C. Livadas and I. Keidar, "The Case for Exploiting Packet Loss Locality in Multicast Loss Recovery," Technical Report MIT/LCS/TR-867, Lab. for Computer Science, MIT, Cambridge, MA, Oct. 2002.

The Case for Exploiting Packet Loss Locality in Multicast Loss Recovery

Carolos Livadas
Laboratory for Computer Science, MIT
clivadas@lcs.mit.edu

Idit Keidar
Dept. of Electrical Engineering, Technion
idish@ee.technion.ac.il

October 30, 2002

Abstract

This paper makes the case for exploiting packet loss locality in the loss recovery of reliable multicast protocols, such as SRM [4]. We claim that packet loss locality in IP multicast transmissions can be exploited by simple caching schemes. In such schemes, receivers cache information about the recovery of recently recovered packets and use this information to expedite the recovery of subsequent losses. We present a methodology for estimating the potential effectiveness of caching within multicast loss recovery. We use this methodology on the IP multicast transmission traces of Yajnik *et al.* [14]. We observe that IP multicast losses exhibit substantial locality and that caching can be very effective.

1 Introduction

Recently, numerous retransmission-based reliable multicast protocols have been proposed [4,6–8,11,12]. The challenge in designing such protocols lies in the requirements to scale to large multicast groups, to cater to a dynamic membership and network, and to minimize the recovery overhead. Most retransmission-based reliable multicast protocols treat losses independently and blindly repeat the recovery process for each loss. Such protocols can potentially reduce recovery latency and overhead by employing simple caching schemes that exploit packet loss locality. Locality is the property that losses suffered by a receiver at proximate times often occur on the same link of the IP multicast tree. We propose the extension of reliable multicast protocols with caching schemes in which receivers cache information about the recovery of recently recovered packets and use this information to expedite the recovery of subsequent losses.

We present a methodology for estimating the degree to which IP multicast losses exhibit locality and quantifying the potential effectiveness of caching in multicast loss recovery. Our methodology involves evaluating the performance of a caching-based loss location prediction scheme. In this scheme, each receiver caches the locations of its most recent losses whose locations it has identified and predicts that its next loss occurs at the location that appears most frequently in its cache. We consider a prediction to be a *hit* if it matches the location of the loss. The *hit rate* achieved by each receiver is an indication of the degree to which the losses suffered by each receiver exhibit locality. A *shared hit* corresponds to the case when the predictions of all receivers sharing a loss are hits; that is, all such receivers predict the same loss location and this loss location is correct. The *shared hit rate* can indicate the potential effectiveness of a caching scheme that relies on the collaboration and coordination of all receivers that share each loss.

We apply our evaluation methodology to the IP multicast transmission traces of Yajnik *et al.* [14]. In particular, we observe the hit rates achieved by our loss location prediction scheme as a function of: the cache size, the delay in detecting losses, the delay in identifying a loss's location, and the precision of the loss location identification. As the delays in detecting losses and in identifying their locations increase, caches become populated by the locations of less recent losses and predictions are made based on less recent information. Knowledge of the IP multicast tree topology may improve the precision with which the locations of losses are identified.

Our analysis reveals that the losses in the traces of Yajnik *et al.* exhibit substantial locality. The per-receiver hit rates achieved by our loss location prediction scheme in most cases exceed 40% and often exceed 80%. The shared hit rates range from 10% to 80% when the loss location identification is topology-oblivious and from 25% to 90% when it is topology-aware. The shared hit rates for a cache of size 10 exceed 35% (70%) for half the traces when the loss location identification is topology-oblivious (respectively, topology-aware). These observations suggest that

exploiting packet loss locality through caching within either existing or novel reliable multicast protocols has the potential of substantially reducing recovery latency and overhead.

Although the IP multicast transmission traces used in this paper are of modest duration and group size [14], we expect packet loss locality to also be prevalent in both longer-lived and larger group size IP multicast transmissions.

Recent studies of IP multicast transmission losses [1,5,14, 15] have investigated whether losses in the multicast setting exhibit *temporal* and *spatial correlation*. Temporal correlation refers to the degree to which losses are bursty and spatial correlation refers to degree to which losses are pairwise shared between receivers. All such studies observe that although packet losses are clearly not independent, they exhibit low temporal and spatial correlation. Our observations do not contradict these results. Loosely speaking, these studies examine whether the *loss* of *consecutive (or, close-by) packets* is correlated whereas we examine whether the *location* of *consecutive (or, close-by) losses* is correlated. Notably, packet loss locality can be exploited in multicast loss recovery.

This paper is organized as follows. Section 2 illustrates how caching can be incorporated within SRM in order to exploit locality. In Section 3, we present the IP multicast transmission trace data that we use in this paper and describe how we interpret and represent it. Section 4 presents our analysis of locality and the effectiveness of caching in multicast loss recovery. Section 5 concludes the paper and suggests future work directions.

2 Exploiting Locality Through Caching

In this section, we illustrate how caching can be used to exploit packet loss locality within the Scalable Reliable Multicast (SRM) protocol [4].

Packet recovery in SRM is initiated when a receiver detects a loss and schedules a retransmission *request* to be multicast in the near future. If the packet is received prior to the transmission of the scheduled request, then the scheduled request is canceled. If a request for the packet is received prior to the transmission of the scheduled request, then the scheduled request is postponed (suppressed and rescheduled). Upon receiving a request for a packet that has been received, a receiver schedules a retransmission of the requested packet (*reply*). If a reply for the same packet is received prior to the transmission of the scheduled reply, then the scheduled reply is canceled (suppressed). All requests and replies are multicast. SRM minimizes duplicate requests and replies using suppression. Unfortunately, suppression techniques delay the transmission of requests and replies so that only few (and, optimally, single) requests and replies are transmitted for each loss.

We suggest enhancing SRM with a caching-based expedited recovery scheme [9, 10]. This scheme operates roughly as follows. Each receiver caches the requestor and replier of the most recently recovered packet. A receiver considers itself to be optimal when its cached requestor is itself. Upon detecting losses, in addition to scheduling requests as is done in SRM, optimal receivers immediately unicast requests to their cached repliers. Upon receiving such a request, a receiver immediately multicasts a reply for the requested packet. A cache hit corresponds to the case when the unicast request is sent to a receiver that is capable of retransmitting the packet. Since unicast requests and the resulting retransmissions are not delayed for purposes of suppression, the recovery resulting from a hit incurs minimum latency. Moreover, it suppresses any requests and replies scheduled by SRM's recovery scheme. In the case of a miss, the recovery of a packet is carried out as prescribed by SRM's recovery scheme. The overhead associated with a miss is a single unicast request.

The above simple caching-based expedited recovery scheme associates loss locations with the requestor-replier pairs that recover the respective packets. This scheme may turn out to be too crude, in the sense that many requestor-replier pairs get associated with particular loss locations. To obtain more precise loss location identification, we propose employing a router-assisted scheme where routers annotate packets so that *turning point* routers [7, 11] are exposed. Turning points identify the subtrees of the IP multicast tree that are affected by each loss; thus, they identify loss locations precisely. This information can be used to associate sets of requestor-replier pairs to particular locations; thus, improving the effectiveness of caching.

SRM is highly resilient to group membership and network topology changes. Unfortunately, such resilience comes at the expense of performance. In static environments, other protocols [3, 6, 7, 11, 12] may outperform SRM by either *a priori* choosing designated repliers, arranging receivers in hierarchies, or extending the functionality of IP multicast routers so as to intelligently forward recovery packets. Our proposed caching-based expedited recovery scheme can substantially improve SRM's performance when the group membership and the network topology are static. Moreover, it may partially bridge the performance gap between SRM and hierarchical or router-assisted schemes,

while still retaining SRM's resilience to dynamic environments.

Of course, many variations on the above caching scheme may be considered: caching several of the most recent requestor-replier pairs and choosing to recover from the most frequent such pair, multicasting the expedited request, *etc.* Moreover, similar caching schemes may benefit either other existing or novel reliable multicast protocols in similar ways.

