Design and Analysis of Loss Indication Filters for Multicast Congestion Control*

Supratik Bhattacharyya, Don Towsley, Jim Kurose
Department of Computer Science
University of Massachusetts Amherst
Amherst MA, USA
{bhattach,towsley,kurose}@cs.umass.edu

CMPSCI Technical Report TR 99-46

February 2, 2000

Abstract

A key issue in the design of source-based multicast congestion control schemes is how to aggregate loss indications from multiple receivers into a single rate control decision at the source. Such aggregation entails filtering out a portion of the loss indications received by the source, and then using the remaining for rate adjustments. In this paper, we first propose a set of goals guiding the design of loss indication filters. We then present a novel loss indication filtering approach, the Linear Proportional Response (LPR) approach. Analysis and simulation is used to compare LPR to two well-known approaches – the Random Listening Algorithm (RLA) ([1]), and the Worst Estimate-Based Tracking (WET) [2] approach. Our results indicate that LPR achieves a desirable tradeoff between stability and response, thereby making it more suitable than WET and RLA for deployment in an Internet-like environment.

*This work was supported by the National Science Foundation under grants NCR 9508274 and CDA-9502639, and by DARPA under agreement No.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.

1 Introduction

Congestion control has emerged as one of the most important challenges in the widespread deployment of multicasting technology in wide-area networks. Multicast traffic is expected to constitute a significant portion of wide-area network traffic in the near future; hence it is important to control the transmission rate of multicast connections in order to prevent congestion collapse and ensure "fair" bandwidth sharing among competing unicast and multicast connections. However, current IP-based networks such as the Internet offer extremely limited support at the network layer for congestion control. Research in multicast congestion control has thus focussed primarily on transport layer solutions, where a multicast source control its transmission rate based on end-to-end measurements of network congestion. The transmission rate is decreased whenever a source detects congestion (usually through packet loss or delay indications), and is increased gradually in the absence of congestion.

A unique challenge in the design of loss indication-based multicast congestion control algorithms is the problem of combining loss indications from multiple receivers into a single rate control decision at the source. Numerous proposals for source-based multicast congestion control have appeared in recent years ([3, 4, 5, 6, 7, 8, 9, 1]) representing a wide range of possibilities for how the combination or aggregation of loss indications may be accomplished. However, a common underlying requirement of all these approaches is that of a **loss indication filter (LIF)** that filters out a portion of the loss indications received from various receivers. The remaining loss indications are then supplied as input to a rate adjustment algorithm to adjust the transmission rate.

Loss indication filtering is needed for two reasons. First, if a multicast session reduces its rate in response to every loss indication that it receives, then its transmission rate will be completely throttled [2]; hence it is necessary to filter out some of the loss indications received. Secondly, allocation of bandwidth to a multicast session may need to be done according to congestion conditions on only certain end-to-end paths in a multicast tree. In the context of a loss indication-based congestion control algorithm, this implies that the transmission rate should be adjusted only when loss indications from one of these paths are received; in such cases, loss indications from all the other paths must be filtered out.

In this paper, we first propose a set of multicast bandwidth allocation goals that should guide the design of loss indication filters. These goals are based on the notion that the bandwidth allocated to a multicast session should be determined by the most congested end-to-end network path in the multicast tree. ([2, 1]). Briefly, the desired behavior is are as follows. When all paths in a multicast tree have comparable levels of congestion, the throughput of a multicast session should be commensurate, on an average, with the congestion level on any one of these paths. If however, one of the paths is much more congested than any of the others, the multicast session's throughput should be determined solely according to this path.

We then present a novel approach towards LIF design - the **Linear Proportional Response (LPR)** approach. In this approach, a multicast source, on receiving a loss indication from any receiver, passes it through the filter with a probability that is proportional to the loss probability at the receiver. We show that the LPR approach meets all the goals that we specify for LIF design.

We present a comparative study of the LPR approach with two previously proposed filtering approaches. The first of these is the LIF approach proposed as part of the Random Listening Algorithm (henceforth referred to as RLA)([1]). The RLA approach also filters out loss indications from a receiver probabilistically; however, the probability is based solely on the number of receivers, not on loss probability measurements. The second approach, motivated by our earlier work in [2], filters out loss indications from every receiver except the most congested one in the multicast group. The most congested or "worst" receiver is identified as the one with the highest reported value of loss probability. Loss indications from this receiver are then used for rate adjustments. we refer to this approach as the Worst Estimate-based Tracking (WET) approach.

In this present work, we derive analytic expressions for the bandwidth allocated to a multicast session under each of the three LIF approaches. We then derive expressions for an upper bound on the excess bandwidth allocated to a multicast connection under LPR and RLA. This upper bound represents the maximum extent to which each scheme can deviate from our fairness goal; we find that the deviation under RLA is much greater than LPR. We also find that WET is more somewhat fairer than LPR in the steady-state, but LPR exhibits much better response to transient changes in network conditions. Thus LPR achieves a better balance between stability and transient response than the other two approaches, making it more suitable for an Internet-like environment.

The rest of the paper is organized as follows. In Section 2 provides an architectural view of loss-indication based multicast congestion control algorithms, of which loss indication filters is an essential component. It then presents a set of goals guiding the design of Loss Indication filters. Section 3 describes a family of rate adjustment adjustment algorithms that are used in this paper to compare different LIF approaches. In Section 4 we describe two well-known LIFS – WET and RLA , and then propose a novel filtering approach, Linear Proportional Response LPR. Section 5 provides an analytical comparison of LPR and RLA. Section 6 presents a simulation study of the steady-state and transient performance of LPR, RLA and WET. Section 7 discusses the suitability of LPR for multicast congestion control in the Internet. Section 8 concludes the paper with discussions on some future research directions.

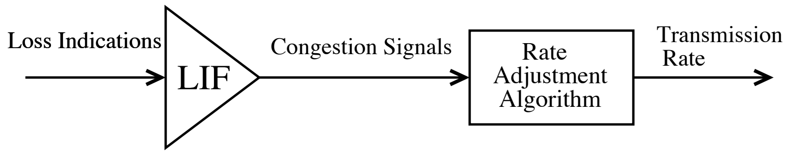

Figure 1: High-level view of loss indication-based congestion control algorithms

2 LIF Design Objectives

Figure 1 is a high-level view of loss-indication based multicast congestion control algorithms which shows two main components : a loss indication filter (LIF), and a rate adjustment algorithm. Loss indications received from multicast receivers are first filtered by passing them through the LIF. Let us refer to a loss indication that passes through the LIF as a **congestion signal** (CS). Congestion signals are provided as input to a rate adjustment algorithm. Most rate adjustment algorithms reduce a source's transmission rate multiplicatively on receiving a CS, and increase the rate additively in the absence of CSs. Such additive increase multiplicative decrease (AIMD) algorithms have been shown to possess certain desirable fairness and convergence properties [10]. We observe that the high-level design in Figure 1 is equally applicable to loss-indication based congestion control for unicast sources. Only, in that case, the LIF simply passes all the loss indications that it receives from the unicast receiver.

The design of the LIF in the case of a multicast session is largely a policy issue. For example, an LIF may filter out LIs from non-representatives, as in the case of a representative-based scheme([6]) . 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 [5]. RLA ([1]) filters out LIs with the goal of responding, on an average, to one out N LIs from any of its N receivers.

The primary consideration in this present work towards LIF design is how much bandwidth a multicast connection is allocated vis-a-vis competing unicast sessions. Though the issue of fairness is bandwidth sharing among multicast and unicast session is still open ([11, 3, 12, 2, 4, 13]), the notion "worst path" fairness is gaining acceptance among the research community as a well-understood fairness goal for the near future [2, 1, 4, 12, 3]. Under worst-path fairness, a multicast session is always allocated bandwidth based on the most congested source-to-destination path in its multicast tree. On that path, the available bandwidth is divided equally among this session and every unicast session that traverses this network path. The most congested path has to be determine on a time-scale that is of the order of tens or (even hundreds) of seconds, depending on network traffic characteristics. This is the time-scale on which congestion control algorithms would enable a source to gradually adapt its transmission rate in response to changing network congestion levels. This is also the time-scale on which notions such as average transmission rate and fairness are meaningful [14].

