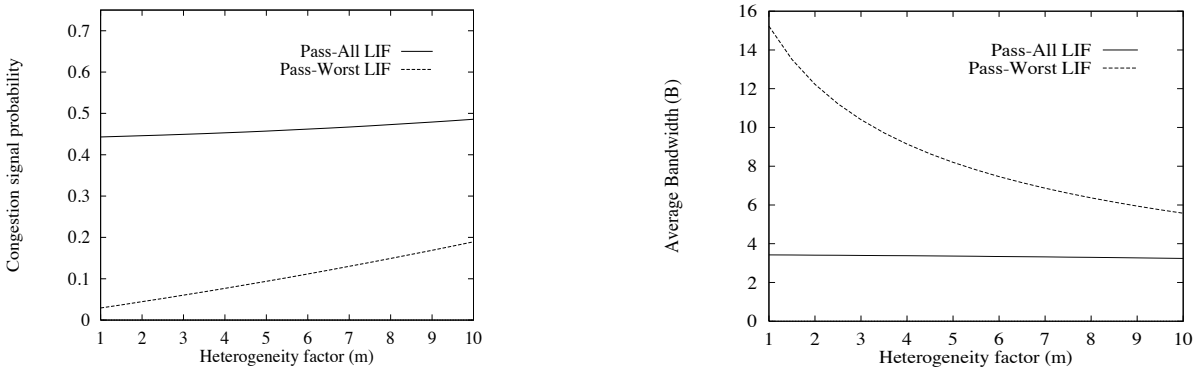

Figure 6: Congestion signal probability (λ) and average session bandwidth(B) vs. heterogeneity factor (m) for reliable data delivery with a modified star topology having $N = 50$. $q = 0.02546$, $p_i = 0.025$, $i = 2, \cdots, N$, $p_1 = m * p$. Algorithm FLICA with $C = 2$, $S = 0.2$ sec.

We observe that for a Pass-All LIF, there is a sharp increase in the value of λ with increasing f. The effect is less significant for reliable delivery since the probability of getting a NAK decreases with repeated retransmissions of a packet. On the other hand, with a Pass-Worst LIF (in this case, any one receiver can be picked as the representative to track), there is no such sharp increase in λ. For no-retransmission delivery, λ is simply the end-to-end loss probability for any receiver; hence it remains invariant with f. Interestingly, in the reliable delivery case with a Pass-Worst LIF, λ decreases with increasing f. The reason for this is as follows. Once the tracked receiver has received a packet, it ignores all subsequent retransmissions of the same packet. Hence if any such retransmission is lost by one of the other receivers, the source does not receive a CS for it. As the spatial loss correlation decreases, there is is greater chance that the tracked receiver receives a packet which one or more of the other receivers have lost. Since no CS is generated for any of the subsequent retransmissions, λ decreases.

For the Pass-All LIF, the increase in λ with f leads to a drastic reduction in the bandwidth (B) actually used by the multicast session, since B is a decreasing function of λ. Significantly, most of this reduction takes place between $f = 0.0$ and $f = 0.1$, indicating that even small amounts of uncorrelated loss can have harmful consequences for a multicast session's average bandwidth. We also observe that, with a Pass-Worst LIF, there is no such degradation in B, since λ remains more or less unchanged.

From Figure 5, we observe that B scales poorly with the number of receivers (N) for a Pass-All filter. The degradation is quite drastic even when the independent loss ratio, f, is as small as 0.1. This clearly shows the scalability problem introduced by loss path multiplicity for a FLICA algorithm that responds to LIs from all receivers.

Let us next consider a loss scenario where the loss probability on the arm leading to one of the receivers is higher than the rest. Without loss of generality, let us assume that this is receiver 1. We choose $q = 0.05$ and $f_i = 0.5$ for $i = 2, \cdots, N$. Hence $p_i = p = 0.025$, $i = 2, \cdots, N$. Let us define a heterogeneity factor m such that $p_1 = m * p$. Figure 6 shows the dependence of λ and B on m for applications requiring reliable delivery. Figure 7 shows the same for applications where the source does not retransmit lost data. We observe that for a Pass-All LIF, λ is large even when $m = 1$, because of the LPM problem. Hence, the increase in λ with m is not significant. Of course, with the Pass-Worst filter, the source tracks only receiver

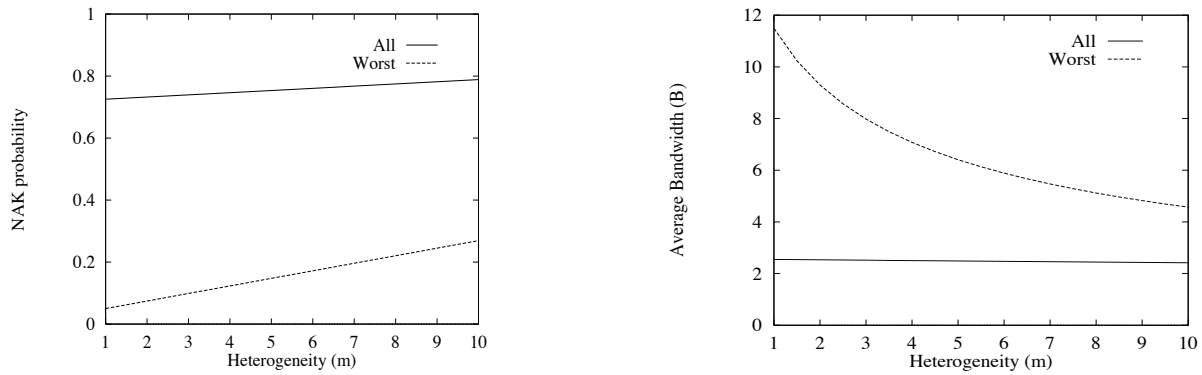

Figure 7: Congestion signal probability (λ) and average session bandwidth(B) vs. heterogeneity factor (m) for no-retransmission data delivery with a modified star topology having $N = 50$. $q = 0.02546$, $p_i = 0.025$, $i = 2, \cdots, N$, $p_1 = m * p$. Algorithm FLICA with $C = 2$, $S = 0.2$ sec.

1 and λ is linear in m. From the B vs. m plot, we observe that a Pass-Worst filter significantly improves the bandwidth share of a multicast session over a Pass-All filter. This is because the Pass-Worst filter is able to eliminate the LPM problem.

We now consider a complete binary tree with all receivers at the leaf nodes and the same loss probability $p = 0.01$ on each link. Figure 8 shows the dependence of λ and B on the height of the tree, H for applications requiring reliable data delivery, and Figure 9 shows the same for applications requiring no-retransmission delivery. As in the case of the modified star, when the source uses a Pass-All LIF (i.e. tracks all receivers), λ increases very sharply with H, resulting in a sharp decrease in B. This again is a manifestation of the LPM problem, since increasing H increases the number of receivers. When the source uses a Pass-Worst LIF, it can pick any one receiver to track, since the loss probabilitiy is identical on all end-to-end paths. In this case also, there is a gradual degradation in the value of B with increasing H. However, this reduction in B is inevitable, since increasing H increases the number of links on every end-to-end path, consequently the end-to-end loss probability increases. Note that for applications requiring no retransmissions from the source, λ is higher (for reasons discussed earlier) for the same value of h and consequently, the value of B is lower.

From the results so far, we infer that the LPM problem arises from tracking LIs from a large number of receivers when not all the losses occur on the same end-to-end network path in a multicast tree. So it is possible that using a representative scheme, where the source tracks only K of N receivers, may alleviate the LPM problem to a certain extent. We now evaluate the performance of some such schemes by considering FLICA algorithms with a Pass-K-of-N LIF for a modified star (Figure 2). Let us choose (without loss of generality) receivers $1, \cdots, k$ to be the representatives. In addition to a FLICA algorithm with $C = 2$, we consider FLICA algorithms with $C = K + 1$. As the value of K increases, λ is expected to increase due to the LPM problem. For a fixed C, this has the effect of reducing B. However, if the source reacts less aggressively to each CS by using a larger value of C, then that should partially compensate for the the increase in B with λ. Note that when there is a single representative, the value of C reduces to 2.