3 IP Multicast Traces and Their Representation

We represent IP multicast traces by per-receiver time series whose elements indicate the locations at which the losses suffered in the trace occur. We consider two such representations. The first representation is oblivious to the IP multicast tree topology and associates the location of each loss with the loss's loss pattern, *i.e.*, the set of receivers that share the loss. The second representation takes into consideration the IP multicast tree topology and estimates the link(s) that are responsible for each loss.

We begin this section by describing the IP multicast trace data that we use throughout the paper. We then describe how we interpret the trace data and produce our two trace representations.

3.1 Trace Data

We use 14 IP multicast transmission traces of Yajnik *et al.* [14]. These traces involve IP multicast transmissions each originating in the *World Radio Network* (WRN), the *UC Berkeley Multimedia Seminar* (UCB), or the *Radio Free Vat* (RFV). In these IP multicast transmissions, packets are transmitted at a constant rate. Each IP multicast transmission is received by a subset of 17 research community hosts spread out throughout the US and Europe. Each IP multicast transmission trace is comprised of per-receiver sequences indicating which packets were received and the order in which they were received. The traces do not include the packet reception times. Table 1 lists the source, date, number of receivers, IP multicast tree depth, packet transmission period, number of packets transmitted, and transmission duration for each of the 14 traces. Yajnik *et al.* also provide the IP multicast tree topology for each trace. For more information regarding the traces, see [14].

Yajnik *et al.* [14], as do the other multicast loss studies [1, 5, 15], represent IP multicast traces by per-receiver binary time series each of whose elements indicates whether the respective packet was lost by the respective receiver. For instance, element i of the binary time series for receiver j is equal to 1 if the receiver j did not receive the i-th packet of the IP multicast transmission. The *loss pattern* observed for packet i is the binary sequence whose j-th element is 1 if receiver j did not receive packet i.

Table 1 IP Multicast Traces of Yajnik *et al.* [14].

	Source & Date	# of Rcvrs	Tree Depth	Period *(msec)*	# of Pkts	Duration *(hr:min:sec)*
1	RFV960419	12	6	80	45001	1:00:00
2	RFV960508	10	5	40	148970	1:39:19
3	UCB960424	15	7	40	93734	1:02:29
4	WRN950919	8	4	80	17637	0:23:31
5	WRN951030	10	4	80	57030	1:16:02
6	WRN951101	9	5	80	41751	0:55:40
7	WRN951113	12	5	80	46443	1:01:55
8	WRN951114	10	4	80	38539	0:51:23
9	WRN951128	9	4	80	44956	0:59:56
10	WRN951204	11	5	80	45404	1:00:32
11	WRN951211	11	4	80	72519	1:36:42
12	WRN951214	7	4	80	38724	0:51:38
13	WRN951216	8	3	80	50202	1:06:56
14	WRN951218	8	3	80	69994	1:33:20

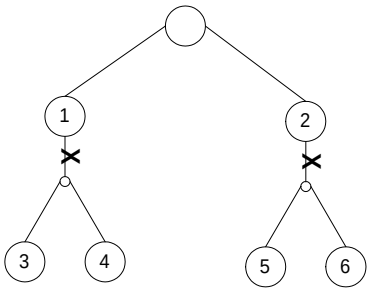
3.2 Virtual Link Trace Representation

Our first representation is oblivious to the IP multicast tree. We associate the location of each loss with the loss's loss pattern, *i.e.*, the set of receivers that share the loss. Although many of the observed loss patterns result from losses on multiple links of the IP multicast tree, we attribute each distinct loss pattern to a loss on a single *virtual link*. For example, a virtual link could represent the fact that receivers 2, 5, 8, and 12 did not receive a given packet.

By assigning a unique identifier to each distinct loss pattern, or virtual link, observed in the trace, we represent each trace by per-receiver time series whose elements are the identifiers of the virtual links responsible for the losses suffered by each receiver. We use the identifier 0 to denote that the particular packet was successfully received.

For the virtual link representation, a loss location prediction is a hit only if the receiver can predict the exact set of receivers that share the loss. However, in order to benefit from caching, a receiver need not predict this exact set. For instance, consider the lossy IP multicast transmission example shown in Figure 1 where a packet is lost on two links, leading to two independent subtrees of the IP multicast tree. Receivers 3 and 4 can recover the packet from receiver 1 and receivers 5 and 6 can recover the packet from receiver 2. Receivers in one subtree are not affected

Figure 1 Example of a Lossy IP Multicast Transmission.



by the fact that a loss also occurs on the other subtree. Were receivers 3 and 4 to predict that the loss is shared by receivers 3 and 4 only, they would be able to recover the packet from receiver 1. However, in the virtual link representation, this scenario is considered a cache miss. Thus, the performance analysis of our loss location prediction scheme using the virtual link representation may underestimate the expected effectiveness of caching in multicast loss recovery. In order to remedy this, in the next section we present a more precise representation which estimates the actual links on which losses occur.

3.3 Concrete Link Trace Representation

Our second representation involves per-receiver time series whose elements compute estimates of the actual links of the IP multicast tree responsible for the losses suffered by each receiver. We estimate the actual links responsible for each loss based on the IP multicast tree topology and the observed loss pattern in the trace for the respective packet. Each loss pattern observed in a trace may be the result of losses on either a single or a combination of actual links. Moreover, it may result from losses on several such combinations. For example, the loss pattern involving all receivers may result from either a single loss on the link leaving the source, or losses on each of the links leading to the receivers. We select a particular combination of links to represent each instance of a loss pattern based on the probability that a packet is dropped on exactly the links comprising each combination. We estimate this probability by first estimating the probability that a packet is dropped on each link of the IP multicast tree, *i.e.*, the link loss rates.

Let L be the set of links comprising the IP multicast tree of a given trace and $l_{nn'} \in L$ be the link that connects the nodes n and n', where n is the parent of n'. We define $p(l_{nn'})$ to be the probability that a packet is dropped along $l_{nn'}$ given that the packet is received by n. The probabilities $p(l_{nn'})$, for $l_{nn'} \in L$, can be estimated either by the method of Yajnik *et al.* [14] or the maximum-likelihood estimator method of Cáceres *et al.* [2]. For the traces used in this paper, both methods yield very similar link loss probability estimates. In this paper, we use the link loss probability estimates obtained using the method of Yajnik *et al.*

Given the IP multicast tree, it is straightforward to deduce the set of link combinations that result in any loss pattern observed in the trace. We assume that the probability of a packet being dropped on a link is independent of it being dropped on any other link. We compute the probability of occurrence of a particular link combination as the product of the probabilities of a packet being dropped on the links comprising the combination and successfully forwarded on the links leading to those comprising the combination.

More precisely, consider an observed loss pattern x. Let C_x be the set of all possible link combinations resulting in x, L_c be the set of links that comprise a combination $c \in C_x$, and U_c be the set of links that are neither in L_c nor downstream of any of the links in L_c. Presuming that the probabilities of loss along the links of the IP multicast tree are independent, the probability of occurrence of the link combination c is estimated by $p(c) = \prod_{l \in L_c} p(l) \cdot \prod_{l' \in U_c}(1 - p(l'))$. Thus, the relative probability that the observed loss pattern x results from the link combination c as opposed to the other combinations in C_x is given by $p_{C_x}(c) = p(c) / \sum_{c' \in C_x} p(c')$.

We select a particular link loss combination to represent an instance of the loss pattern x in the trace based on the relative probabilities of occurrence of all link loss combinations resulting in x. For 13 out of 14 of the traces we consider, more than 90% of the link combinations selected to represent the losses have relative probabilities of occurrence that exceed 95% and are often very close to 100%. For the remaining trace, 85% of the link combinations selected to represent the losses have relative probabilities of occurrence that exceed 98%. Thus, our estimates of the links responsible for the losses observed in each trace are predominantly accurate.