An obvious way of realizing worst-path fairness is to have a multicast source adjust its rate in response to to loss indications from only the "worst" receiver in a multicast group and ignore all other loss indications. This corresponds to an LIF that filters out loss indications from all but the worst receiver [2]. The worst receiver is determined as the one with the highest loss probability estimate (determined on an appropriate timescale). We refer to this LIF approach as the **Worst Estimate-based Tracking** (WET).

Although WET can ideally realize worst-path fairness, we shall show in later sections that it exhibits poor responsiveness to changes in network conditions. Hence, in order to examine a desirable tradeoff between fairness and responsiveness, we have adopted a more pragmatic goal for bandwidth sharing. We require an LIF to realize worst-path fairness only under certain conditions, while limiting the extent of unfairness under all other conditions. At the same time, an LIF should be able to react quickly to changing conditions such as sudden onset of congestion, link failures and dynamically changing multicast group membership.

We now present a set of bandwidth sharing goals guiding the design of the loss indication filters. But first let us introduce some terminology that will be used throughout this paper. Let us consider a multicast session M, with N receivers numbered 1 through N. Assuming that packet losses are temporally uncorrelated, let p_i be the packet loss probability on the end-to-end path from the source to receiver i. Without loss of generality, let us also assume that $p_1 \leq p_2 \leq \cdots \leq p_N$. Assume that the same AIMD algorithm is used by all unicast sessions to control their rates. It has been established that for such AIMD algorithms, the average throughput of a session, defined as the average number of packets transmitted per unit time, is a decreasing function of the end-to-end loss probability [15]. Let $B(p_i)$ be the average throughput of a unicast session that traverses the end-to-end path from the multicast source to receiver i. Let B_M be the average throughput of session M.

Then the design of LIF for M should satisfy the following conditions :

- Condition (1) : If $N = 1$, then $B_M = B(p_1)$.
- Condition (2) : If $p_i = p,\ \forall i = 1, 2, \cdots, N$, then $B_M = B(p)$.
- Condition (3) : If $p_i/p_N \to 0,\ i = 1, 2, \cdots, N-1$, then $B_M/B(p_N) \to 1$.

Condition (1) mandates that the average throughput of a multicast session with a single receiver should be no different than that of a unicast session experiencing the same end-to-end loss probability. Condition (2) states that, when all receivers in a multicast group experiences identical packet loss probabilities, the average throughput of a multicast session should be the same as that of a unicast session traversing any of the end-to-end paths in the multicast tree. Condition (3) states that if one receiver in a multicast session is far more congested than every other receiver, then the average throughput of a multicast session should be close to that of a unicast session that traverses the end-to-end path leading to this heavily congested receiver.

It is clear from Figure 1 that the throughput of a source is determined by a combination of the LIF and the rate adjustment algorithm in use. Therefore, any comparison of LIF performance with respect to bandwidth allocation must be done in the context of the same rate adjustment algorithm. In the next section, we describe a family of rate adjustment algorithms that we consider throughout this paper to provide a common basis for the comparison of different LIFs.

3 A Family of Rate Adjustment Algorithms

The family of rate adjustment algorithms that we consider belongs to the class of AIMD algorithms. For every algorithm in this family, 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 :

On receiving a CS, : $r \leftarrow r - r/C$,
In the absence of any CS for time S ; $r \leftarrow r + 1$.

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 algorithm in this class is completely defined by specifying the values of C and S. We note here that these algorithms are not specific to multicast or unicast connections, and can be used by both types of sessions. The only difference lies in the LIF used; a unicast session would consider every LI received to be a CS, whereas a multicast session would filter out some of LIs and use the LIs that pass through the filter as CSs.

For a *unicast* session experiencing an end-to-end packet loss probability of p, the average throughput of the algorithm is

$$B(p) = \sqrt{C}/\sqrt{pS} \tag{1}$$

See the appendix for details.

Moreover, the "worst path fair" average throughput for multicast session M is given by $B_M^{ideal} = B(p_N)$, i.e.

$$B_M^{ideal} = \sqrt{C}/\sqrt{p_N\, S} \tag{2}$$

In the rest of this paper, we assume that every source (both unicast and multicast) uses a rate adjustment algorithm from this family of algorithms. This allows to focus solely on the design of loss indication filters.

4 Three Loss Indication Filters

In this section, we first describe two previously proposed approaches towards filter design, and then present a novel approach - Linear Proportional Response (LPR). For each, we determine the the average throughput of a multicast session as a function of the receiver loss probabilities.

4.1 Random Listening Algorithm (RLA) Filter

In the Random Listening Algorithm proposed in [1], a multicast source, on receiving a loss indication from any one of its receivers, reduces its rate with probability $1/N$, where N is the number of receivers in the multicast group. This corresponds to having an LIF which allows a received loss indication to pass through with probability $\alpha = 1/N$. We refer to such a filter as the Random Listening Algorithm (RLA) filter. The RLA algorithm proposes a window-based rate-adjustment algorithm for use in conjunction with the RLA filter. However, our present interest is in LIF design, hence we focus on the filter component of the algorithm only.

For a multicast source that uses an RLA filter in conjunction with one of the rate adjustment algorithms described in Section 3, the average throughput is given by

$$B_M^{RLA} = [CN/(S * \sum_{i=1}^{N} p_i)]^{1/2} \tag{3}$$

The derivation of this result is provided in the appendix.

From equation (3), it is straightforward to see that the RLA filter satisfies conditions (1) and (2) for LIF design. However, a drawback is that RLA does not satisfy condition (3). In fact, as $p_i/p_N \to 0,\ i = 1, 2, \cdots, N-1$, $B_M^{RLA} \to [(CN)/(Sp_N)]^{1/2}$. Hence $B_M^{RLA}/B_M^{ideal} \to \sqrt{N}$. This implies that when receiver is far more congested than the others, the average throughput increases with increasing N, resulting in increasingly "unfair" treatment of competing unicast sessions.

Strictly speaking, [1] does not consider *all* of the receivers in a multicast group for the RLA algorithm. Instead it uses only "troubled" receivers, i.e. the ones that are experiencing a loss level higher than a predefined threshold. However, using such a threshold can only alleviate the "unfairness" problem described above; it does not fundamentally solve the problem. Moreover, the method for choosing this threshold in [1] does not necessarily restrict the set of troubled receivers to a small size; hence the problem of "unfair" treatment of unicast sessions for large values of N still exists.

4.2 Worst-Estimate Based Tracking (WET) Filter

The WET filter allows only loss indications from the most congested receiver in a multicast group to pass through, and ignores all others. As described in Section 2, this identification is made by a multicast source, based on loss probability reports received from its receivers.

For a multicast session M with N receivers, let receiver $i, ,\ i \in \{1, 2, \cdots, N\}$, estimate its end-to-end loss probability by counting the number of losses, X_i, over W consecutive packets. Then the loss probability reported by receiver i to the source is X_i/W. The WET filter then selects receiver w such that $w = \arg \max_{i, 1 \leq i \leq N} \{X_i\}$. Thereafter, every loss indication from receiver w is allowed through, while a loss indication from every other receiver is blocked. The worst receiver w is re-determined every time a source receives the next set of loss probability updates from its receivers.

Let π_i be the probability that the highest number of losses was reported by receiver i, i.e., $w = i$. Therefore,

$$\pi_i = \prod_{j=1, j \neq i}^{N} Prob[X_i \geq X_j] \tag{4}$$

The expected bandwidth allocation for a multicast session using a WET filter, B_{WET}, is thus

$$B_{WET} = \sum_{i=1}^{N} \pi_i * B(p_i) \tag{5}$$

where $B(p_i)$ is as defined earlier.

It can be easily shown that WET satisfies all three filter design requirements specified in Section 2. Also, as mentioned earlier in Section 2, "worst path" fairness in bandwidth sharing can be realized with the WET filter in an idealized setting. However, there are practical difficulties with WET which we shall illustrate in a later section.