We consider three different loss scenarios. In the first case all receivers experience the same end-to-end loss probability. We choose $Q_i = Q = 0.05$ and $f_i = f = 0.5$. Hence $q = 0.02546$ and $p_i =$

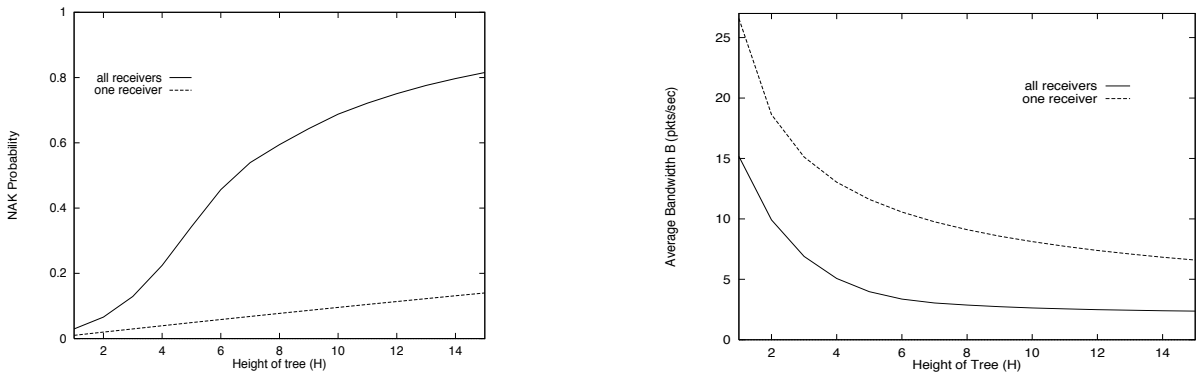

Figure 8: Congestion signal probability (λ) and average session bandwidth(B) for a complete binary tree vs. height of tree (h) for reliable data delivery. Loss probability on each link of the tree is $p = 0.01$. Algorithm FLICA with $C = 2$, S = 0.2 sec.

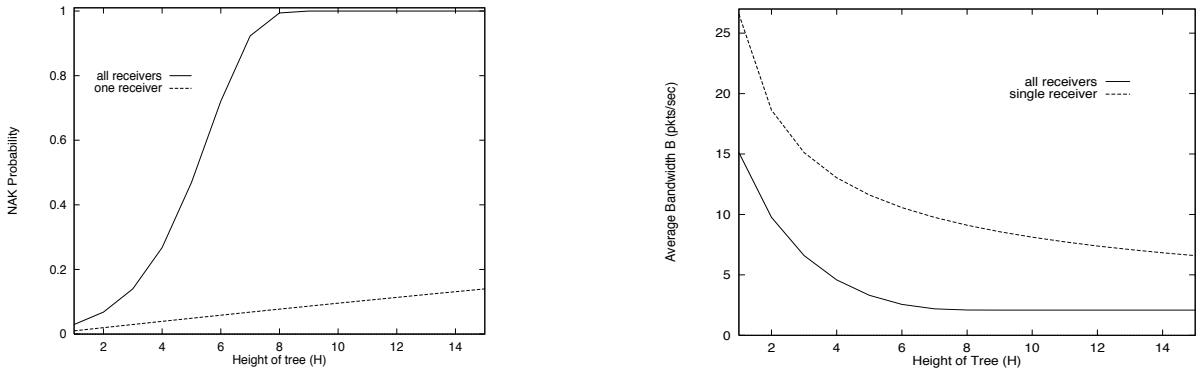

Figure 9: Congestion signal probability (λ) and average session bandwidth(B) for a complete binary tree vs. height of tree (h) for no-retransmission data delivery. Loss probability on each link of the tree is $p = 0.01$. Algorithm FLICA with $C = 2$, S = 0.2 sec.

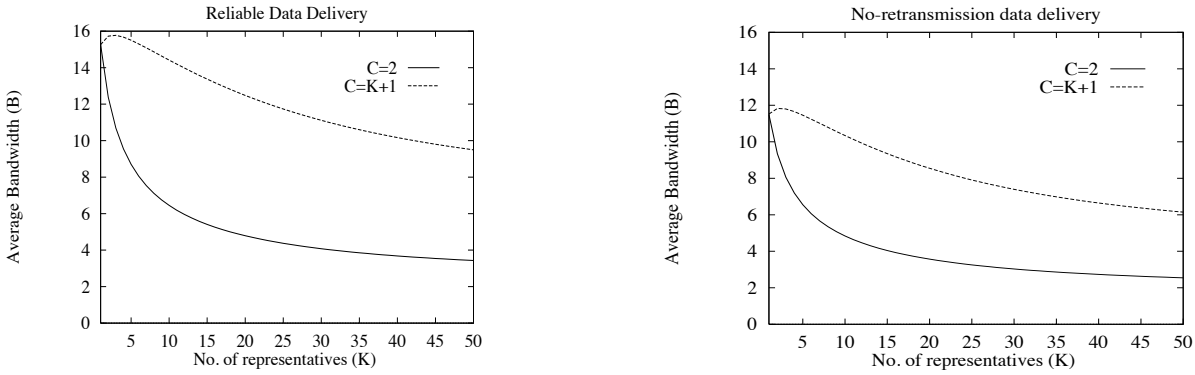

Figure 10: Average session bandwidth(B) vs. no. of representatives (K) for (1) reliable data delivery and (2) no-retransmission data delivery with a modified star topology having $N = 50$. End-to-end loss probability for each receiver $Q = 0.05$, with the independent loss ratio $f = 0.5$.

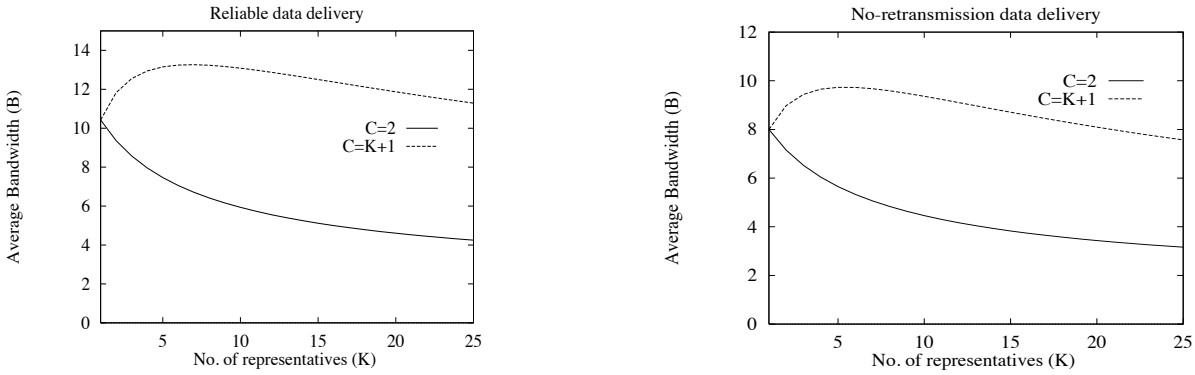

Figure 11: Average session bandwidth(B) vs. no. of representatives (K) for (1) reliable data delivery and (2) loss-tolerant data delivery with a modified star topology having $N = 50$. Shared loss probability $q = 0.025$, $p_1 = 0.0769$, $p_i = 0.02546,\ i = 2, 3, \cdots, N$.