By assigning a unique identifier to each link of the IP multicast tree of each trace, we represent each trace by per-receiver time series whose elements are the identifiers of the links responsible for the losses suffered by each receiver. We use the identifier 0 to denote that the particular packet was successfully received.

While the performance analysis of our loss location prediction scheme using the virtual link trace representation may under-estimate the expected effectiveness of caching, the analysis using the concrete link trace representation may over-estimate it. Firstly, receivers may not always be able to deduce the exact locations at which losses occur. In SRM, for instance, receivers may identify a loss loca-

tion by the requestor-replier pair that recovers the loss, *i.e.*, the first receiver to request a retransmission and the first receiver to retransmit the packet. However, sometimes different requestor-replier pairs can emerge for different losses on the same link and sometimes the emerging requestor or replier is not optimal. Secondly, even an accurate identification of the link responsible for a loss at each receiver does not always yield optimal recovery. Consider the case where two receivers, 1 and 2, lose a given packet on separate links and there are two repliers, 3 and 4, that are equidistant from 1 and 2 and are both potential optimal repliers for both. Even when receivers 1 and 2 can accurately identify the links on which the packet was dropped, receiver 1 may request the packet from 3, and 2 may request it from 4, leading to two retransmissions. Although for the concrete link trace representation such predictions are considered hits, they do not lead to the desired recovery behavior involving a single request and a single reply. In contrast, in the case of the virtual link representation, such predictions are considered misses.

4 Evaluating the Effectiveness of Caching

In this section, we demonstrate that the IP multicast transmission traces of Yajnik *et al.* [14] exhibit substantial locality and that caching can be very effective. In particular, we analyze the performance of a caching-based loss location prediction scheme. In this scheme, each receiver caches the locations of its most recent losses whose locations it has identified and predicts that its next loss occurs at the location that appears most frequently in its cache. We refer to correct and incorrect per-receiver loss location predictions as *hits* and *misses*, respectively.

In the subsequent sections, we present and compare the hit rates achieved by our loss location prediction scheme for several cache sizes. A cache of size 1 predicts that the location of the next loss is that of the most recent loss whose location has been identified. An infinite cache records the location of all prior losses whose locations have been identified. Predictions made based on an infinite cache correspond to the most frequent loss location identified by the receiver up to that point in the trace.

We analyze the performance of our loss location prediction scheme using both virtual and concrete link trace representations. As noted above, the virtual link representation may under-estimate the expected effectiveness of caching in multicast loss recovery, while the concrete link representation may over-estimate it.

In Section 4.1, we assume that both the detection of losses and the identification of their location are immediate. In Section 4.2, we assume that losses are detected upon the receipt of later packets and their locations are identified immediately. In Section 4.3, we evaluate the performance of our loss location prediction scheme as the delay in identifying loss locations increases. In Section 4.4, we observe the degree to which all receivers that share a loss make the same predictions, under the assumption that loss detection is delayed and loss location identification is immediate. Prediction consistency would be required in cases when the loss recovery process requires the coordination of all receivers that share each loss. In Sections 4.1 through 4.3, we consider caches of size 1, 10, and infinity. In Section 4.5, we analyze the effect of the cache size on the shared hit rates.

In order to estimate the times at which receivers detect losses and identify their locations, we need to know the packet reception times. Since the trace data contains no timing information, we assume that all packets received by each receiver incur the same transmission latency; that is, we assume that packets are received at a constant rate.

4.1 Immediate Detection/Immediate Identification

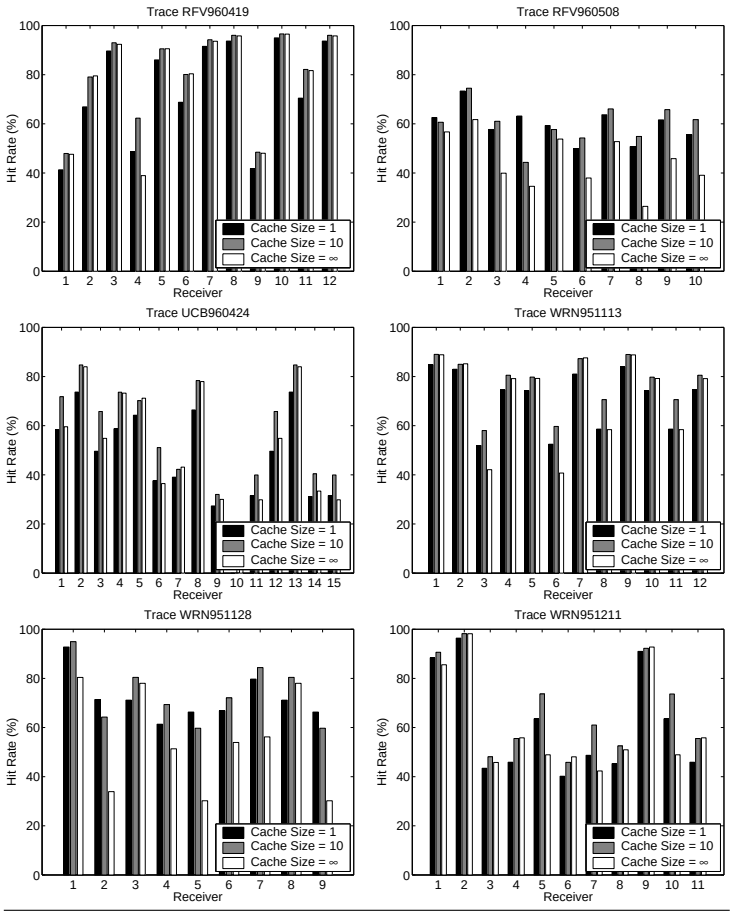
We present the hit rates achieved by our caching-based loss location prediction scheme, under the assumption that the detection of losses and the identification of their location are both immediate. That is, we assume that the loss location prediction scheme is aware of the location of all losses that precede the loss whose location is being predicted.

Figure 2 presents the per-receiver hit rates for the virtual link trace representation for 6 out of the 14 traces. The per-receiver hit rates for the rest of the traces are similar. Each of the graphs in Figure 2 plots the percentage of predictions that are correct, *i.e.*, the hit rate, for each of the receivers in the given trace.

We observe that the cache of size 10 outperforms the cache of size 1 in most cases. As observed by the multicast loss studies of [1, 5, 14, 15], IP multicast transmissions involve a few highly lossy links that generate a large percentage of the losses and a large number of slightly lossy links. With a larger cache, it is more likely that each prediction corresponds to a highly lossy link.

We also observe that caches of size 1 and 10 often outperform the infinite cache size. In fact, the infinite cache size performs as well as the others only for receivers whose losses are predominantly due to single locations. Consider, for instance, the hit rates achieved by receivers 2 and 3 of trace WRN951128. The caches of size 1 and

Figure 2 Virtual Link Trace Representation — Immediate Detection/Identification.



10 substantially outperform the infinite cache size for receiver 2. In the case of receiver 3, the hit rates achieved by caches of size 1 and 10 are comparable to those achieved by the infinite cache size. Figure 3 depicts the loss distributions for receivers 2 and 3 of trace WRN951128; that is, the percentage of losses suffered by each receiver that occur on each loss location. The loss percentages are shown in log scale. Three loss locations account for large percentages of the losses suffered by receiver 2. In this case, smaller cache sizes that can adapt quicker to changing loss conditions outperform the infinite cache. Conversely, the losses suffered by receiver 3 occur predominantly on a single location. In this case, the infinite cache size predicts that all losses occur at the highly lossy location and thus performs similarly to the smaller cache sizes.

Figure 4 presents the per-receiver hit rates for the concrete link trace representation for the same 6 traces. Again, the per-receiver hit rates for the rest of the traces are similar. The per-receiver hit rates for the concrete link trace representation are substantially higher than those for the virtual link trace representation. This is not surprising given the fact that in the case of the concrete link representation each receiver witnesses a small number of dis-

Figure 3 Virtual Link Trace Representation — Per-receiver Loss Distributions, Receivers 2 & 3, Trace WRN951128.