4.3 Linear Proportional Response (LPR) filter

When the Linear Proportional Response filter receives a loss indication from receiver i, it allows the loss indication to pass through with probability α_i, where

$$\alpha_i = X_i / (\sum_{j=1}^{N} X_j), \qquad i = 1, 2, \cdots, N. \tag{6}$$

Thus the response to loss indications from a specific receiver is proportional to the loss probability estimate reported by that receiver. The rationale behind this approach is as follows. The LPR filter pays attention to loss indications from many receivers instead of one, and this makes it potentially more responsive to changes in network conditions, than WET. At the same time, LPR pays more attention to receivers reporting a higher level of congestion. We will see that makes it more responsive to the conditions on the "worst" path in the multicast tree than WET.

Strictly speaking, the RLA algorithm proposed in [1] *does* require receivers to send periodic loss probability reports to the source, based on which the source identifies "troubled" receivers. Hence LPR does not introduce the need for any additional congestion feedback. The difference between the two lies in that LPR makes better use of the information that is already available at the source.

The average throughput of a source using an LPR filter and one of the rate adjustment algorithms from Section (3) is :

$$B_{LPR} = [(C \sum_{i=1}^{N} X_i) / (S \sum_{i=1}^{N} X_i p_i)]^{1/2} \tag{7}$$

The derivation of this result is provided in the appendix.

As $W \to \infty$, $X_i / W \to p_i$, hence (7) reduces to

$$B_{LPR} = [(C \sum_{i=1}^{N} p_i) / (S \sum_{i=1}^{N} p_i^2)]^{1/2} \tag{8}$$

From (8), it is fairly straightforward to show that all three design goals for LIF design are fulfilled by LPR, when $W \to \infty$.

5 Analytic Comparison of LPR and RLA filters

Equations (7) and 3 show that under RLA and LPR, a multicast session may be usurp more than its "fair" share of throughput. While it may be acceptable to allow multicast sessions to be somewhat "unfair" to competing unicast traffic, it is important to ensure that this unfairness does not increase to the point of completely starving the unicast session. Hence a desirable goal is to bound the amount of throughput that a multicast session may get in excess of its fair share [1]. In this section, we find analytic upper bounds on the excess throughput under RLA and LPR, and show that in the case of LPR, the bound grows grows much more gradually as the number of receivers increases, than that in the case of RL. The expression for the upper bound for RLA has been derived in [1], albeit for a different rate adjustment algorithm. We find that our result matches exactly with the one in [1].

Let us define the **throughput ratio** r_f as the ratio of the average throughput of a multicast session using the LIF f to the ideal average throughput given by equation (2).

We now consider each of the two approaches :

- **Random Listening (RLA)** : From equations (2) and (3), the throughput ratio for RLA is obtained as

$$r_{RL} = [N/(1 + \sum_{i=1}^{N-1} p_i/p_N)]^{1/2} \quad (9)$$

  This is maximized when $p_i/p_N = 0, i = 1, 2, \cdots N$. This maximum value r_{RL} is

$$r_{RL}^{max} = \sqrt{N} \quad (10)$$

- **Linear Proportional Response (LPR)** : Here we consider the case where $X_i/W \rightarrow p_i$ as $W \rightarrow \infty$. From (2) and (8), we obtain

$$r_{LPR} = [(1 + \sum_{i=1}^{N-1} p_i/p_N)/(1 + \sum_{i=1}^{N-1} p_i^2/p_N^2)]^{1/2} \quad (11)$$

  Of course, the validity of the above expression for finite W depends on how fast $X_i/W \rightarrow p_i$. We will address this issue shortly. But let us first consider the upper bound on the value of r_{LPR} from equation (11). Maximizing r_{LPR} is equivalent to solving the following constrained minimization problem

  ($C1$) Minimize $[1 + \sum_{i=1}^{N-1} x_i^2] \ / \ [1 + \sum_{i=1}^{N-1} x_i]$

  s.t. $\quad 0 \leq x_i \leq 1, \ \ i \in \{1, 2, \cdots, N-1\}$

The solution to this optimization problem is $x_1 = x_2 = \cdots = x_{N-1} = 1/(\sqrt{N}+1)$. Details of this solution are provided in the appendix.

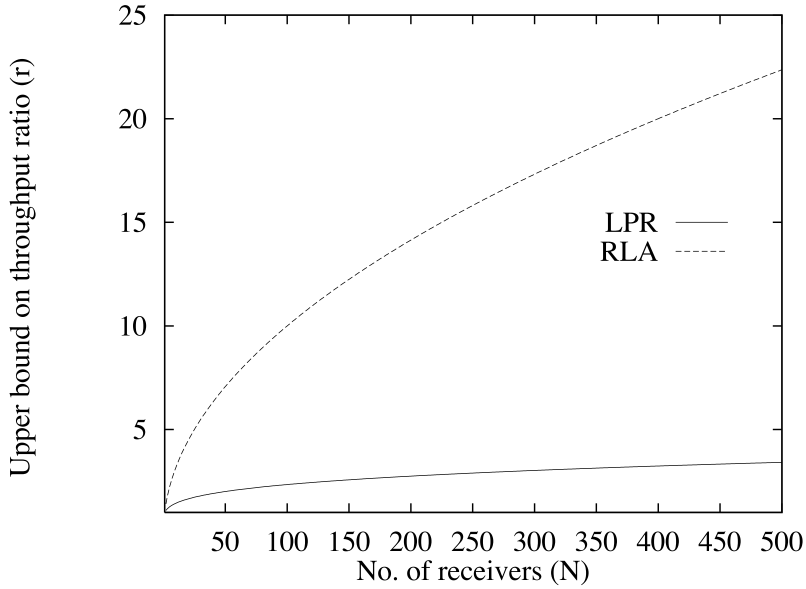

Figure 2: Upper bound on throughput ratio (r) for LPR and RLA

Substituting these values of $\{x_i\}$ in (11) yields

$$r_{LPR}^{max} = \sqrt{(\sqrt{N}+1)/2} \tag{12}$$

Figure 2 shows a plot of r_{LPR}^{max} and r_{RL}^{max} for values of N ranging from $N = 2$ to $N = 500$. We can clearly see that LPR yields a much tighter bound on the throughput ratio than RLA, and that the bound for LPR scales much better with increasing values of N. This implies that LPR is better at restricting the degree of "unfairness" of a multicast session towards competing unicast sessions, and is, in that sense, more "unicast-friendly".

These results on upper bounds do not imply any ordering on the average throughput under LPR and RLA for a given set of $\{p_i\}$. However, it is straightforward to show that as $W \to \infty$, the average throughputs under LPR is *always less* than that under RLA for any given set of p_is, $i = 1, 2, \cdots, N$. This follows from equations (3) and (8) and from the simple algebraic result that

$$N * \sum_{i=1}^{N} p_i^2 \geq \left(\sum_{i=1}^{N} p_i \right)^2 \tag{13}$$

5.1 Asymptotic behaviors of the RLA and LPR filters

Let us next consider the average throughputs under the RLA and LPR filtering approaches, when the number of receivers (N) is infinitely large. For this purpose, we assume that the loss probabilities, $\{p_i\}$, are random variables drawn from a distribution with mean μ and standard deviation σ.