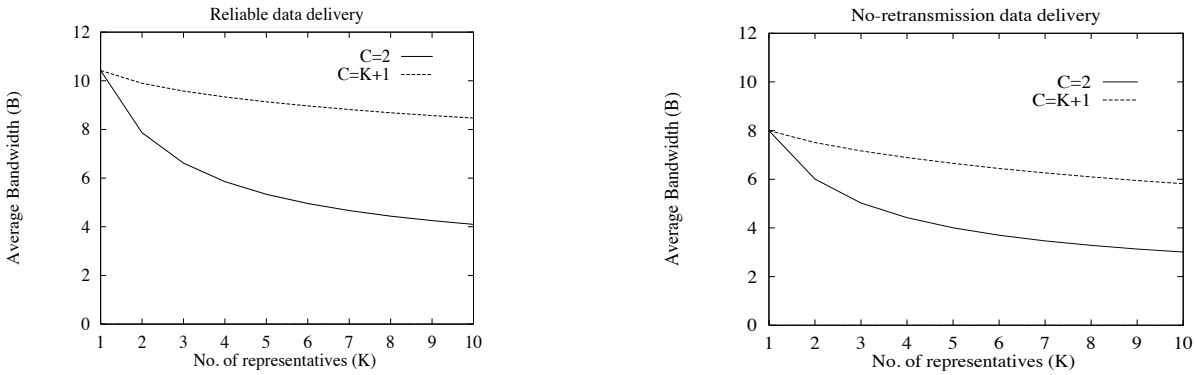

Figure 12: Average session bandwidth(B) vs. no. of representatives (k) for (1) reliable data delivery and (2) loss-tolerant data delivery with a modified star topology having $N = 50$. For all receivers $q = 0.025$. Independent loss probability $p_i = 0.02546$ for a non-representative. For a representative, $p_i = 0.0769$. Algorithm FLICA with $C = 2, S = 0.2$ sec.

0.025. We observe from Figure 10 that the degradation in B with increasing K is less severe when $C = K + 1$ than when $C = 2$, proving our intuition to be correct. In both cases however, as the number of representatives increases beyond a certain value, there is a considerable decrease in B. This implies that only a representative scheme with a very small number of representatives ($K \leq 5$) can counter the effect of the LPM problem.

The second loss scenario (Figure 11) differs from the first in that $p_1 = 3 * p_i,\ i = 2, \cdots, N$. The loss probability values chosen are $q = 0.02546$, $p_1 = 0.075$ and $p_i = 0.025$, $i = 2, \cdots, N$. As expected, B decreases with K when $C = 2$. However, when $C = K + 1$, B initially increases with increasing K up to about $K = 5$, before starting to decrease. The reason is as follows. For $2 \leq K \leq 5$, the source does observe a higher λ when K increases. However this is more than compensated by the less aggressive reaction to every individual CS, which is a result of the increase in the value of C. Hence the average session bandwidth, B, actually increases.

Finally we consider a case where each representative has a higher loss probability than each non-representative. Specifically, $q = 0.02546$, $p_i = 0.075$, $i = 2, \cdots, k$, and $p_i = 0.025$, $i = k + 1, \cdots, N$ (Figure 12). Again, we observe that B decreases as k increases, though the reduction is significantly less when $C = K + 1$ and the number of number of representatives is small ($K \leq 10$).

From these three examples, we conclude that the effect of loss path multiplicity can be partially alleviated by using a representative scheme. However the average session bandwidth is sensitive to the choice of representatives and the choice of the rate adjustment algorithm. These choices are difficult since they must be tailored to the observed network loss conditions. At the same time, we observe that these complications can be avoided and max-min fair sharing of bandwidth can be achieved by having each multicast source choose its worst receiver as the single representative ($K = 1$).

5.2 PGM congestion control algorithm

The LPM problem is not restricted to the family of FLICA algorithms alone. It affects any source-based rate control algorithm that reduces the source's rate in response to LIs from receiver without due consideration

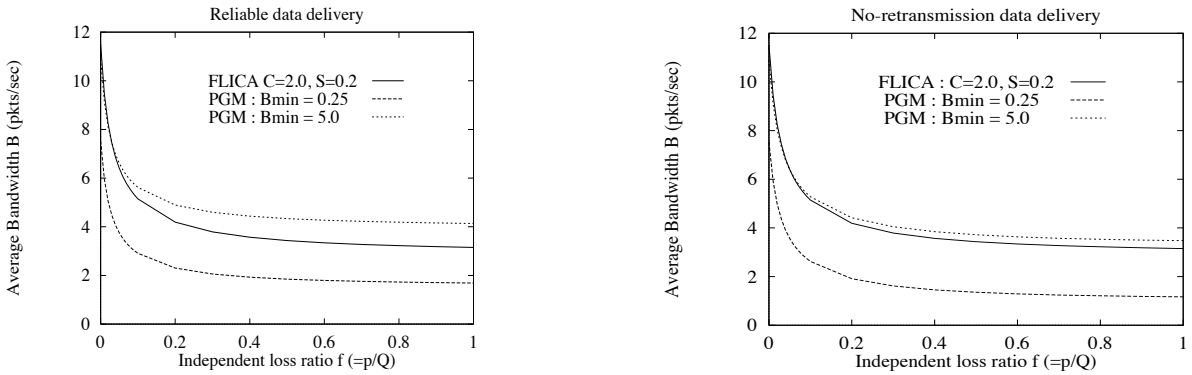

Figure 13: Average session bandwidth(B) vs. Independent loss ratio f for (1) reliable data delivery and (2) loss-tolerant data delivery with a modified star topology having $N = 50$ and $Q = 0.05$. Algorithms : (1) FLICA with $C = 2, S = 0.2$ sec, (2) PGM with $B_{min} = 0.25$ and $B_{min} = 5.0$, $R = 2.0$.

of the existence of multiple loss paths in a multicast tree. In order to illustrate this, we next consider the effect of the LPM problem on the congestion control algorithm proposed for the PGM protocol.

The PGM protocol [15] has proposed the idea of using active network elements to aggregate/selectively discard loss indications, in the form of negative acknowledgments (NAKs), as they propagate up the multicast tree from receivers towards the source. Therefore, a source will ideally receive exactly one loss indication (NAK) per lost packet. [15] also proposes a preliminary congestion control algorithm that adjusts the source transmission rate as follows :

$$\begin{array}{lll} \text{On every NAK} & : r \leftarrow B_{min}, & \\ \text{Thereafter, per time } R & : r \leftarrow 2 * r\ , & r < 0.5\, B_c, \\ & \leftarrow r + 1, & 0.5\, B_c \leq r < B_c, \\ & \leftarrow B_{max}, & \text{otherwise.} \end{array}$$

where B_{min} and B_{max} are predefined values, R is the worst-case round-trip time from any receiver to the source, and B_c was the value of r when the last NAK was received.