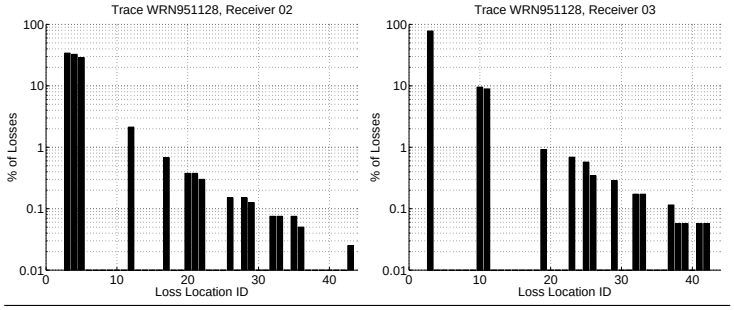
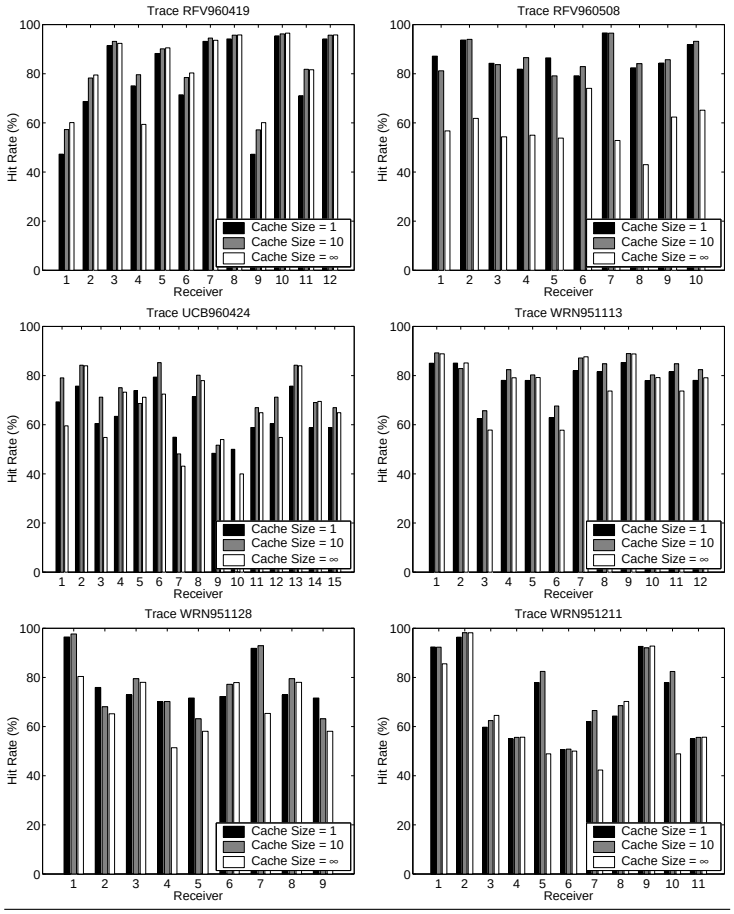


Figure 4 Concrete Link Trace Representation — Immediate Detection/Identification.



tinct losses — equal to the path length from the source to each receiver. Moreover, in the case of the concrete link trace representation, loss patterns resulting from simultaneous losses on highly lossy links are not misinterpreted as losses occurring at distinct locations; rather, each receiver attributes each loss to one of the IP multicast tree links that are on the path from the source to the particular receiver.

Figure 5 Virtual Link Trace Representation — Delayed Detection/Immediate Identification.

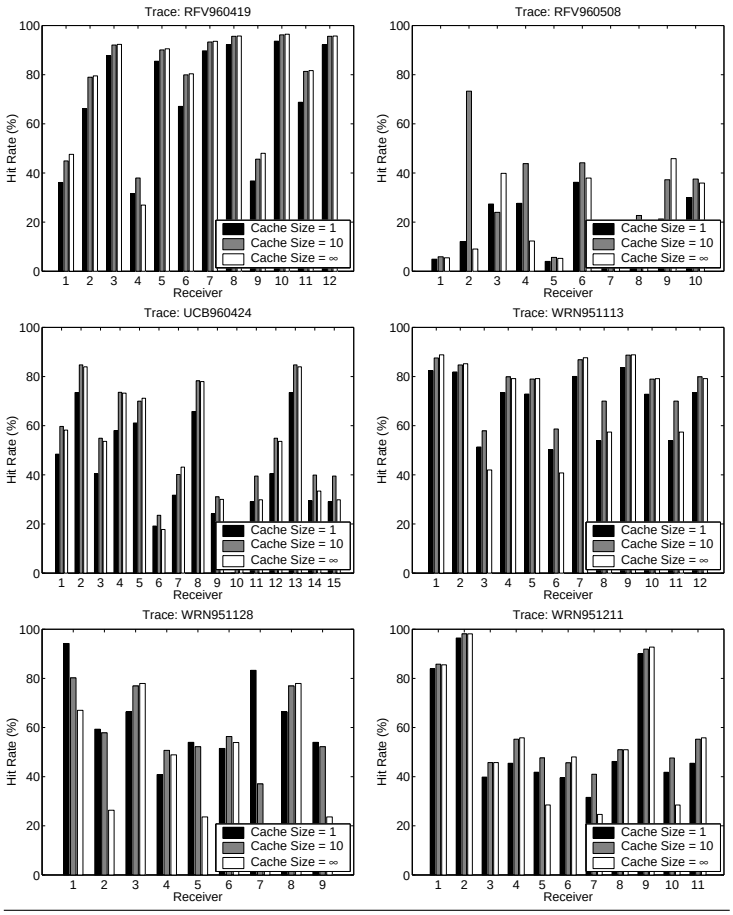
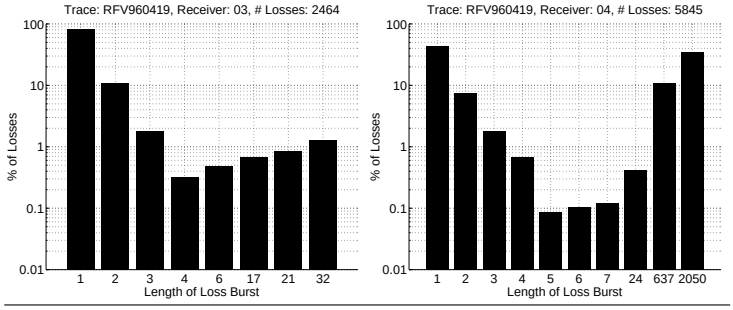


Figure 6 Loss Distribution wrt Burst Length, Receivers 3 & 4, Trace RFV960419.



