Incremental Deployment Strategies for Router-Assisted Reliable Multicast

Xinming He, Christos Papadopoulos, *Member, IEEE,* Pavlin Radoslavov

***Abstract*— Incremental deployment of a new network service or protocol is typically a hard problem, especially when it has to be deployed at the routers. First, an incrementally deployable version of the protocol may be needed. Second, a systematic study of the performance impact of incremental deployment is needed to evaluate potential deployment strategies. Choosing the wrong strategy can be disastrous, as it may inhibit reaping the benefits of an otherwise robust service and prevent widespread adoption.**

We focus on two router-assisted reliable multicast protocols, namely PGM and LMS. Our evaluation consists of three parts: (a) selection and classification of deployment strategies; (b) definition of performance metrics; and (c) systematic evaluation of deployment strategies. Our study yields several interesting results: (a) the performance of different deployment strategies varies widely; for example, with some strategies, both PGM and LMS approach full deployment performance with as little as 5% of the routers deployed; other strategies require up to 80% deployment to approach the same level; (b) our sensitivity analysis reveals relatively small variation in the results in most cases; and (c) the impact associated with partial deployment is different for each of these protocols; PGM tends to impact the network, whereas LMS the endpoints. Our study clearly demonstrates that the choice of a strategy has a substantial impact on performance.

***Index Terms*— incremental deployment, router-assisted services, reliable multicast.**

I. Introduction

As the Internet evolves, new functions may be inserted to network routers to assist emerging services. Examples include router-assisted reliable multicast services [1], [2], Anycast [3], Concast [4], [5], security services [6], [7], [8], and other end-to-end services [9], [10]. Due to the large scale and inherent heterogeneity of the Internet, new functions are typically deployed gradually among Internet routers. Hence, the network may go through extended periods with partial deployment. During this state, the performance and utility of the service may suffer. It is important to follow proper approaches to select routers for deployment, as thoughtful deployment strategies may tip the scale toward success, whereas careless strategies may hamper an otherwise sound service.

Selecting the right deployment strategy is a hard problem because many variables are involved. To date, there has been no systematic methodology to study incremental deployment techniques for network services. Thus, network planners and operators have to resort to ad-hoc methodologies when a new service is to be deployed.

In this paper, we define a methodology for evaluating incremental deployment strategies for a class of network services, namely *router-assisted reliable multicast*. Our methodology consists of the following three parts: (a) selection and classification of deployment strategies; (b) definition of metrics to measure performance; and (c) systematic evaluation of deployment strategies. We use the following guidelines in defining our methodology. First, we strive for good coverage of the problem space by evaluating numerous deployment strategies, which include both service-specific strategies (*e.g.*, strategies that take advantage of the multicast tree structure for multicast services), and service-independent strategies (*e.g.*, strategies that deploy a service at the AS border routers). Then, we define a series of metrics that are essential for performance measurement. Finally, our evaluation of deployment strategies is done over a large-scale mapped Internet topology to avoid potential artifacts from topology generators.

Our work offers two main contributions. The first is the definition of the methodology itself, which may be adapted and reused in studies for incremental deployment of other router-assisted network services. The second is the results, which are important because they provide clues to help network planners answer questions such as: (a) what is the best deployment strategy for my network and application? (b) what is the minimum level of deployment such that the benefits justify the cost? and (c) how many routers need to be deployed before we begin to experience diminishing returns?

We have selected two router-assist reliable multicast schemes, namely PGM [1] and LMS [2], because their specification includes detailed incremental deployment methods. Note that in this study we are not evaluating the merits of router assistance nor carry out a comparative evaluation of these protocols. Such studies have been done elsewhere [1], [2], [11]. We are simply interested in how performance of these protocols varies with incremental deployment.

Our study helps understand both the general behavior of these protocols under partial deployment and the specific issues raised by each protocol. PGM and LMS differ significantly in their operation and thus behave differently under partial deployment. For example, PGM routers aggregate NAKs and guide retransmissions where needed, whereas LMS delegates these actions to the receivers with minimal assistance from the routers; additionally, PGM retransmissions typically emanate from the sender, whereas in LMS they come from

§ A note to the reviewers: An earlier version of this paper was published in Infocom 2003. This version has been updated to include another important metric, namely recovery latency, and the assumption of optimal NAK suppression in PGM is replaced by incorporating the dynamic adjustment of NAK back-off interval as specified in PGM protocol.