Thus the rate control algorithm operates by reducing r to a minimum every time a NAK received. Then the rate is exponentially increased to half the transmission rate in use when the last NAK was received (**slow-start**) and thereafter increased linearly (**congestion avoidance**) until a predefined maximum is reached. The rate remains at this maximum value (**saturation**) until the next NAK is received. Let us assume ideal operation of the PGM NAK aggregation algorithm. Hence the source receives no more that one NAK per transmitted packet (new or retransmitted) and it reduces its rate in response to every such NAK. Hence the congestion signal probability λ at the source is the same as that of a source using a Pass-All loss indication filter with no aggregation of NAKs in the network. The expression for λ for the latter case has already been derived in section 4. The functional dependence of B on λ in this case given by is :

$$B(\lambda) = \begin{cases} \frac{6}{\lambda R\left[\left(-5+\sqrt{25+24(B_{min}+1/(\lambda R))}\right)+2\log_2\left(-5+\sqrt{25+24(B_{min}+1/(\lambda R))}\right)-2\log_2 B_{min}\right]}, & \lambda \geq \lambda_s, \\ \frac{B_{max}}{\lambda R[(B_{max}^2)/8+B_{max}(\log_2 B_{max}-\log_2 B_{min}-5/4)+B_{min}]+1}, & \text{otherwise.} \end{cases} \quad (15)$$

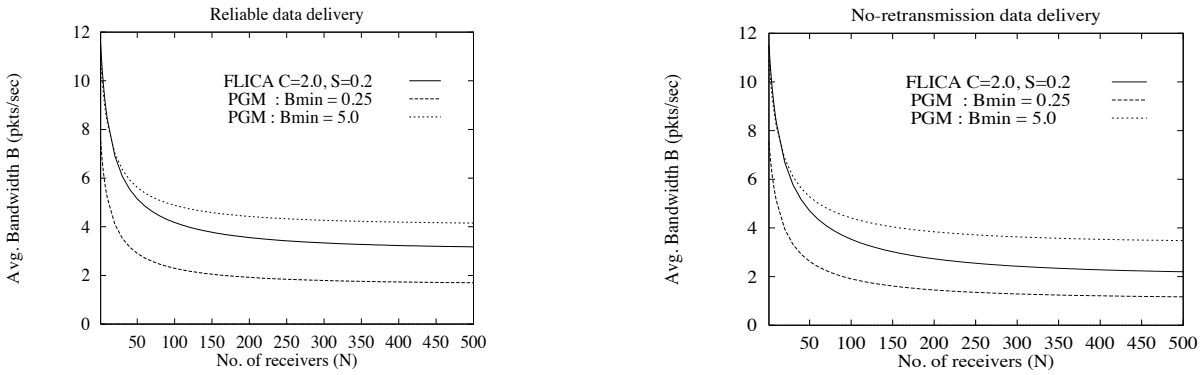

Figure 14: Average session bandwidth(B) vs. no. of receivers (N) for (1) reliable data delivery and (2) loss-tolerant data delivery with a modified star topology having $N = 50$ and $Q = 0.05$ and $f = 0.1$. Algorithms : (1) FLICA with $C = 2, S = 0.2$ sec, (2) PGM with $B_{min} = 0.25$ and $B_{min} = 5.0, R = 2.0$.

where

$$\lambda_s = \frac{1}{R\ (3B_{max}^2/8 + 5B_{max}/4 - B_{min})} \tag{16}$$

A detailed derivation of this result is provided in the appendix.

Let us first consider the effect of independent loss for a modified star topology with $N = 50$ and identical loss probabilities for all receivers , i.e., $Q_i = 0.05, i = 1, \cdots, 50$. The dependence of λ on f has already been shown in Figures 3 and 4. Figure 13 shows the corresponding variation in the value of B for PGM. The round trip time R is chosen to be 0.2 sec and two different values of B_{min}, 0.25 and 5.0, are considered. Note that if we assume that $B_{max} = 2^8 B_{min}$ following the recommendation in [15], then the values of λ_s (equation 16) are 0.0031 and almost zero respectively. The bandwidth plot for a FLICA algorithm with $C = 2.0, S = 0.2$ seconds and a Pass-All LIF, is also shown for the purpose of comparison.

We observe that the effect of independent loss on B for PGM is very similar to what we have observed for the FLICA algorithms earlier. There is a sharp decrease in the value of B with increasing f, with most of this reduction occurs between $f = 0$ and $f = 0.2$. In addition, we observe that the value of B increases with B_{min} for a given f. This is because increasing the value of B_{min} leads to a smaller reduction in the source's rate for every NAK received, hence the session's overall bandwidth share increases.

In Figure 14 we study the scalability of B with the number of receivers N, keeping f fixed at 0.1. Even with such a small probability of independent loss per receiver (≈ 0.005), the value of B decreases rapidly with increasing N.

6 A Simulation Study of Bandwidth Sharing

The results in the last section indicate that the LPM problem can severely reduce the average session bandwidth of a multicast session. Representative schemes can partially alleviate the problem, but may not be able to eliminate it altogether. At the same time, tracking the worst receiver is a promising approach for ensuring max-min fair bandwidth sharing. In this section, we explore this approach more carefully through simulation. We will see that it is indeed possible to eliminate the LPM problem and ensure max-main fairness

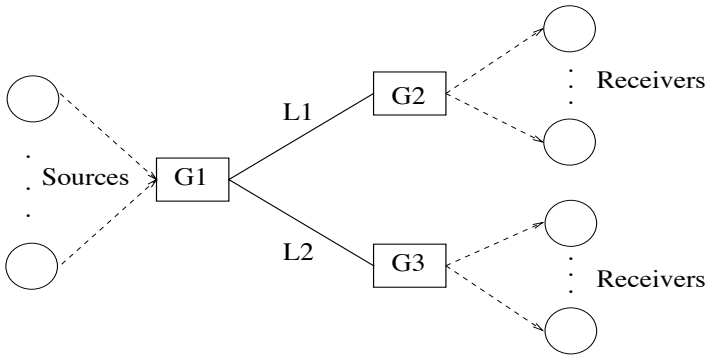

Figure 15: Star network with two arms.

by tracking the worst receiver, provided all the sessions use the same rate adjustment algorithm. However, fair sharing may not be possible if a multicast session mistakenly tracks a receiver other than the worst one. Therefore it is important that the source correctly identifies the the most congested path in its tree and then chooses a receiver at the end of that path as the one to track. The main difficulty in doing so arises from changes in network traffic conditions in different parts of the multicast tree. The source needs to be aware of a change in the congestion level on any path in its tree, so that it can determine the path that is *currently* the most congested one. We will describe a protocol for tracking the most congested path in a multicast tree and illustrate its behavior through simulation.

An event-driven simulator has been used to simulate two simple networks – a two-armed star and a two-link tandem, that are shared by a number of unicast and multicast sessions. Every session, unicast or multicast, has an infinite data source and uses a FLICA algorithm with $C = 8$ and $S = 500$ msec. We assume that data packets are never reordered, though they may be lost due to buffer overflow at the gateways. Loss indications (LIs) are in the form of negative acknowledgment (NAKs). The reverse path used by these NAKs is different from the forward path for data packets and NAKs are never lost or reordered. Lost data is never retransmitted by the source, hence NAKs are used only for the purposes of loss detection and rate adjustment at the source. This corresponds to the no-retransmission data delivery model described earlier. We also assume that the propagation delay on the reverse path is variable, but that the distribution of reverse path propagation delays is the same for all sessions. The bandwidth share of a session is measured in terms of the **average transmission rate, r** which is defined as follows. If the source transmits b packets in the interval $[t_1, t_2]$, then $r = b/(t_2 - t_1)$.

6.1 Star Network

The 2-armed star network consists of gateways $G1$, $G2$ and $G3$, connecting links $L1$ and $L2$, as shown in Figure 15. Each of links $L2$ and $L3$ has a bandwidth of 300 packets/second. Each gateway uses a FIFO service discipline and has a buffer size of 75. All sessions have their source connected to $G1$. Every unicast session has its receiver connected to either $G2$ or $G3$, whereas every multicast session has a receiver connected to each of $G2$ and $G3$. Thus each multicast session consists of two non-overlapping end-to-end paths over $L1$ and $L2$ respectively.

		Transmission Rate (pkts/sec)		
	Session Groups	Mean	Max	Min
Simulation 1	Multicast	29.8	30.8	29.3
	Unicast over L1	30.2	30.9	29.4
	Unicast over L2	30.3	30.9	29.4
Simulation 2	Multicast	20.9	21.1	20.6
	Unicast over L1	20.9	21.2	20.7
	Unicast over L2	39.9	40.5	39.2
Simulation 3	Multicast	30.0	30.2	29.4
	Unicast over L1	17.1	16.9	17.3
	Unicast over L2	30.5	30.8	30.2

Table 1: Transmission rates (packets/second) of unicast and multicast sessions for simulations $1, 2$ and 3.