4.2 Delayed Detection/Immediate Identification

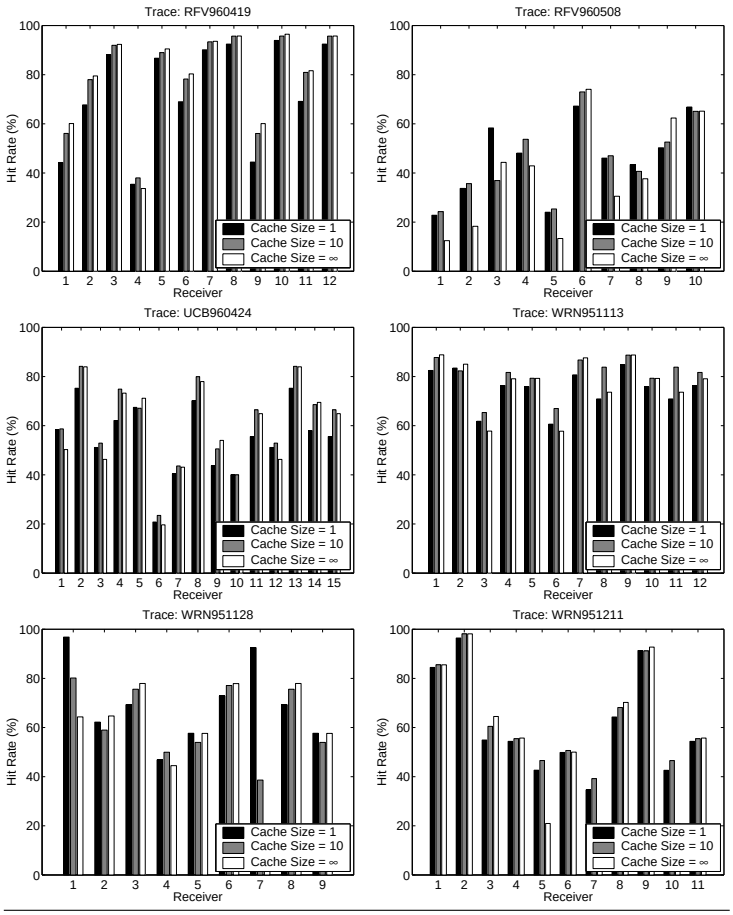
The packet loss locality exhibited in the previous section may not be exploitable, since losses may not be immediately detectable. Many reliable multicast protocols detect losses upon the receipt of later packets. Thus, in the case of loss bursts, losses are detected all at once upon the receipt of a packet following the loss burst. In this section, we observe the effect of delayed loss detection. In particular, we assume that: i) losses are detected upon the receipt of a later packet (delayed detection), and ii) the loss location prediction scheme is aware of the location of all losses that are detected earlier than the detection time of the loss whose location is being predicted (immediate loss location identification).

Figure 5 presents the per-receiver hit rates of our loss location prediction scheme for the virtual link trace representation of 6 out of the 14 traces. By comparing the hit rates presented in Figures 2 and 5, we observe that the delay in detecting losses heavily affects the hit rates of some traces; the trace RFV960508 is the most heavily affected trace and achieves the lowest hit rates of all 14 traces. This effect is due to loss bursts. With immediate detection, the prediction of the location of trailing losses within a burst is based on the location of the leading losses of the burst. In contrast, when losses are detected upon the receipt of a later packet, the losses comprising the burst are detected simultaneously and their locations are all predicted based on the locations of losses suffered prior to the burst. Thus, the (in)correct prediction of the losses comprising long loss bursts heavily affect the prediction hit rates.

Consider for instance the hit rates of receivers 3 and 4 of trace RFV960419. Figure 6 depicts the distribution of losses across loss bursts of increasing length for receivers 3 and 4 of trace RFV960419. More precisely, the graphs in Figure 6 plot the percentage of losses that comprise loss bursts of different lengths. The loss percentages are shown in log scale. Receiver 3 suffers predominantly isolated losses. Conversely, receiver 4 suffers a couple of long loss bursts. The adverse effect of these loss bursts on the hit rate of receiver 4 is evident when one compares receiver 4's hit rates in Figures 2 and 5; the hit rates of receiver 3 are barely affected by the delayed detection, while those of receiver 4 are nearly cut in half.

The adverse effect of the delay in detecting losses suggests that it would be beneficial to design schemes for detecting losses sooner. SRM's exchange of session messages is one such scheme. Session messages are used by receivers to periodically advertise the per-source transmission progress they have observed. Thus, receivers may discover losses by detecting discrepancies in the observed transmission progress of the receivers. When packets are transmitted at a fixed frequency, as is done in audio and video transmissions, an alternative approach may be to track the inter-packet delays and to declare a packet missing when its arrival with respect to its predecessor has exceeded some jitter threshold. In order for such schemes to allow the early detection and recovery of packets, session and recovery packets must avoid the congested links responsible for the loss burst, *e.g.*, using a source-based IP multicast tree implementation [13].

Figure 7 Concrete Link Trace Representation — Delayed Detection/Immediate Identification.



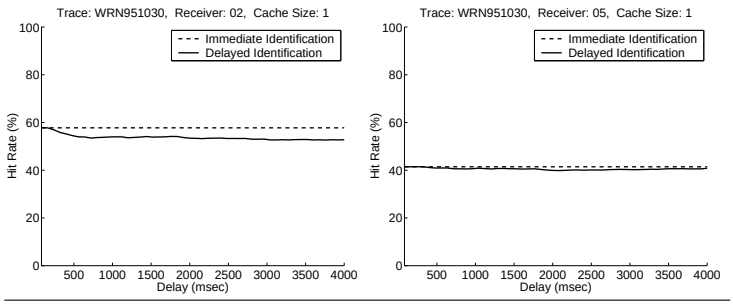
Early detection schemes may potentially allow the reliable multicast protocol to identify the location of the leading losses of a burst sooner, thus benefiting the location prediction of the trailing losses of the burst. Alternatively, it may be beneficial to treat all the losses that comprise particular loss bursts collectively. For instance, upon detection of a loss burst, a receiver could recover the first loss of the burst and, subsequently, recover the remaining losses of the burst in the manner in which the first loss of the burst was recovered.

Figure 7 presents the hit rates of our loss location prediction scheme for the concrete link trace representation for the same 6 traces. The effects of delayed loss detection for the concrete link loss representation are similar to, yet less severe than, those observed for the virtual link loss representation.

4.3 Delayed Detection/Delayed Identification

In this section, we observe the degree to which the delay in identifying the location of losses affects the per-receiver hit rates of our loss location prediction scheme. We define the loss location identification delay to be the time that

Figure 8 Virtual Link Trace Representation — Prediction hit rates wrt loss identification delay, cache of size 1 (Trace WRN951030).



elapses from the time a loss is detected to the time its location is identified.

We first consider the virtual link trace representation. Figures 8 and 9 present the hit rates of a couple of receivers of trace WRN951030 with respect to the loss location identification delay for caches of size 1 and 10, respectively. These plots depict the per-receiver hit rates that are least and most affected by the loss location identification delay for the given trace. The plots for the remaining receivers and traces are similar. The dashed lines correspond to the hit rates achieved with delayed detection and immediate loss location identification (presented in Figure 5).

Figures 8 and 9 present the hit rates obtained for a delay of up to 4 seconds. We presume that a loss's location can be identified within the amount of time required to recover from losses. Several reliable multicast protocols, such as SRM [4] and LMS [11], recover from the vast majority of losses well within 3–4 round-trip-times (RTTs), on average. Thus, presuming a 1 second RTT upper bound, a 4 second upper bound on the location identification delay is reasonable.

We observe that the hit rates of the loss location prediction scheme only slightly decrease as the loss location identification delay increases and the available loss location information becomes less recent. This is because 4 seconds is a short enough time interval for locality to still hold. The hit rates achieved with a cache of size 1 are more sensitive to the loss location identification delay. This is because the larger cache sizes favor the prediction of more frequently lossy locations (links); that is, locations (links) that are probabilistically better candidates for being liable for losses.

We now consider the concrete link trace representation. Figures 10 and 11 present the hit rates of a couple of receivers of trace WRN951030 as the loss location identification delay increases for caches of size 1 and 10, respectively. Again, these plots depict the per-receiver hit

Figure 9 Virtual Link Trace Representation — Prediction hit rates wrt loss identification delay, cache of size 10 (Trace WRN951030).