- **RLA Filter** : From the Law of Large Numbers, it follows that $\lim_{N\to\infty} \left(\sum_{i=1}^{N} p_i\right)/N = \mu$. Hence from equation (3), we have

$$\lim_{N\to\infty} B_{RL}(p_1, p_2, \cdots, p_N) = [C/(S\mu)]^{1/2} \tag{14}$$

- **LPR Filter** : We again consider the case where $W \to \infty$. Since $E[p_i] = \mu$, and $Var(p_i) = \sigma^2$, it follows that $E[p_i^2] = \mu^2 + \sigma^2$. Since $\{p_i\}$ are i.i.d. variables, it follows that $\{p_i^2\}$, $i = 1, 2, \cdots, N$, are also i.i.d. random variables with mean $\mu^2+\sigma^2$. Then from the Law of Large Numbers it follows that $\lim_{N\to\infty} \left(\sum_{i=1}^{N} p_1^2\right)/N = \mu^2 + \sigma^2$. Hence it follows from equation (8) that

$$\lim_{N\to\infty} B_{LPR}(p_1, p_2, \cdots, p_N) = [(C\mu)/(S(\mu^2 + \sigma^2)]^{1/2} \tag{15}$$

It follows from equations (14) and (15) that

$$\lim_{N\to\infty} \frac{B_{RL}}{B_{LPR}} = \sqrt{1 + (\sigma^2/\mu^2)} \tag{16}$$

Thus when $N \to \infty$, the average multicast throughput with an LPR filter is always less than or equal to that with an RLA filter, the allocation being equal in the case that the variance of the probability distribution of the $\{p_i\}$ is zero.

5.2 Case Studies for finite W

Thus far, we have compared the performance of RLA and LPR for the case that $W \to \infty$. However, when W is finite, the average throughput of LPR is determined by equation (7) rather than equation (8). This means that the throughput depends on the random variables $\{X_i\}$. On the other hand, the throughput for RLA is still determined by equation (3), and is independent of $\{X_i\}$. We now present some case studies comparing the average throughput under LPR and RLA when W is finite.

The methodology for these case studies is as follows. For a given W and a given set of $\{p_i\}$s, $i \in 1, 2, \cdots, N$. a set of values of X_is are generated, each X_i being a random variable following the Binomial distribution with parameters p_i and W. The average throughput under LPR for this set of X_is is then computed using equation (7) with $C = 2.0$ and $S = 1.0$. The same experiment is repeated ten thousand times for each case study.

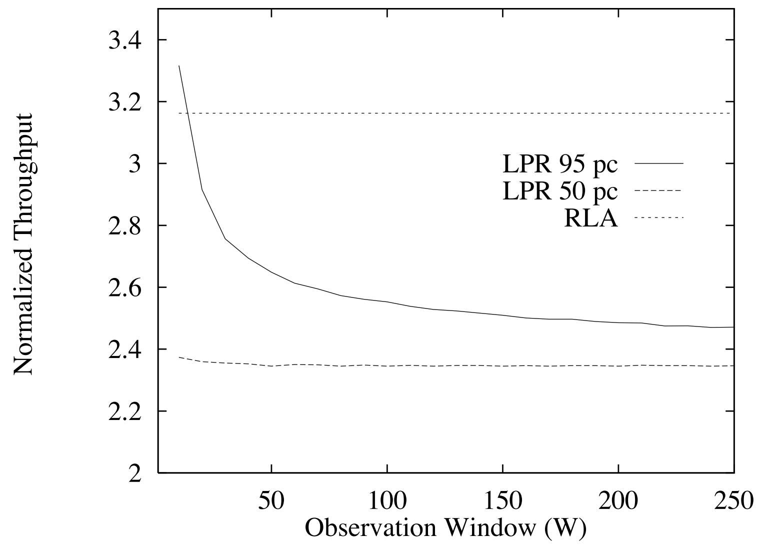

Figure 3: Case study for finite W and distribution A with $N = 100$.

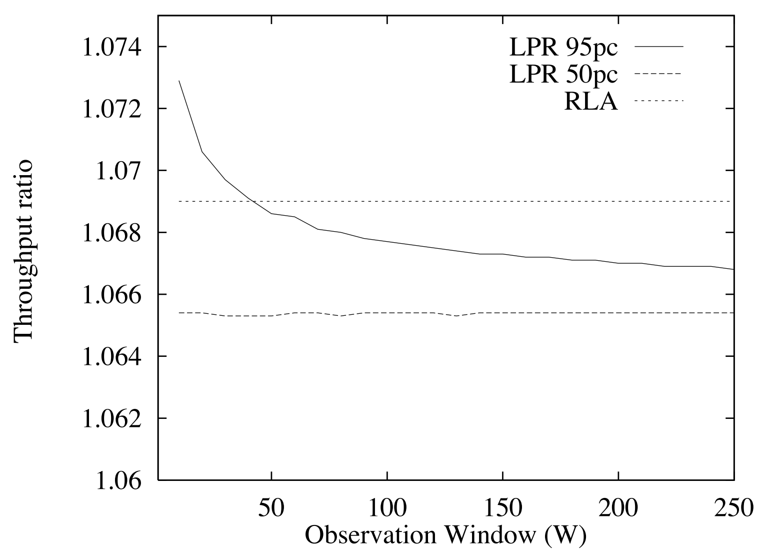

Figure 4: Case study for finite W and distribution B with $N = 100$.

Two distributions of the p_is were considered. For the first one, which we will refer to as **distribution** A, $p_N = 0.2$ and $p_1 = p_2 = \cdots = p_{N-1} = 1/(\sqrt{N}+1)$ (corresponds to r_{LPR}^{max} in equation (12)). For the second one, **distribution** B, $p_N = 0.1$ and $p_i = 0.75 * p_N + (i-1)/N * 0.25 * p_N$.

Figure (3) plots the throughput ratio against W for distribution A when $N = 100$. In this case, the "fair" multicast throughput value (from equation (2)) is 10.0 pkts/sec. The graphs show that under LPR, the median of the throughput ratio is much smaller than RLA. Even the 95th percentile value is smaller in general, except when W is very small.

Figure (4) plots the throughput ratio against W for distribution B when $N = 100$. The "fair" multicast throughput value (from equation (2)) is 14.14 pkts/sec. We see here that the $50$the percentile value of the throughput ratio under LPR is marginally smaller that RLA.The 95th percentile value for LPR is also quite close to RLA, for $W \geq 50$.

We have observed similar results for other values of N and other loss probability distributions. We infer from these case studies, that the throughput ratio under LPR is, on an average, much smaller than RL. The difference in the throughput ratio in the two cases increases with an increase in the variance of the distribution of receiver loss probabilities. This is in agreement with the result in (16). The case studies therefore provide a strong case for choosing LPR over RLA for realistic values of W, typically a a few hundred packets.

6 Simulations

Thus far, we have presented analytic results showing the advantages of LPR over RLA in terms of steady state behavior. We have also conjectured that the response time of WET to changes in network conditions is slower than LPR and RLA. In this section, we present simulation results illustrating the differences in steady-state behavior and transient response of the three filtering approaches. The simulations do not constitute a comprehensive study; instead, instead they are meant to provide an understanding of some of the tradeoffs of using the WET, LPR and RLA filters. We focus on a simple star topology, since it is sufficient for gaining a number of useful insights.

We focus on a simple star topology (Figure 5), since it is sufficient for gaining a number of useful insights. Every arm of the star corresponds to a single link with a finite buffer, that queues packets according to the FIFO discipline. All sessions, unicast and multicast, have their sources at the center of the star; every unicast session has its receiver at the end of one of the arms of the star, while every multicast session spans all the arms, with a receiver at the end of each arm. Every session (whether multicast or unicast) has an infinite data source, and uses a FLICA rate adjustment algorithm with $C = 2.0$ and $S = 25$ msecs. We assume that data packets are never reordered, although they may be lost due to buffer overflows at the links. A receiver sends per-packet loss indications to the source, as well as periodic loss probability reports. Lost packets are never retransmitted by the source, hence the loss indications are used solely for the purpose of rate adjustments. The reverse paths traversed by the LIs are different from the forward data path, and LIs are never lost or reordered. Each receiver monitors the conditions on its end-to-end path, and maintains a loss probability estimate p, that is updated as follows :

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

where $0 \leq g \leq 1$. Each receiver periodically reports its latest loss probability estimate to the source. The frequency of these reports varies with the simulation setting.

6.1 Steady State Behavior

The first two simulations compare the fairness of steady-state multicast throughput under LPR, RLA and WET. The performance metric used for this comparison is the throughput ratio, defined as the ratio of the *actual throughput* of a session to the worst-path fair throughput. The *actual* throughput, R, of a multicast session is calculated as follows. If the source transmits b packets in the interval $[t_1, t_2]$, then $R = b/(t_2 - t_1)$.