Simulation 1 involves five multicast sessions spanning $L1$ and $L2$, five unicast sessions over $L1$ and five unicast sessions over $L2$. The loss indication filter (LIF) at every multicast source is designed such that all NAKs from the receiver attached to $G2$ pass through while no NAK from the other receiver do so. In effect, every multicast session tracks NAKs from only one of two equally congested paths. All the sessions are allowed to transmit packets for 2200 seconds, with the session starting times being staggered over the first second. Table 1 shows the mean, maximum and the minimum value of the average transmission rates for three groups of sessions : the five multicast sessions, the five unicast sessions over $L1$ and the five unicast sessions over $L2$. The measurement interval is taken to be $[200 \text{ sec}, 2000 \text{ sec}]$. We observe that each session receives approximately the same share of bandwidth on both $L1$ and $L2$, implying that it is possible to achieve, or at least approach, max-min fair bandwidth sharing in this case.

In simulation 2, five additional unicast sessions are started on $L1$, thereby making it more congested than $L2$. Each multicast session still uses the same LIF as in simulation 1, thereby tracking the more congested path of its two paths. We observe (Table 1) that on $L1$, all sessions (unicast or multicast) receive approximately an equal share ($\approx$ 20 packets/sec) of the bottleneck bandwidth. There is less traffic on $L2$, hence more available bandwidth, however each multicast sessions is constrained to consume $\approx$ 20 packets/sec on *all* of its end-to-end paths. This leaves an available bandwidth of about 200 packets/sec of bandwidth available on $L2$, which is then shared equally among the five unicast sessions traversing that link. Thus by using the same control algorithm at every source and by determining a multicast session's share by its most congested path, max-min fairness has been realized.

Simulation 3 differs from simulation 2 in that the LIF for each multicast session lets through NAKs only from the receiver attached to $G3$ and filters all NAKs from the one attached to $G2$. Hence each multicast session now regulates its rate according to the less congested of its two paths. We observe that $L2$'s bandwidth is shared equally among the five multicast sessions and the five unicast sessions traversing it. But due to this, every multicast session is able to attain a rate of about 30 packets/sec over $L1$ as well. This leaves each unicast session on $L1$ with a share of about 17 packets/sec and max-min fairness is not realized. This observation emphasizes the importance of each multicast session being able to correctly identify its most congested path. However, the available bandwidth on different paths of a multicast tree may be time variant; hence, a one-time identification of the most congested path may not be not sufficient. A multicast source has to monitor all its end-to-end paths, determine which one is *currently* the most congested and then choose a receiver at the end of that path to track.

		Transmission Rate (pkts/sec)		
200-1200 sec.	Session Groups	Mean	Max	Min
	Multicast	21.1	21.4	20.8
	Unicast over L1	20.8	21.1	20.3
	Unicast over L2	39.7	41.2	38.8
1400-2400 sec.	Multicast	16.1	16.3	15.7
	Unicast over L1	23.1	24.2	22.0
	Unicast over L2	16.2	16.7	15.8

Table 2: Transmission rates (packets/second) of unicast and multicast sessions for simulation 4.

We next outline the design of an idealized protocol for doing this. In this protocol, every receiver in a multicast group to monitor packet losses on its end-to-end path and maintains a loss probability estimate p. On receiving packet i, this estimate is updated as follows :

$$p_{i+1} \leftarrow \begin{cases} (1-\alpha)p_i + \alpha, & \text{if packet } i \text{ is detected to be lost,} \\ (1-\alpha)p_i & \text{if packet } i \text{ is received successfully.} \end{cases} \tag{17}$$

where α is a predefined constant and p_1 is one if the very first packet is detected to be lost, and zero otherwise. Every receiver periodically reports the value of p to the source. The source uses a Pass-Worst LIF that remembers the identity of the receiver currently reporting the highest value of p, and allows only NAKs from that receiver to pass through. A change in the congestion condition on any end-to-end path is reflected in the value of p reported by a receiver at the end of that path. Hence the source is able to detect such changes and always regulate its rate according to the worst end-to-end path.

Simulation 4 illustrates how such an LIF can ensure max-min fair sharing of bandwidth, even under changing network conditions. In this simulation, the setting is initially identical to simulations 2 and 3, hence $L1$ is the more congested of the two links. However, at $t = 1200$ second, a set of ten unicast sessions are started on $L2$, making it more congested than $L1$. The value of α has been chosen as 0.01. From the result of Table 2 it is clear that max-min fair sharing of bandwidth both before and after the onset of additional congestion on $L2$. This is made possible because the LIF at each multicast source is able to always identify the most congested path. So for the interval $[200$ sec, 1200 sec$]$, it identifies the receiver attached to $G1$ as the worst, but by $t = 1400$ sec, it has switched to the receiver attached to $G2$.

6.2 Tandem Network

In the case of the star network, there is no spatial loss correlation between two receivers in a multicast session. We now briefly consider a two link tandem network (Figure (16)) to show that the proposed Worst-Pass LIF works well in the presence of spatial loss correlation. Gateways $G1$, $G2$ and $G3$ connect links $L1$ and $L2$, which have capacities 300 packets/sec and 200 packets/sec respectively. Each gateway has a buffer size of 50 packets and uses the FIFO service discipline. There are five multicast sessions, each having their source connected to $G1$, one receiver connected to $G2$ and one connected to $G3$ and using $\alpha = 0.01$ for estimating p (equation (17)). Additionally, there are five unicast sessions over $L1$ only, five over $L2$ only and five over both $L1$ and $L2$.

The results in Table 3 indicate that bandwidth on the bottleneck link $L2$ is shared almost equally among the multicast sessions and the unicast sessions traversing both $L1$ and $L2$; hence each session receives about

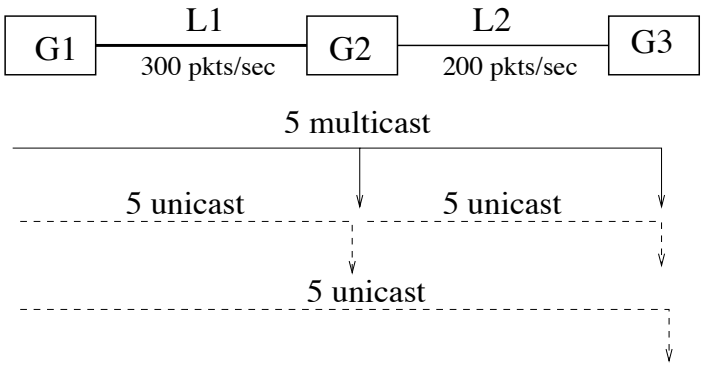

Figure 16: Tandem Network with 2 links. Arrows show direction of data flow for sessions.

	Transmission Rate (pkts/sec)		
Session Groups	Mean	Max	Min
Multicast	14.8	15.1	14.4
Unicast over L1	31.6	32.7	30.9
Unicast over L2	14.3	14.6	13.9
Unicast over L1 and L2	14.8	15.0	14.5

Table 3: Transmission rates (packets/second) of unicast and multicast sessions for tandem network.

14.8 packets/sec. Since all of these sessions also traverse $L1$, they use up about 150 packets/sec of $L1$'s bandwidth. The remaining available bandwidth on $L1$ is then shared almost equally among the five sessions traversing only $L1$. For each multicast session, the receiver connected to $G2$ is affected by losses on both $L1$ and $L2$ and reports a higher value of p than the receiver connected to $G1$. The Pass-Worst LIF at each multicast source is able to recognize the receiver attached to $G2$ as the worse of the two and filters out all NAKs from the other one. As a result, the bandwidth share of each multicast session is commensurate with the loss probabilities on the end-to-end path spanning both $L1$ and $L2$ and max-fairness is realized.