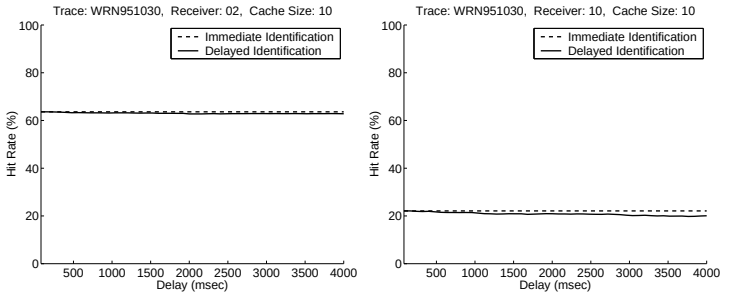
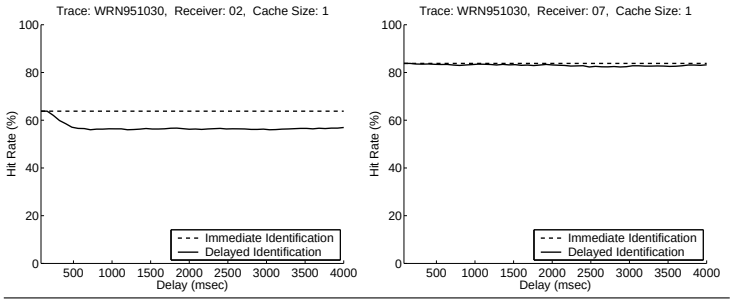


Figure 10 Concrete Link Trace Representation — Prediction hit rates wrt loss identification delay, cache of size 1 (Trace WRN951030).

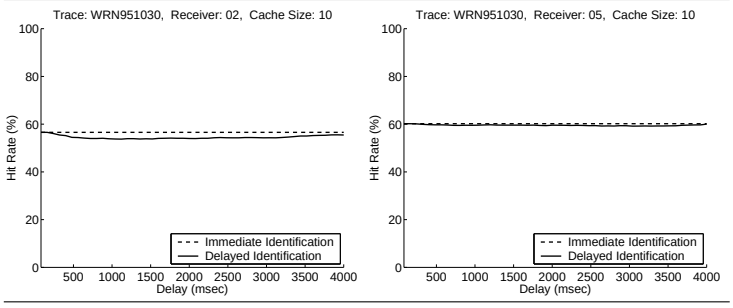


rates that are least and most affected by the loss location identification delay for the given trace. The effects of delayed loss location identification for the concrete link trace representation are similar to those observed for the virtual link trace representation.

4.4 Shared Hit Rates

In this section, we evaluate the degree to which receivers that share losses predict the same loss locations. Throughout this section, we assume that losses are detected upon the receipt of later packets and that loss location identification is immediate.

Figure 11 Concrete Link Trace Representation — Prediction hit rates wrt loss identification delay, cache of size 10 (Trace WRN951030).



Each receiver's loss location prediction may be either upstream, accurate, or downstream of the estimated location of the loss. We designate such predictions as high, accurate, and low, respectively. In the case of the virtual link trace representation, we determine whether a loss location prediction is high, accurate, or low by comparing the predicted virtual link's loss pattern to the observed loss pattern. Loss patterns dictate the set of hosts that share the loss. Thus, the loss location prediction is high, accurate, or low when the predicted set of hosts that share the loss is a strict superset, equal to, or a strict subset of the observed set of hosts that share the loss. When the predicted or observed loss patterns correspond to simultaneous losses on multiple links of the IP multicast tree, the predicted and observed sets of hosts sharing the loss may be incomparable; that is, they may be neither equal, nor strict supersets or subsets of each other. In such cases, we say that the predicted and the estimated loss locations are *incomparable*. In the case of the concrete link trace representation, the notions of upstream, accurate, and downstream are dictated by the IP multicast tree topology. For the concrete link trace representation, predicted and estimated loss locations are never incomparable; they both correspond to links on the path from the source to the given receiver.

We classify the loss location predictions into four types: i) *consistent high predictions*, where all the receivers that share the loss predict that the loss location is upstream of the estimated loss location, ii) *consistent accurate predictions*, where all the receivers that share the loss accurately predict the loss location, iii) *consistent low predictions*, where all the receivers that share the loss predict that the loss location is downstream of the estimated loss location, and iv) *inconsistent predictions*, where the receivers that share the loss predict a combination of upstream, accurate, downstream, and incomparable locations. We refer to consistent accurate predictions as *shared hits* and to the percentage of losses for which predictions are consistent and accurate as the *shared hit rate*.

In terms of the loss recovery process, consistent high predictions overestimate the extent of the loss. Thus, retransmission requests may be sent to receivers that are part of a larger subtree of the IP multicast tree than required. In such cases, the recovery may be exposed to a larger region of the IP multicast tree than required and incur unduly latency. Consistent low predictions underestimate the extent of the loss. In such cases, retransmission requests may be addressed to hosts that share the loss. The recovery based on such predictions would thus fail. The effect of inconsistent predictions would depend on how predictions are used by the recovery scheme at hand; that is, it would depend on which of the receivers suffering the loss

Figure 12 Virtual Link Trace Representation — Consistent High/Accurate/Low and Inconsistent Prediction Percentages.

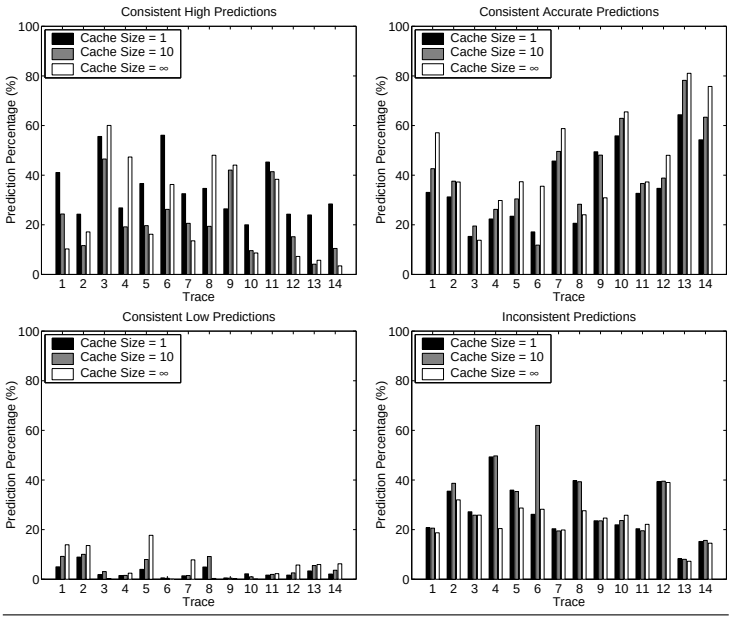
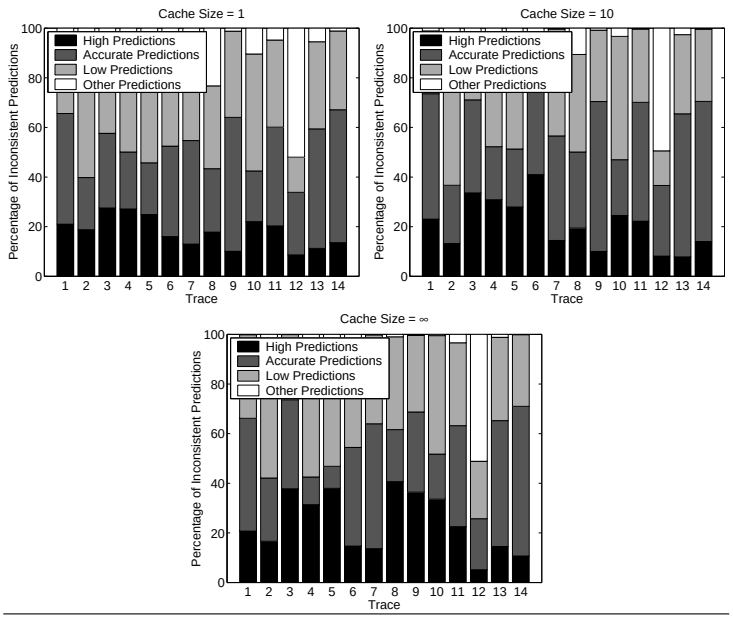


Figure 13 Virtual Link Trace Representation — Mean inconsistent prediction distributions.



would actually transmit retransmission requests.