Simulation 1 examines the effect of the number of disjoint source-to-destination paths in a multicast tree on the performance of the filters. For a star topology this corresponds to studying the effect of varying the number of arms. Each arm of the star is configured to have a bandwidth of 200 pkts/sec and a buffer capacity of 10 packets. There are 5 unicast sessions on each arm of the star, and there are 5 multicast sessions spanning all the arms. One of the arms has 20 additional unicast sessions. The value of the g 17 is 0.005, and the loss probability reporting interval is 20 seconds. Since there are 40 sessions on the most congested arm of the star, the “worst path fair” throughput of each multicast session is $200/40$, or 5 packets/sec. The actual throughput of each multicast session is measured over 500 seconds, and then divided by 5 to obtain the throughout ratio.

Figure 6 shows the average throughput ratio of the ten multicast sessions under WET, LPR and RLA plotted against the number of number of arms of the star. We observe that the throughput ratio for WET is closest to the ideal value of 1, followed by LPR, whereas RLA is the furthest off. An intuitive understanding for the behavior RLA can be obtained from equation (9). Let p_i be the loss probability on arm i of the star, where there are N arms, and arm N is the one that is more congested than the others. Hence $p_1 = p_2 = \cdots p_{N-1} < p_N$. By increasing the number of arms of the star, we are effectively increasing the value of N in the numerator on the right hand side of equation (9). This

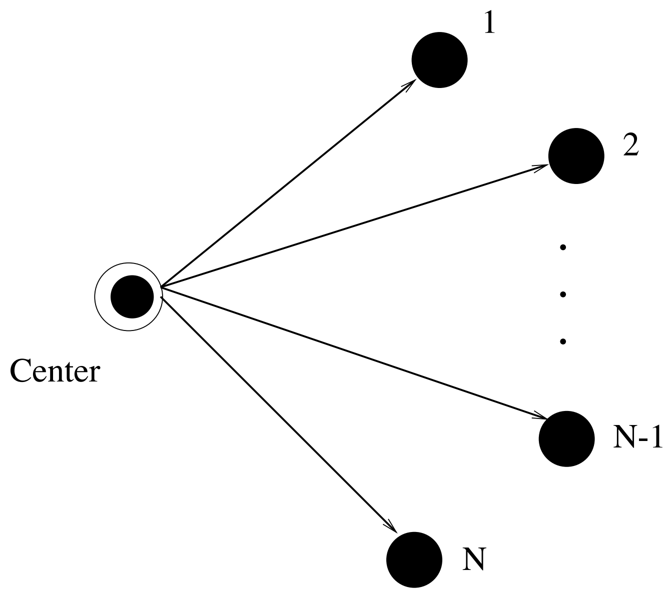

Figure 5: Star Topology

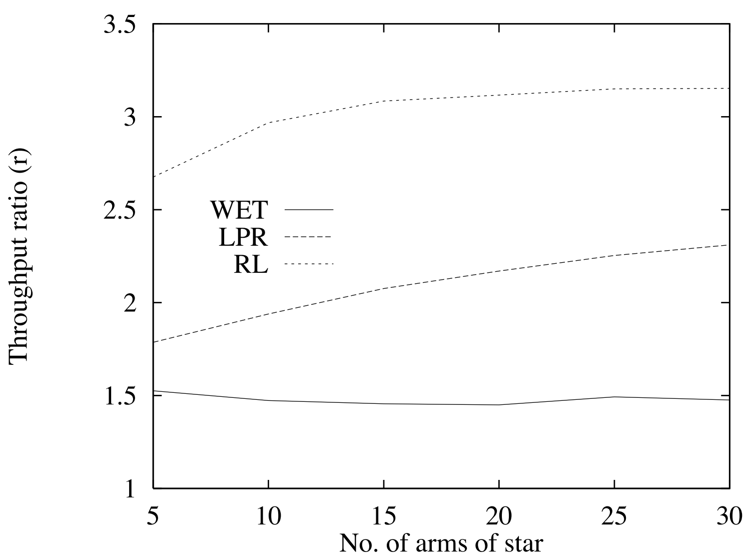

Figure 6: Simulation 1

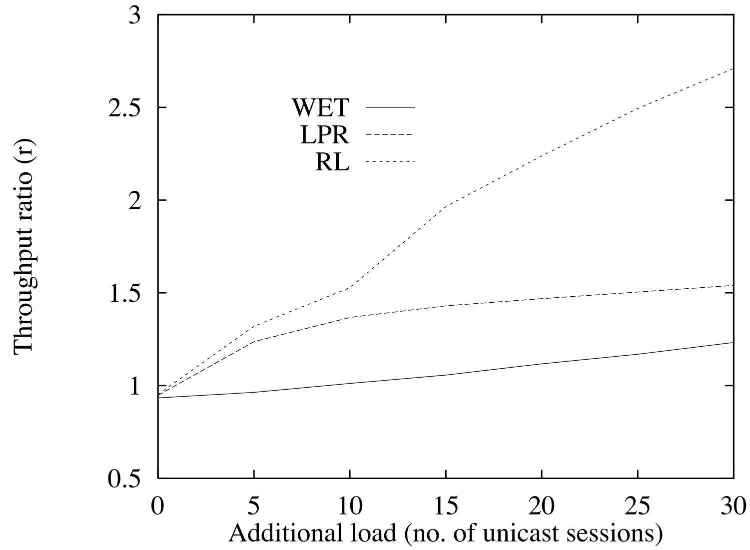

Figure 7: Simulation 2

causes a gradual increase in the value of r. However, every new arm adds a factor p_i/p_N to the denominator; for our simulation the value of p_i/p_N is sufficiently large to counter the growth of the numerator, resulting in only a gradual increase in r in Figure 6. However, if p_i/p_N is very small, then the increase in the value of r from RLA will be even sharper. A similar understanding of the behavior of LPR can be obtained from equation (11).

Simulation 2 examines the effect of one source-to-destination path in a multicast group being far more congested than any of the others. For this, the number of arms of the star is fixed at 10; but now the additional load on arm 1 is gradually increased by increasing the number of additional unicast sessions traversing it. We observe from Figure 7 that the throughput ratio under RLA grows rapidly as the load on arm 1 increases; however the throughput ratios under WET and LPR exhibit much slower growth. These findings agree with our earlier analytic results.

6.2 Transient Behavior

The next three simulations examine the responsiveness of the three LIFs to changes in network conditions. The first of these (simulation 3) simulates the case where a source suddenly stops receiving feedback from the receiver that is at the end of the most congested path in the multicast tree. This may happen if the receiver dies or leaves the multicast group, or if there is a breakdown on the network path leading to the receiver.

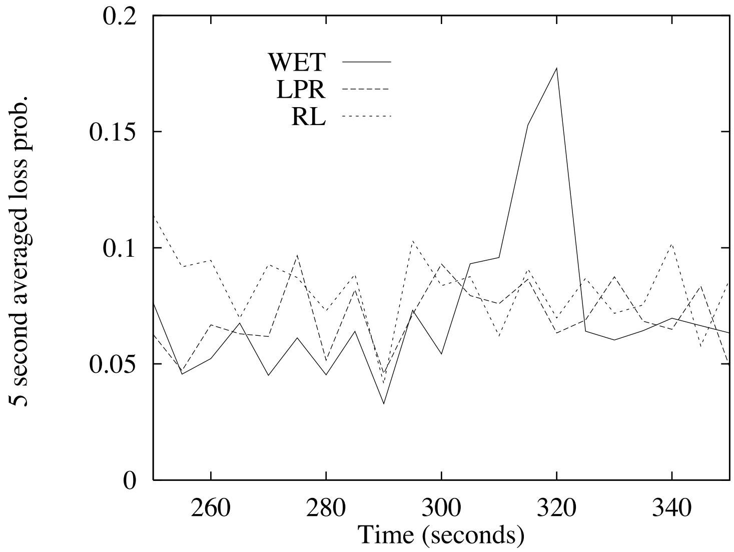

Figure 8: Simulation 3

A ten-arm star topology is considered, with each arm having a link bandwidth of 100 packets/sec and a link buffer of size 10 packets. Five unicast sessions traverse each of the arms, and there are five multicast sessions spanning all the arms. One of the arms has five additional unicast session, making it more congested than the rest. g (17) is set to 0.005, and the loss estimate reporting interval is 10 seconds The simulation is run for 500 seconds; after 300 seconds, two of the multicast sessions stop receiving feedback from their respective receivers at the end of the heavily loaded arm of the star. We examine the effect of this sudden change on on the loss probability at the queue of one of the less congested arms of the star. This loss probability is calculated as the ratio of the number of packets losses to the number of packet arrivals at the link queue over five second intervals.