6.3 A Discussion on the Design of Tracking Protocols

We have presented the design of an idealized protocol for tracking the most congested path in a multicast tree. We now discuss two key issues, **responsiveness** and **accuracy**, that have to be considered in order to build a practical tracking protocol. The responsiveness of a tracking protocol is determined by how quickly it enables a source to react to changes in network congestion conditions. Accuracy is determined by how successful the source is in being able to correctly identify the most congested end-to-end path, based on reports from receivers. To build a protocol that is both responsive and accurate, we first need to answer the following two questions :

1. What constitutes a "significant" change in the network congestion state?
2. What is(are) the timescale(s) of changes that are of interest?

Since we propose to use end-to-end loss probability estimates to infer about the congestion state of network paths, question 1 above is equivalent to asking how we should interpret changes in the value of the loss probability estimate at a receiver. An incorrect interpretation may lead to a loss of accuracy in tracking the most congested path. At the same time, we need to minimize the response time, i.e. the interval between the time at which the change takes place and the time at which the change is detected, since the source may be tracking the "wrong" end-to-path during this interval. We propose to investigate this issue further by studying change point detection algorithms [1].

The issue of timescales is central to the design of tracking protocols. As the granularity of the timescale of changes increases, the response time of the protocol must decrease. More importantly, the accuracy of the protocol depends on the knowledge of timescale of changes that we wish to track. Let us use a simple example to illustrate why this may be so. Consider a long-lived (> 100 seconds) multicast connection having a two-armed star topology, with receivers $R1$ and $R2$ attached to the two arms. Assume that both arms of the star are equally loaded during the lifetime of the connection. Assume also that the loss process on each arm is an ON-OFF process with an ON and an OFF period each of duration 5 seconds, and with a packet loss probability of 0.10 during the ON period. With complete knowledge of this loss process, the source may wish to track only one of the two receivers during the entire duration of the session. However if the the tracking protocol tracks changes on a timescale of 5 seconds, $R1$ may be identified as the worst receiver during its ON periods, and $R2$ during its. Since the ON and OFF periods of the two receivers may not coincide, the congestion signal probability at the source will be higher than what it would be if the source always tracked one of the two end-to-end paths.

Therefore, it is important to understand temporal dependence in network packet loss [18] and characterize the loss process, choose the appropriate timescale(s) on which to detect changes in the network congestion state and then design a tracking protocol accordingly.

7 Conclusions and Future Work

In this paper, we have identified and studied the problem of loss path multiplicity that arises in the case of source-based multicast congestion algorithms. Our study indicates that, unless due attention is paid to the existence of multiple loss paths in a multicast tree, a multicast session's share of bandwidth may be severely reduced. As a result, max-min fair sharing of bandwidth among multicast and unicast sessions cannot be realized. Representative schemes may alleviate the LPM problem partially, but may not be able to eliminate it completely. We have also identified an approach for ensuring max-min fairness, in which every multicast source identifies the lossiest receiver (and hence, the most congested path) in its multicast tree and regulates its rate according loss indications from that receiver. We have described an idealized protocol for identifying and tracking the worst receiver in the presence of changing congestion levels in a network.

There are many issues that remain open for future research. The design of a practical protocol for tracking the worst receiver in a multicast group is an important one. The most important component of a tracking protocol is the loss probability estimation algorithm. An algorithm that provides accurate loss probability estimates and is responsive to changes in network conditions, is beneficial for not only multicast congestion control, but also for unicast congestion control where the source periodically adjusts its rate based on loss estimates from its receiver, [13, 6].

The issue of fairness in bandwidth sharing is a challenging problem. We have considered one possible definition of fairness in order to provide the context in which to study the LPM problem. However, there are many other issues that may need to be considered when defining fairness, eg., inter-receiver fairness [10],

TCP-friendliness [6, 17, 14, 5] providing additional incentives to multicast sessions as reward for efficient use of network bandwidth, etc.

Our results indicate that max-min fairness is achievable when all sessions, multicast and unicast, use the same rate adjustment algorithm. However, unicast sessions may use an existing congestion control algorithm like TCP, which may not be appropriate for multicast sessions (since it is ACK-based). In that case, it is important to design additive increase multiplicative decrease multicast algorithm that enables a multicast session to attain the same average rate as a unicast session using TCP, when both observe the same network loss probability.

Finally, we need to consider how to extend our approach of tracking loss indications from the worst receiver to active networking protocols. Active nodes may be able to provide support for determining, in a distributed manner, the worst end-to-end path in a multicast tree [14]. Also, the design of filters at active nodes, that will collectively perform the function of the loss indication filter in Figure 1 is an interesting problem.

Acknowledgments

The first author wishes to thank Dr. Jamal Golestani of Bell Laboratories, Lucent Technologies, for his insights into various congestion control issues([7]), which have greatly contributed to the author's understanding of the multicast congestion control problem and the subsequent identification of some of the issues studied in this paper.

The same author would also like to thank Maya Yajnik for the many fruitful discussions at various stages of this work.

References

[1] BASSEVILLE, M., AND NIKIFOROV, I. *Detection of Abrupt Changes Theory and Applications*. Prentice-Hall, Englewood Cliffs, N.J., 1993.

[2] BERTSEKAS, D. P., AND GALLAGER, R. G. *Data Networks*. Prentice-Hall, Englewood Cliffs, N.J., 1992.

[3] BHAGAWAT, P., MISHRA, P., AND TRIPATHI, S. Effect of Topology on Performance of Reliable Multicast Communication. In *Proc. IEEE Infocom'94 Conf.* (1994), pp. 602–609.

[4] CHIU, D., AND JAIN, R. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems 17* (1989), 1–14.

[5] DELUCIA, D., AND OBRACZKA, K. A Multicast Congestion Control Mechanism for Reliable Multicast. Tech. Rep. 97–685, University of Southern California, Computer Science Dept., August 1997.

[6] FLOYD, S., AND FALL, K. Promoting the Use of End-to-End Congestion Control in the Internet. *Submitted to IEEE/ACM Transactions on Networking* (1998).

[7] GOLESTANI, S. J. Private Communications, 1997.

[8] HANDLEY, M. A Congestion Control Architecture for Bulk Data Transfer. Presentation at IRTF RMRG meeting, Cannes, France, September 1997. Expires on July 8, 1998.

[9] JACOBSON, V. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM'88 Conf.* (1988), pp. 158–173.

[10] JIANG, T., AMMAR, M., AND ZEGURA, E. Inter-Receiver Fairness : A Novel Performance Measure for Multicast ABR Sessions. In *Proc. ACM SIGMETRICS'98 Conf.* (1998), pp. 202–211.

[11] LEVINE, B., PAUL, S., AND GARCIA-LUNA-ACEVES, J. Organizing Multicast Receivers Deterministically by Packet Loss Correlation. In *To Appear in Proc. ACM Multimedia'98 Conf.* (1998).

[12] MONTGOMERY, T. A Loss Tolerant Rate Controller for Reliable Multicast. Tech. Rep. NASA-IVV-97-011, West Virginia University, August 1997.

[13] PADHYE, J., FIROIU, V., TOWSLEY, D., AND KUROSE, J. Modeling TCP Throughput : A Simple Model and its Empirical Verification. In *To Appear in Proc. ACM SIGCOMM'98 Conf.* (1998).

[14] RHEE, I., BALLAGURU, N., AND ROUSKAS, G. MTCP : Scalable TCP-like Congestion Control for Reliable Multicast. Tech. Rep. TR–98–01, North Carolina State University, Department of Computer Science, January 1998.