We first consider the virtual link trace representation. Figure 12 presents the distribution of the predictions of our loss location prediction scheme among consistent high/accurate/low and inconsistent prediction types. With a cache of size 10, the shared hit rates always exceed 10% and exceed 35% for half the traces.

We now examine what type of predictions individual receivers make when predictions are inconsistent. For each inconsistent prediction, we compute the percentage of receivers that share the loss and predict upstream, accurate, or downstream locations. The average of these percentages over all inconsistent predictions made in each trace are presented in Figure 13. The percentage of receivers that generate upstream and accurate predictions often account for more than half of the receivers sharing the loss. This indicates that more than half of the losses resulting in inconsistent predictions may be recovered through caching.

We now consider the concrete link trace representation. Figure 14 presents the distribution of the predictions of our loss location prediction scheme among consistent high/accurate/low and inconsistent prediction types. The shared hit rates of the prediction schemes for the concrete link trace representation are substantially higher than those for the virtual link trace representation.

The shared hit rates for all cache sizes exceed 25% for all traces. For most of the traces, the cache of size 10 outperforms the cache of size 1. Moreover, its shared hit rate exceeds 70% for half the traces. The infinite cache performs similarly to the cache of size 10. This indicates that, in the case of the concrete link trace representation, a single loss location is responsible for a large percentage of the losses suffered by most of the receivers.

We expect that as the size of the reliable multicast group increases and as the IP multicast transmissions become longer-lived, i) several links will be responsible for large percentages of the losses suffered by individual receivers, and ii) the links responsible for a large percentage of the losses suffered by individual receivers will change over time. Smaller cache sizes would in such cases be preferable so as to adapt quicker to changing loss characteristics and accommodate multiple highly lossy links.

A comparison of Figures 12 and 14 suggests that the precise identification of the links on which losses occur may be highly beneficial to the effectiveness of caching. Reliable multicast protocols that feature local recovery schemes may be particularly suitable both for precisely identifying the links on which losses occur and for effectively exploiting this information by recovering from losses locally.

4.5 Optimal Cache Size

Finally, we examine the effect of the cache size on the shared hit rate. Figure 15 presents the shared hit rates of the loss location prediction scheme for the virtual link trace representation for different cache sizes. We present the plots for 6 out of the 14 traces; the plots for the other traces are similar. For many of the traces, a cache of finite size outperforms the infinite cache. In particular, a cache

Figure 14 Concrete Link Trace Representation — Consistent High/Accurate/Low and Inconsistent Prediction Percentages.

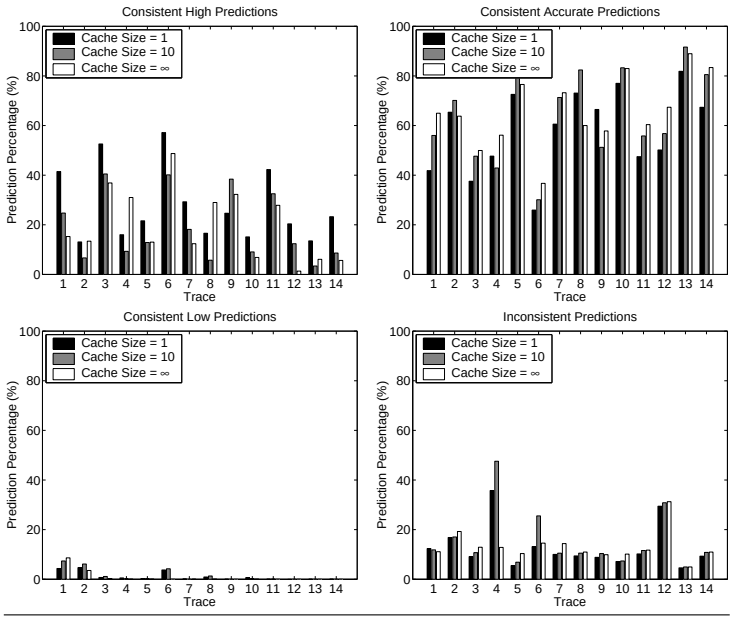
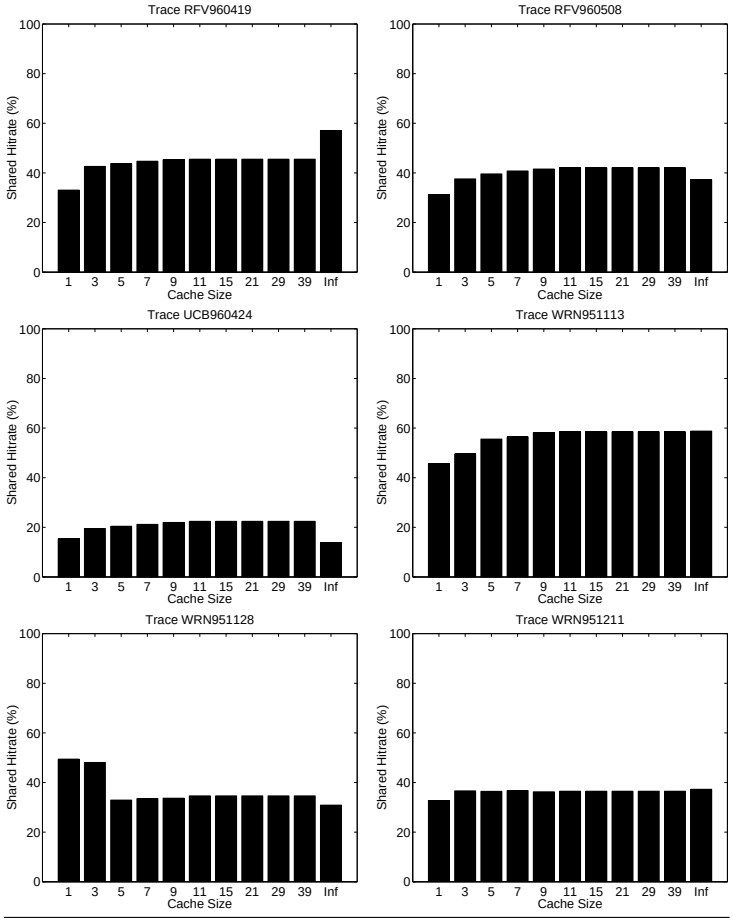


Figure 15 Virtual Link Trace Representation — Consistent accurate hit rates wrt cache size.



size of 11 performs well in comparison to all other cache sizes for most of the traces.

Figure 16 Concrete Link Trace Representation — Consistent accurate hit rates wrt cache size.