Figure 8, shows a plot of the loss probability from $t = 250$ seconds to $t = 350$ seconds. We notice that in the case of WET, there is a sudden sharp increase in the loss probability immediately after $t = 300$ seconds, lasting for longer than ten seconds. The intuitive explanation for this is as follows. Before the sudden change, every multicast source was regulating its rate according to loss indications from its receiver at the end of the congested arm of the star. When two of the sessions stop hearing from their worst receivers, they start increasing their transmission rates, causing a sudden influx of

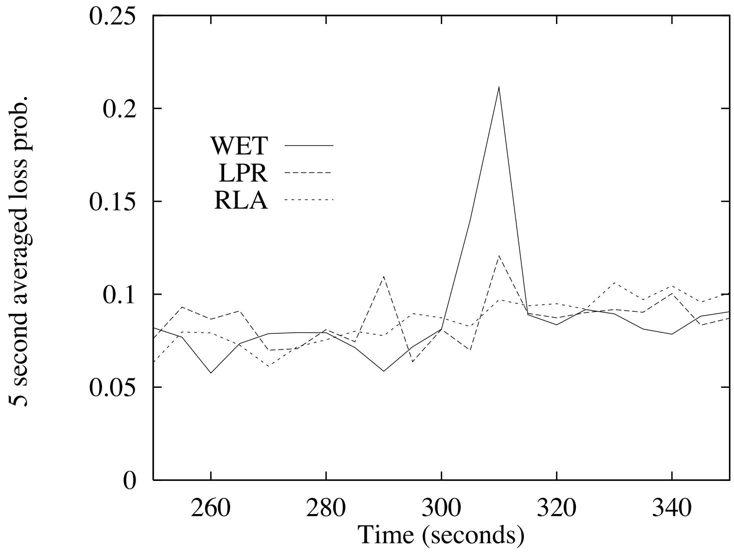

Figure 9: Simulation 4

traffic at the link queues of the other arms of the star. In our simulator, the situation is rectified when each of these sources receive the next round of periodic feedbacks from its receivers at $t = 310$ seconds, realize that they have "lost" the worst receiver, and pick one of its other receivers as the "worst". After that, the transient change in the loss probability gradually dies down.

There are several negative consequences of this sudden increase in loss probability. For example, rate-controlled unicast sessions on the links experiencing this sudden change will back off aggressively and it will take a while for them to be restored to steady state. Moreover such bursts of losses are extremely harmful for short-lived sessions, such as telnet. This phenomenon can potentially affect multiple links across wide sections of a network, and can occur frequently if multicast group membership is dynamic; hence it is desirable to prevent it from occurring. WET is incapable of preventing this sudden increase in loss probability, but LPR and RLA are able to, since both of them respond to loss indications from a number of receivers at any given time.

Simulation 4 considers a case where congestion suddenly clears up on the most congested path in a multicast tree. The simulation setting is a two-arm star, with with ten unicast sessions on each arm, and ten multicast sessions spanning all of the arms. There are 5 additional unicast sessions on one arm. Ten of the unicast sessions on the

Table 1: Simulation 5

	Packets lost/Packets sent	
Filter Type	Multicast	Unicast
LPR	297 / 690	231 / 482
WET	680 / 1237	248 / 459

heavily congested arm terminate after 300 seconds, suddenly clearing the congestion on that arm. All the other parameters are the same as in simulation 3.

We observe from Figure 9 see that under WET there is again a sudden transient increase in the link loss probability for the other arms of the start. This is because every multicast session was adjusting its rate according to loss indications from the receiver at the end of the heavily loaded arm before congestion on that arm cleared up. When the ten unicast sessions terminate, the link utilization goes down suddenly. Very few losses occur on that link while the multicast sessions continue to increase their rates to probe the link for the bandwidth that has been freed up. Eventually they overshoot the link capacity causing a sudden increase in loss probability; this in turn causes them to slow down their transmission rates; eventually steady state is re-established. Once the next round of periodic loss reports arrive at each source, the source discards its current worst receiver, in favor of one of the others. After that steady-state is re-established. We notice that LPR and RLA do not suffer from this transient fluctuation in loss probability since they respond to loss indications from multiple receivers at any given time.

Simulation 5 considers a two-arm star with thirteen unicast sessions on arm 1, ten unicast sessions on arm 2, and ten multicast sessions spanning both arms. Each arm of the star has a bandwidth of 300 packets/sec and a buffer size of 40 packets/sec. Receivers estimate their loss probability with $g = 0.0025$ (17), and report to the source once every twenty seconds. Between time $t = 201$ seconds and $t = 210$ seconds, there is a sudden surge of traffic on arm 2, created by a constant-rate source transmitting at a rate of 290 packets/sec. Since the duration of this sudden surge is shorter than the reporting interval for loss reports, it is futile for a source to react to this change via the periodic updates it receives; a source needs to react during the period of congestion.

Ideally all sessions should reduce their rate quickly at the onset of the surge, since they are likely to lose most of the packets that they transmit during this interval. This is particularly important in the case of reliable data transfer, where all lost packets must be retransmitted. Table 1 illustrates the effectiveness of WET and LPR in reacting to this sudden change. In both cases, the unicast sessions on arm 2 back off quickly thereby limiting the number of packets they send, and hence the number they lose. With LPR, the multicast sessions were already paying attention to the conditions on arm 2 before the surge began. As a result, they are able to cut back their rates, and limit the total number of transmissions to 690 packets, out of which 297 are lost. However, with WET, multicast sessions remain completely unaware of the the surge and keep transmitting at the rates that they were transmitting before the surge began. As a result they transmit a much larger number of packets, 1237, but also lose a proportionately higher number, 680. The number of transmissions and losses for the multicast sessions in this case are more than twice those for the unicast sessions on arm 2.

In summary, we reiterate that an appropriate choice of a loss indication filter requires a consideration of both steady-state and transient behaviors. WET achieves the best steady state behavior but has poor transient response. RLA shows good transient response but poor steady state behavior. Only LPR is able to achieve a balance between fairness and responsiveness.

7 Internet Deployment Considerations

In this section, we discuss the viability of combining LPR with an appropriate rate adjustment algorithm to provide congestion control for single-source multicast sessions in the Internet.

Given the current Internet architecture, there is an immediate need for an end-to-end solution for multicast congestion control. LPR does not assume any support from the network, hence it satisfies the end-to-end requirement. Moreover, it does not make any assumption about the spatial loss correlation (or lack of it) among receivers in a multicast group[1]. Fairness towards unicast sessions is another important requirement for a multicast congestion control protocol. Worst-path fairness is one possible fairness definition. A less stringent version of it – "essential" fairness – was proposed in [1] as a motivation for RLA. Essential fairness allows multicast sessions to usurp somewhat more bandwidth than their worst-path fair share. However, it imposes an upper bound on the amount of additional bandwidth obtained by a multicast session over and above its fair share. We have shown in Section 5 that LPR represents a significant improvement over RLA in this regard (Figure 2). In particular, when unicast sessions use the TCP protocol (as is expected in today's Internet), LPR can be combined with a window-based algorithm (such as the one in [1]) to effectively limit the "TCP-unfriendliness" of multicast sessions.