[15] SPEAKMAN, T. E. A. Pretty Good Multicast (PGM) Transport Protocol Specification. Internet Draft draft-speakman-pgm-spec-00.txt, January 1998. Expires on July 8, 1998.

[16] TZENG, H., AND SIU, K. On Max-Min Fair Congestion Control for Multicast ABR Service in ATM. *IEEE JSAC 15*, 3 (April 1997), 545–556.

[17] VICISANO, L., RIZZO, L., AND CROWCROFT, J. TCP-like Congestion Control for Layered Multicast Data Transfer. In *Proc. IEEE Infocom'98 Conf.* (1998).

[18] YAJNIK, M., MOON, S., KUROSE, J., AND TOWSLEY, D. Measurement and Modelling of the Temoral Dependence in Packet Loss. Tech. Rep. 98-78, University of Massachusetts, Amherst, Department of Computer Science, July 1998.

Appendix

A Derivation of average session bandwidth (B) as a function of congestion signal probability (λ)

We assume, for the sake of simplicity, that Congestion Signals (CSs) at the source are equally spaced, i.e. if A be the number of packets transmitted between two successive CSs, then

$$A = 1/\lambda$$

Let us refer to the period between two successive CSs as a **CS cycle**. Let T be the duration of each CS cycle. The average bandwidth B can be expressed as

$$B = A/T$$

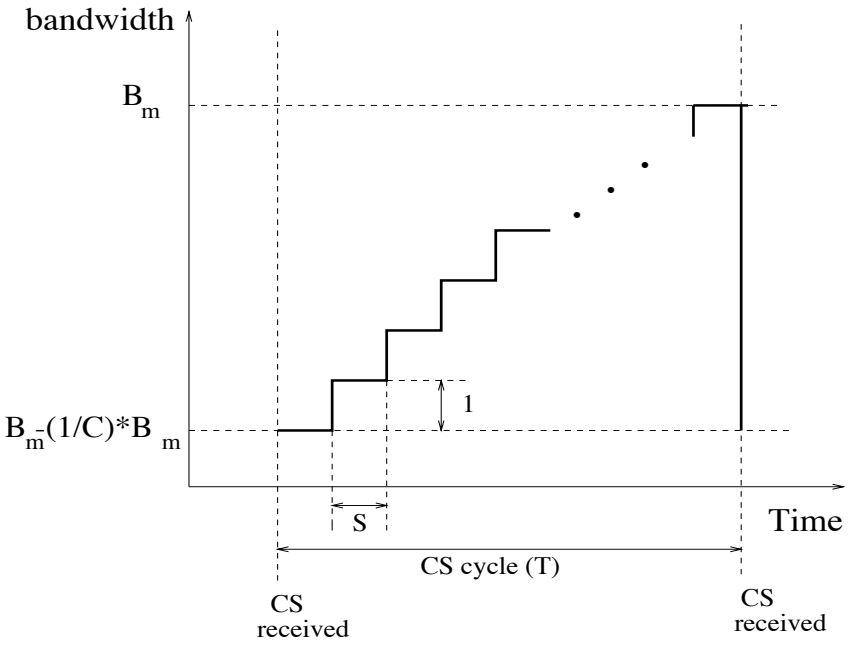

Figure 17: CS Cycle for Filtered Loss Indication-based Congestion Avoidance(FLICA) algorithms

A.1 FLICA algorithms

Let the maximum rate attained by the source in any CS cycle be B_m. Then the transmission rate at the beginning of a cycle, i.e. immediately after receiving a CS, is $B_m(1-1/C)$. Assume for simplicity that $T = NS$ where $N \in I$, i.e. the CS interval is divisible into N intervals, each of duration S. Hence, during each interval, the transmission rate remains unchanged (Figure 17) and the transmission rate during the kth interval ($1 \leq k \leq N$) is $r_k = B_m(1-1/C) + (k-1)$. Knowing $r_N = B_m$, we can write

$$B_m = B_m\,(1-1/C) + (N-1)$$

or,

$$N = (1 + B_m/C) \tag{18}$$

Hence,

$$T = S\,(1 + B_m/C) \tag{19}$$

Again, the number of packets transmitted during the kth subinterval is $S * r_k$. Hence the total number of packets transmitted during the CS cycle

$$\begin{aligned} A &= \sum_{k=1}^{N} S r_k \\ &= S \sum_{k=1}^{N} (B_m\,(1-1/C) + (k-1)) \end{aligned}$$

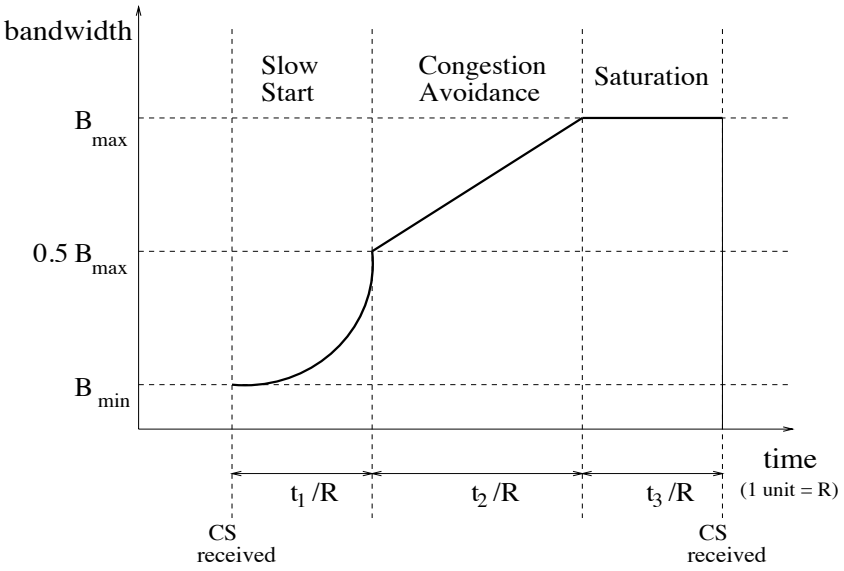

Figure 18: CS Cycle for PGM Mode 1

$$
\begin{aligned}
&= S\left(N\,B_m\,(1-1/C)+(N(N-1)/2)\right)\\
&= SB_m\left((1+B_m/C)\,(1-1/C)+1/(2C)\,(1+B_m/C)\right) \qquad \text{[using (18)]}\\
&= SB_m\left(\left(1/C-1/(2C^2)\right)B_m+(1-1/(2C))\right)
\end{aligned}
$$

Knowing that $1/\lambda$ packets are transmitted in every CS cycle, we have,

$$1/\lambda = SB_m\left(\left(1/C-1/(2C^2)\right)B_m+(1-1/(2C))\right)$$

Using the above equations we can obtain B_m, and in turn T, as a function of λ. Using this value of T in the equation $B = 1/(\lambda T)$, we obtain,

$$B = \frac{1}{\lambda\, S\,\left(0.5+\sqrt{.25+\left(\frac{2}{2C-1}\right)\,\frac{1}{\lambda S}}\right)}$$

A.2 PGM Congestion Control Algorithm

Two regimes of operation have to be considered for the PGM congestion control algorithm:

- **Mode** 1 : Let us define **saturation point** as the point in a CS cycle where the transmission rate cannot be increased any further. This is reached at time R after the the rate has reached B_{max}. Mode 1 is the regime of operation where the CS cycle sufficiently long that the source reaches the saturation point. In this case, the CS cycle in this case starts with the slow-start phase that lasts till the rate reaches $0.5B_{max}$ and continues with congestion avoidance until the rate reaches B_{max}. Thereafter the rate remains at B_{max} until the end of the CS cycle (Figure 18).