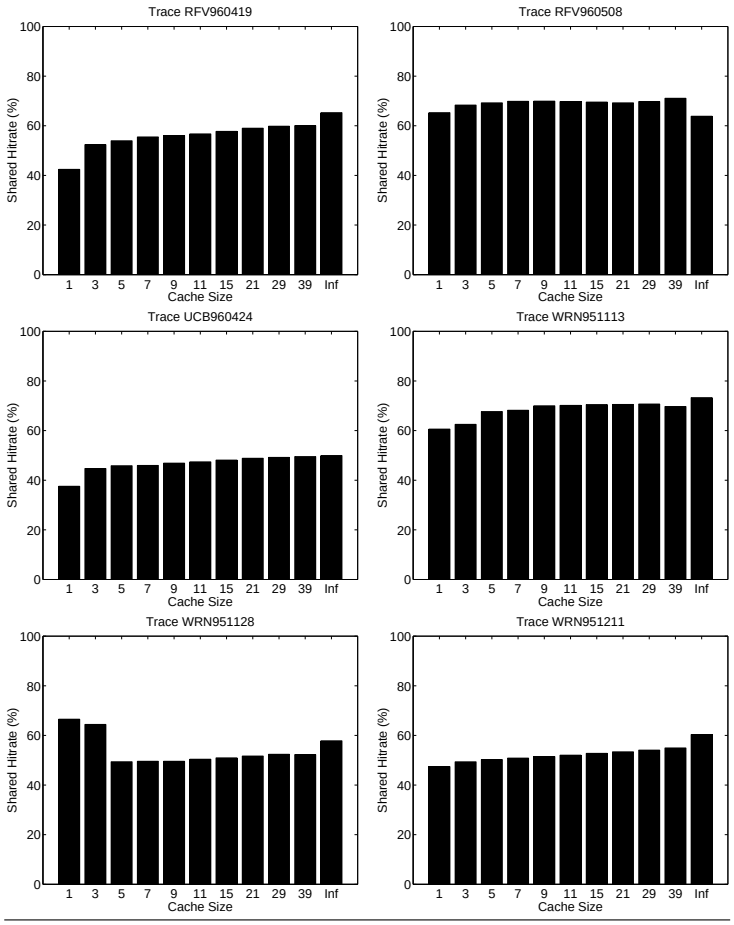


Figure 16 presents the shared hit rate of the loss location prediction scheme for the concrete link trace representation for different cache sizes. Again, we present the plots for 6 out of the 14 traces; the plots for the other traces are similar. For many of the traces, the shared hit rate increases as the cache size grows. This suggests that, in the case of the concrete link trace representation, the losses suffered by individual receivers occur predominantly on single links. In the case of the concrete link trace representation, cache sizes of 11 and 15 perform well for most of the traces.

In summary, modest cache sizes of 11 or 15 perform well for most of the traces and both trace representations. This indicates that effective caching in multicast loss recovery is achievable without prohibitive resource requirements.

5 Summary, Conclusions, and Future Work

In this paper, we proposed exploiting packet loss locality within existing or novel reliable multicast protocols through caching. We presented a methodology for esti-

mating the potential effectiveness of caching in multicast loss recovery. Our methodology involved analyzing the performance of a caching-based loss location prediction scheme. We applied our methodology to the IP multicast transmission traces of Yajnik *et al.* [14] and observed that packet loss locality is indeed substantial.

Presuming immediate loss detection and loss location identification, per-receiver hit rates in most cases exceeded 40% and often exceeded 80%. The delay in detecting losses did not substantially affect the per-receiver hit rates, except in the cases where the receivers suffer long loss bursts. The delay in identifying the locations of losses did not substantially affect the hit rates of individual receivers. In most cases, a cache of size 10 outperformed a cache of size 1. The infinite cache performed similarly to the cache of size 10 only when the losses suffered by individual receivers occur predominantly at single locations. Smaller cache sizes respond quicker to changing loss characteristics and achieve higher hit rates for traces involving multiple highly lossy links.

We also observed substantial shared hit rates. In the case of the virtual link trace representation, shared hit rates ranged from 10% to 80%. The shared hit rate for a cache size of 10 exceeded 35% for half the traces. In the case of the concrete link trace representation, shared hit rates ranged from 25% to 90%. The shared hit rate for a cache size of 10 exceeded 70% for half the traces. In our analysis of the effect of cache size on the shared hit rate, modest caches of size 11 or 15 achieved high shared hit rates for most of the traces and both trace representations.

Shared hit rates indicate the percentage of losses whose recovery latency and overhead may be reduced through caching. Thus, they are a good indication of the expected effectiveness of caching. The shared hit rates achieved by our loss location prediction scheme suggest that caching can be very effective.

The work presented in this paper may be extended in several directions. First, our methodology can be applied to IP multicast transmissions of larger group size and longer duration. Such work will reveal whether the effectiveness of caching scales. Second, caching schemes that exploit locality can be designed and incorporated in either existing or novel reliable multicast protocols. Finally, the effectiveness of such schemes can be evaluated through simulation or deployment and compared to the expected effectiveness indicated by our observations. We are currently in the process of enhancing SRM with the caching-based expedited recovery scheme briefly described in Section 2.

Acknowledgments

We thank Nancy Lynch for helpful discussions, comments, and suggestions. We thank Yajnik *et al.* [14] for making their multicast transmission traces available online.

References

[1] Bolot, J.-C., Crépin, H., and Vega Garcia, A. Analysis of Audio Packet Loss in the Internet. In *Proc. 5th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)* (Durham, NH, Apr. 1995), vol. 1018 of *Lecture Notes in Computer Science*, pp. 154–165.

[2] Cáceres, R., Duffield, N. G., Horowitz, J., and Towsley, D. F. Multicast-Based Inference of Network-Internal Loss Characteristics. *IEEE Transactions on Information Theory 45*, 7 (Nov. 1999), 2462–2480.

[3] Chayat, R., and Rom, R. Applying Deterministic Feedback Suppression to Reliable Multicasting Protocols. In *Proc. 10th IEEE International Conference on Computer Communications and Networks (IEEE/ICCCN'01)* (Scottsdale, AZ, Oct. 2001), pp. 81–88.

[4] Floyd, S., Jacobson, V., McCanne, S., Liu, C.-G., and Zhang, L. A Reliable Multicast Framework For Light-Weight Sessions And Application Level Framing. *IEEE/ACM Transactions on Networking 5*, 6 (Dec. 1997), 784–803.

[5] Handley, M. An Examination of MBone Performance. Research Report RR-97-450, University of Southern California (USC)/Information Sciences Institute (ISI), Jan. 1997.

[6] Holbrook, H. W., Singhal, S. K., and Cheriton, D. R. Log-Based Receiver-Reliable Multicast For Distributed Interactive Simulation. In *Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM Special Interest Group on Data Communication (ACM/SIGCOMM'95)* (1995), ACM Press, New York, pp. 328–341.

[7] Li, D., and Cheriton, D. R. OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol. In *Proc. 6th IEEE International Conference on Network Protocols (IEEE/ICNP'98)* (Austin, Texas, 1998), pp. 237–245.

[8] Lin, J. C., and Paul, S. RMTP: Reliable Multicast Transport Protocol. In *Proc. 15th Annual Joint Conference of the IEEE Computer and Communications Societies, Networking the Next Generation (IEEE/INFOCOM'96)* (San Francisco, CA, Mar. 1996), vol. 3, pp. 1414–1424.

[9] Livadas, C., Keidar, I., and Lynch, N. A. Designing a Caching-Based Reliable Multicast Protocol. In *Proc. International Conference on Dependable Systems and Networks (IEEE/DSN'01), Fast Abstracts Supplement* (Göteborg, Sweden, July 2001), IEEE Computer Society, pp. B44–B45.

[10] Livadas, C., Keidar, I., and Lynch, N. A. Caching Enhanced Scalable Reliable Multicast (CESRM). Unpublished Manuscript, July 2002.

[11] Papadopoulos, C., Parulkar, G., and Varghese, G. An Error Control Scheme For Large-Scale Multicast Applications. In *Proc. 17th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE/INFOCOM'98)* (San Francisco, CA, Mar. 1998), vol. 3, pp. 1188–1196.

[12] Paul, S., Sabnani, K. K., Lin, J. C., and Bhattacharyya, S. Reliable Multicast Transport Protocol (RMTP). *IEEE Journal on Selected Areas in Communications 15*, 3 (Apr. 1997), 407–421.

[13] Semeria, C., and Maufer, T. Introduction to IP Multicast Routing. Internet-Draft (Informational), Internet Engineering Task Force, July 1997. Also, Technical Memo, Networking Solutions Center, 3Com Corporation.

[14] Yajnik, M., Kurose, J., and Towsley, D. Packet Loss Correlation in the MBone Multicast Network. In *Proc. Global Telocommunications Conference (IEEE/GLOBECOM'96), Communications: The Key to Global Prosperity* (London, England, Nov. 1996), pp. 94–99.

[15] Yajnik, M., Moon, S. B., Kurose, J., and Towsley, D. Measurement and Modeling of the Temporal Dependence in Packet Loss. In *Proc. 18th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE/INFOCOM'99)* (New York, NY, Mar. 1999), vol. 1, pp. 345–352.