Multicast trees in the Internet are expected to span multiple networks, with hundreds, or even thousands, of receivers. Hence conditions are expected to be dynamic – receivers may join and leave groups, links may fail and there may be sudden changes in congestion levels in parts of a multicast tree. Although WET realizes worst-path fairness, it is

[1]There is no spatial correlation among multicast receivers for the star topology that we simulate. But the conclusions should hold for topologies where spatial loss correlation does exist.

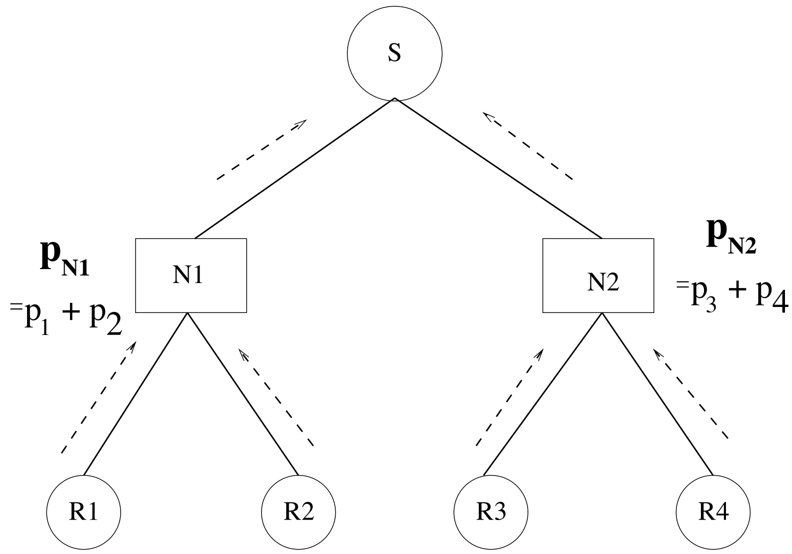

Figure 10: Hierarchical LPR example

incapable of responding to such changes in a timely manner. LPR, on the other hand, is a more practical approach that trades off some degree of steady-state fairness for higher responsiveness.

An important concern for any LIF approach is whether it requires a source to receive loss indications from *every* receiver in a multicast group, since that may not be feasible in practice. Though LPR assumes LIs from all receivers ideally, its performance will not degrade significantly if there is no feedback from some of the receivers. In particular, if there are a few very lossy receivers, then LPR can tolerate a lack of response from the other receivers, since most of the LIs from these receivers would be filtered out anyway.

Finally, LPR has been designed in a way that allows the filtering functionality to be distributed among multiple nodes in a multicast tree. We close this section with a brief example of how this may be accomplished.

Consider a simple tree topology with a source S, receivers R_1, R_2, R_3 and R_4, and two intermediate nodes $N1$ and $N2$ (Figure 10). Suppose that $S, N1$ and $N2$ are all capable of filtering out loss indications probabilistically. Let p_i be the loss probability corresponding to R_i $i = 1, \cdots, 4$. R_1 and R_2 report their loss estimates periodically to $N1$, while R_1 and R_2 report theirs to $N2$. Each of $R1$ and $R2$ calculate the sum of the loss estimates reported by its downstream receivers, and reports this value periodically to the source. Thus $N1$ reports the value $p_{N1} = p_1 + p_2$ to S, while $N2$ reports the value $p_{N2} = p_3 + p_4$.

Loss indication filtering is done in the following manner. When $N1$ receives a loss indication from receiver j ($j = 1, 2$), it filters out the LI with probability $[1 - p_j/(p_{N1})]$. Similarly, when $N1$ receives a loss indication from receiver j ($j = 3, 4$), it filters out the LI with probability $[1 - p_j/(p_{N2})]$. S filters out a loss indication from $N1$ (that was not filtered out by $N1$) with probability ([1 - $p_{N1}/(p_{N1} + p_{N2})$], and a loss indication from $N1$ (that was not filtered out by $N1$) with probability $[1 - (p_{N2}/(p_{N1} + p_{N2})]$. Note that the source now maintains aggregated loss estimates p_{N1} and p_{N2}, instead of per-receiver estimates. A careful inspection reveals that the aggregate effect of this hierarchical filtering is identical to having an LPR filter only at the source, with all four receivers reporting their loss estimates directly to the source. The two-level filtering approach described in this example can be generalized to a multi-level hierarchy of filters.

The above filtering approach, which we refer to as Hierarchical LPR (or HLPR) reduces the processing load at the source. It also reduces signalling overhead, since a large portion of loss indications can be filtered out long before they reach the source. Thus HLPR has the potential to scale well to large multicast groups that are likely to proliferate in the Internet.

8 Conclusions and Future Work

In this paper, we have focussed on the design of loss indication filters for source-based multicast congestion control. Our approach towards filter design is guided by a set of goals that specify the bandwidth allocation to a multicast session under different conditions, and are based on the notion of worst path multicast fairness [2]. We have presented a novel approach towards filter design - Linear Proportional Response - and have compared it with two previously proposed approaches, RLA and WET. Analysis and simulation reveal that among the three approaches, LPR achieves the best balance between steady-state behavior and responsiveness.

A number of research directions remain unexplored. Foremost, experimentation in a real network is needed to validate our findings, and to uncover implementation-specific issues. We need to compare LPR and WET in the context of a window-based rate control algorithm. The effect of heterogeneous round-trip times has to be investigated, as well as the effect of lost or irregular loss estimate reports. We also intend to pursue the design of hierarchical LPR, where LPR functionality is distributed among multiple nodes organized hierarchically in a multicast tree rooted at the source.

The LPR approach is one of many possible ways of filtering loss indications from multiple receivers. Several enhancements and modifications to this approach are possible. The primary advantage of LPR over WET is its faster response to changes in network conditions. However, we found that the multicast bandwidth allocation under WET is somewhat closer to our fairness goal than LPR. One way of improving the fairness of steady-state bandwidth allocation under LPR is to adjust the parameters of the multicast rate adjustment algorithm as follows. The fair throughput of a multicast session with rate control parameters C and S is given by

$$B_M^{ideal} = \sqrt{C}/\sqrt{p_N\, S} \tag{18}$$

whereas the throughput under LPR, for an infinitely large loss estimation window, is given by

$$B_{LPR} = [(C\sum_{i=1}^{N} p_i)/(S\sum_{i=1}^{N} p_i^2)]^{1/2} \tag{19}$$

We may replace the parameter C in equation (19) with $C^{'}$, such that B_{LPR} in (19) equals B_M^{ideal} in equation (18). This leads to

$$C/(p_N S) = (C^{'}\sum_{i=1}^{N} p_i)/(S\sum_{i=1}^{N} p_i^2)] \tag{20}$$

which on simplification, yields

$$C^{'} = C.\Delta \tag{21}$$

where

$$\Delta = [1 + \sum_{i=1}^{N-1} (p_i^2/p_N^2)]/[1 + \sum_{i=1}^{N-1} (p_i/p_N)] \tag{22}$$

This implies that if every multicast session uses an LPR filter with a rate adjustment parameter $C^{'}$, and if each unicast session uses the rate adjustment parameter C, we can expect the bandwidth allocations to be very close to our fairness goal. Of course, in practice, the loss probability values may not be available for computing Δ, but loss probability estimates can be used to obtain an approximate value. This remains to be verified through simulations.

The loss indication filtering probabilities $\alpha_i \;\; i = 1, 2 \cdots, N$, in LPR can be generalized as follows :

$$\alpha_i = {X_i}^h/(\sum_{j=1}^{N} {X_j}^h), \qquad i = 1, 2, \cdots, N. \tag{23}$$

where h is a non-negative constant. It is clear that LPR corresponds to the case where $h = 1$, whereas RLA corresponds to $h = 0$.

References

[1] H.A. Wang and M. Schwartz. Achieving Bounded Fairness for Multicast and TCP traffic in the Internet. In *Proceedings of ACM SIGCOMM*, 1998.

[2] S. Bhattacharyya, D. Towsley, and J. Kurose. The Loss Path Multiplicity Problem in Multicast Congestion Control. In *Proceedings of IEEE Infocom*, 1999.