Assume that $B_{max} = 2^N B_{min},\ N \in I$ (following the recommendation in [15]. Let d_1, d_2 and d_3 be the durations of the slow-start phase, congestion avoidance phase and the saturation phase respectively. Let A_1, A_2 and A_3 be the number of packets transmitted in each of these phases. Let us now compute these values.

During the slow start-phase, the rate doubles at every time interval R, till it reaches $0.5B_{max}$ or, $2^{N-1}B_{min}$. Hence the slow-start phase can be subdivided into N intervals, each of duration R. During the the kth such interval $(1 \leq k \leq N)$, the source transmits at rate $r_k = 2^{k-1}B_{min}$. Knowing that $B_{max} = 2^N B_{min}$, we can write

$$N = \log_2 \left(\frac{B_{max}}{B_{min}} \right)$$

Hence,

$$d_1 = R \log_2 \left(\frac{B_{max}}{B_{min}} \right)$$

The total number of packets transmitted during slow-start, A_1, is

$$\begin{aligned} A_1 &= \sum_{k=1}^{N} R r_k \\ &= RB_{min} \left(1 + 2 + 2^2 + \ldots + 2^{N-1} \right) \\ &= RB_{min} \sum_{k=1}^{N} 2^{k-1} \\ &= RB_{min} \left(2^N - 1 \right) \\ &= R(B_{max} - B_{min}) \end{aligned}$$

During congestion avoidance, the rate starts from an initial value of $B_{max}/2 + 1$ and increases by 1 after every time interval R till it reaches B_{max}. So we can write $d_2 = MR$, for some non-negative integer M. The rate during the kth interval starting from the beginning of the CS cycle is $r_k = B_{max}/ + k$. Since $r_M = B_{max}$, we can write $B_{max} = B_{max}/2 + M$, or,

$$M = B_{max}/2 \tag{20}$$

Hence,

$$d_2 = RB_{max}/2$$

The number of packets transmitted during congestion avoidance, $A2$, is

$$\begin{aligned} A_2 &= \sum_{k=1}^{M} R r_k \\ &= R \sum_{k=1}^{M} (B_{max}/2 + k) \\ &= R \left((MB_{max})/2 + M(M+1)/2 \right) \\ &= RM \left(B_{max}/2 + (M+1)/2 \right), \qquad \text{[using (20)]} \\ &= (RB_{max}/2) \left(B_{max}/2 + (B_{max}/2 + 1)/2 \right) \\ &= RB_{max} \left(3B_{max} + 2 \right)/8 \end{aligned}$$

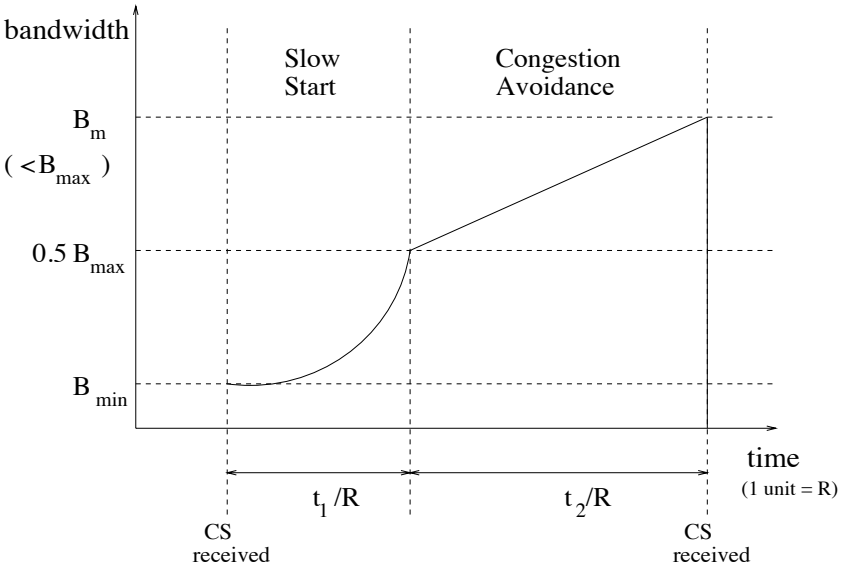

Figure 19: CS Cycle for PGM Mode 2

Since the transmission rate is constant (B_{max}) during the saturation phase,

$$d_3 = A_3/B_{max}$$

Again, the total number of packets transmitted in a CS cycle is $1/\lambda$, hence $A_3 = 1/\lambda - (A_1 + A_2)$. Therefore,

$$d_3 = (1/(\lambda B_{max}) - (R/B_{max}) \left(\frac{3}{8}B_{max}^2 + \frac{5}{4}B_{max} - B_{min}\right)$$

Again, $B = \frac{1}{\lambda(d_1+d_2+d_3)}$, which leads to

$$B = \frac{B_{max}}{\lambda R[(B_{max}^2)/8 + B_{max}\,(\log_2 B_{max} - \log_2 B_{min} - 5/4) + B_{min}] + 1}$$

- **Mode** 2 : In this mode, the CS cycle ends before the transmission rate reaches the saturation point. Let the maximum rate attained by the source in a CS cycle be B_m where $B_m \leq B_{max}$ (Figure 19). Hence, the source is in slow-start until the rate reaches $B_m/2$ and thereafter it operates in congestion avoidance. Let d_1 and d_2 be the durations of the slow-start and congestion avoidance phases respectively and let A_1 and A_2 be the number of packets transmitted during these phases. Following the approach used for mode 1, but using B_m in place of B_{max} and assuming that $B_m = 2^H B_m in \;\; (H \in I)$, we can derive the expressions for d_1 and A_1 as

$$d_1 = R\log_2\left(\frac{B_m}{B_{min}}\right) \tag{21}$$

and

$$A_1 = R\,(B_m - B_{min}) \tag{22}$$

For the congestion avoidance phase, we again follow the approach in mode 1 to obtain

$$d_2 = RB_m/2 \tag{23}$$

and

$$A_2 = RB_m\left(3B_m/8 + 1/4\right) \tag{24}$$

B_m is not known in this case. However we can obtain B_m in terms on known quantities by noting that $A_1 + A_2 = 1/\lambda$, hence from (22) and (24),

$$1/\lambda = R[3B_m^2/8 + 5B_m/4 - B_{min}]$$

This leads to,

$$3B_m^2 + 10B_m - 8\left(B_{min} + 1/(\lambda R)\right) = 0$$

Hence,

$$B_m = \frac{-5 + \sqrt{25 + 24\ \left(B_{min} + 1/(\lambda R)\right)}}{3}$$

Since $B = 1/(\lambda\left(d_1 + d_2\right)$, from (21) and (23) we have,

$$B = \frac{1}{\lambda R\left(\log_2 B_m - \log_2 B_{min} + B_m/2\right)}$$

or

$$B = \frac{6}{\lambda R\left[\left(-5 + \sqrt{25 + 24\left(B_{min} + 1/(\lambda R)\right)}\right) + 2\log_2\left(-5 + \sqrt{25 + 24\left(B_{min} + 1/(\lambda R)\right)}\right) - 2\log_2 B_{min}\right]}$$

Let λ_s denote the minimum CS probability such that the source operates in mode 2. Therefore the source will reach the saturation point (mode 1) only when $\lambda < \lambda_s$. In order to do so, the NAK cycle must be long enough for the source to transmit at least than $R(B_{max} - B_{min})$ packets in the slow-start phase and $RB_{max}\left(3RB_{max}/8 + 1/4\right)$ packets in the congestion avoidance phase of a NAK cycle. Since the number of packets in a NAK cycle is $1/\lambda$, the source will reach the saturation point only if

$$1/\lambda > R(B_{max} - B_{min}) + RB_{max}\left(3RB_{max}/8 + 1/4\right)$$

Hence,

$$\lambda_s = \frac{1}{R\left(3B_{max}^2/8 + 5B_{max}/4 - B_{min}\right)}$$