[3] B. Whetten and J. Conlan. A Rate Based Congestion Control Scheme for Reliable Multicast . White Paper, GlobalCast Communications, Oct. 1998.

[4] M. Handley, S. Floyd, and B. Whetten. Strawman Specification for TCP Friendly (Reliable) Multicast Congestion Control . RMRG Meeting, Pisa, Italy, June 1999.

[5] T. Montgomery. A Loss Tolerant Rate Controller for Reliable Multicast. Technical Report NASA-IVV-97-011, West Virginia University, August 1997.

[6] D. DeLucia and K. Obraczka. A Multicast Congestion Control Mechanism for Reliable Multicast. Technical Report 97–685, University of Southern California, Computer Science Dept., August 1997.

[7] I. Rhee, N. Ballaguru, and G.N. Rouskas. MTCP : Scalable TCP-like Congestion Control for Reliable Multicast. Technical Report TR–98–01, North Carolina State University, Department of Computer Science, January 1998.

[8] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like Congestion Control for Layered Multicast Data Transfer. In *Proc. IEEE Infocom'98 Conf.*, 1998.

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

[10] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[11] D. Rubenstein, J. Kurose, and D. Towsley. The Impact of Multicast Layering on Network Fairness. In Proceedings of ACM SIGCOMM, 1999.

[12] H. Tzeng and K. Siu. On Max-Min Fair Congestion Control for Multicast ABR Service in ATM. *IEEE JSAC*, 15(3):545–556, April 1997.

[13] T. Jiang, M.H. Ammar, and E.W. Zegura. Inter-Receiver Fairness : A Novel Performance Measure for Multicast ABR Sessions. In *Proc. ACM SIGMETRICS'98 Conf.*, pages 202–211, 1998.

[14] S.J. Golestani and S. Bhattacharyya. A Class of End-to-End Congestion Control Algorithms for the Internet. In *Proc. ICNP*, 1998.

[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput : A Simple Model and its Empirical Verification. In *In Proc. ACM SIGCOMM'98 Conf.*, 1998.

[16] V. Misra, W. Gong, and D. Towsley. Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior. In *Proceedings of PERFORMANCE99, Istanbul, Turkey*, 1999.

[17] D. Luenberger. *Introduction to Linear and Non-Linear Programming*. Addison Wesley Publising Co., Reading, MA, 1965.

[18] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM'88 Conf.*, pages 158–173, 1988.

[19] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *Submitted to IEEE/ACM Transactions on Networking*, 1998.

APPENDIX

A Derivation of throughput using stochastic differential equations

Let there be N receivers in a multicast group numbered $1, 2, \cdots, N$. Assuming that packet losses are temporally uncorrelated, let p_i be the loss probability of a packet on the path leading to receiver i $(i = 1, 2, \cdots, N)$. Let $B(i)$ be the average bandwidth share of the multicast session when it adjusts its rate according to LIs from receiver i using one of the AIMD algorithms under consideration, and ignores all others.

We use stochastic differential equations to express $B(i)$ as a function of p_i [16]. Data traffic is modeled as a fluid, which enables us to consider the increase in the rate variable r (under the AIMD algorithms) to be continuous, instead of being in steps of 1 per S units of time. Hence we can represent the increase in the rate in time dt to be dt/S.

Let us model the arrival of LIs at the source from each receiver i as a Poisson process with rate λ_i, $i = 1, 2, \cdots, N$

Then we can represent the underlying loss process by a Poisson counter dP_{λ_i}, and express the evolution of the rate variable by the following differential equation :

$$dr = \frac{dt}{S} + (-r/C) dP_{\lambda_i} \tag{24}$$

To calculate the expected value of r, we can write the equation for the expected of r, $E[r]$), as

$$\frac{d}{dt} E[r] = E[1/S] - \frac{E[r]\,\lambda_i}{C}, \tag{25}$$

Solving the above equation for $E[r]$, we get

$$E[r](t) = \frac{C}{\lambda_i S} + D \ \exp(-\lambda_i t/C) \tag{26}$$

where D is a constant.

From this we can obtain the steady state solution ($t \to \infty$) as

$$E[r] = [C/(\lambda_i S)] \tag{27}$$

The approximate relation between λ_i and p_i is given by $p_i = \lambda_i / E[r]$. Hence from (27) it follows that $B(p_i) = E[r] = [C/(p_i S)]^{1/2}$.

A.1 Average throughput for an LPR filter

Let us refer to the LIs arriving at the LPR filter from the ith receiver as LI stream i. Every arrival on the ith stream is allowed to pass through with a probability $\alpha_i = X_i / \sum_{j=1}^{N} X_j$, where X_i is the number of reported to be lost by receiver i over a window of W packets. Let us refer to the filter output corresponding to receiver i as output stream i.

As before, let us model LIs from each receiver as a Poisson process. Let Y_i be the random variable corresponding to input stream i, having a Poisson distribution with rate λ_i. Let Z_i be the random variable associated with output stream i. Then it follows that Z_i has a Poisson distribution with rate $\alpha_i \lambda_i$. The stream of CSs that are input to the rate adjustment algorithm is a superposition of the N output streams. Therefore CSs also form a Poisson process with rate $\lambda = \sum_{i=1}^{N} \alpha_i * \lambda_i$. As before, we can write the differential equation for the rate evolution as :

$$dr = \frac{dt}{S} + (-r/C) dP_\lambda \tag{28}$$

Following the same procedure as before, we can derive the expression for the steady-state expected value of r as

$$\begin{aligned} E[r] &= [C/(\lambda S)]^{1/2} \\ &= [C/(S \sum_{i=1}^{N} \alpha_i \lambda_i)]^{1/2} \end{aligned}$$

Substituting each $\lambda_i = E[r] * p_i$ in the above equation leads to $B_{LPR} = E[r] = [C/(S \sum_{i=1}^{N} \alpha_i p_i)]^{1/2}$, from which equation (7) follows.

A.2 Average throughput for an RLA filter

With the same assumption about the loss process and with the same reasoning as in the LPR case, we can show that the stream of CSs forms a Poisson process with rate $\lambda = \frac{\sum_{i=1}^{N} \lambda_i}{N}$.

Using stochastic differential equations as before, we can derive the expression for the steady-state expected value of r to be

$$E[r] = [(CN)/(S\sum_{i=1}^{N} \lambda_i)]^{1/2} \tag{29}$$

Substituting each $\lambda_i = E[r] * p_i$ in the above equation leads to $B_{RL} = E[r] = [(CN)/(S \sum_{i=1}^{N} p_i)]^{1/2}$, from which equation (3) follows.

B Solution to minimization problem ($C1$) in Section 4

The problem that we want to solve is :

($C1$) Minimize $\quad f(x_1, x_2, \cdots, x_{N-1})$
s.t. $\quad 0 \leq x_i \leq 1, \quad i \in \{1,, 2 \cdots, N-1\}$

where $f(x_1, x_2, \cdots, x_{N-1}) = [1 + \sum_{i=1}^{N-1} x_i^2]/[1 + \sum_{i=1}^{N-1} x_i]$
It is straightforward to prove that the minimum value for $f(x_1, x_2, \cdots, x_{N-1})$ is obtained when $x_1 = x_2 = \cdots = x_{N-1}$. Hence the optimization problem reduces to

($C2$) Minimize $\quad g(x) = \frac{1+(N-1)x_i^2}{1+(N-1)x_i}$ s.t. $0 \leq x \leq 1$

Taking the first derivative of $g(x)$ with respect to x and setting it to zero leads to

$$(N-1)x^2 + 2x - 1 = 0 \tag{30}$$

which yields $x = 1/(\sqrt{N} + 1)$ as the non-negative solution.

We have also verified that the second derivative of $g(x)$ with respect to x is positive for all values of x. Hence the above solution corresponds to a minima for $g(x)$.

It therefore follows that the solution to problem $C1$ is $x_1 = x_2 = \cdots = x_{N-1} = 1/(\sqrt{N} + 1)$.