X. He and C. Papadopoulos are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (email: xhe@usc.edu; christos@isi.edu).

P. Radoslavov is with International Computer Science Institute, Berkeley, CA, USA (email: pavlin@icsi.berkeley.edu).

the receivers. Evaluating these protocols within the same framework helps distinguish their behavior better.

An earlier study has investigated the performance of LMS under incremental deployment [12], but in a more limited setting. Our current work is more systematic and contains several significant improvements, including: (a) the definition of a methodology for systematic study of incremental deployment; (b) the study of both LMS and PGM under incremental deployment, whereas the previous work studied only LMS; (c) the use of a mapped Internet topology of over 27,000 nodes obtained by a topology mapping software [13], whereas the earlier work used much smaller synthetic topologies (about 400 nodes) generated by the GT-ITM [14]; (d) the investigation of more realistic deployment strategies, such as strategies based on router fanout and AS size.

Our study reveals some interesting results. First, of the five evaluation metrics we use, only a few show strong impact due to partial deployment. Second, for the best deployment strategies the performance for both protocols approaches full deployment levels at only a fraction of deployment, which can be as little as 5%. This is very encouraging because it implies that virtually all the benefit of the service can be realized very early in the deployment phase. Third, there is a significant difference among deployment strategies, with some clear winners and some unexpected losers. Fourth, our sensitivity analysis reveals only small variation in most cases. Finally, the results show that the impact of deployment differs significantly between the two protocols. PGM tends to place the burden in the network, whereas in LMS the impact is on the endpoints.

The rest of the paper is organized as follows. Section II describes the various incremental deployment strategies in our methodology. Section III presents an overview of the data recovery mechanisms in PGM and LMS, and the definition of our evaluation metrics. Simulation results are presented in Section IV, followed by sensitivity analysis in Section V. Section VI reviews related work, and Section VII concludes the paper.

II. Incremental Deployment Strategies

A router-assisted service requires additional functions deployed at network routers through upgrade of either software, hardware, or both. While it is easy to deploy one router, it is a long process to change all Internet routers due to the large scale and inherent heterogeneity of the Internet. There are many different approaches for network operators to select routers for deployment. In this section, we define and classify a number of incremental deployment strategies over the entire space.

We first classify deployment strategies into two main categories: *Network-Aware* strategies or *service-independent* strategies that utilize only general information about the network structure, and *Multicast-Tree-Aware* strategies or *service-specific* strategies that take advantage of the multicast tree information. Within these two categories, strategies can be further subdivided based on factors such as AS (Autonomous System) size, AS tier level, network connectivity, proximity to the sender and receivers, and multicast tree connectivity.

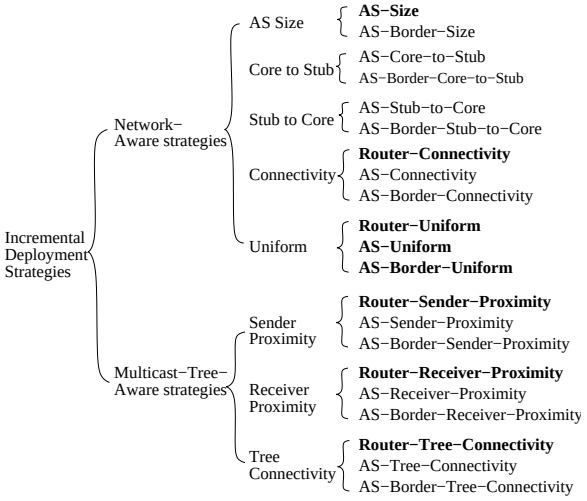

Fig. 1. Classification of Incremental Deployment Strategies (Strategies studied in this paper are in bold)

We consider three deployment granularities: (a) router, (b) all border routers in an AS, and (c) the entire AS. A router is a natural deployment unit. Border routers are typically good traffic aggregation points and seem a logical intermediate step before full AS deployment. Figure 1 shows our classification, which we explain next.

A. Network-Aware Deployment Strategies

Network-Aware strategies utilize information about the general network structure, such as router-level and AS-level structure. They are service independent as they do not rely on information for specific services, like knowledge on the multicast tree structure. In this category, we identify the following strategies:

- *AS Size Strategies.* With the AS size-based strategies, the largest AS gets deployed first, followed by the smaller ASs, sorted by the number of routers inside the AS. Such strategies assume that large ISPs will deploy the new service first, followed by smaller ISPs.
- *Core-to-Stub and Stub-to-Core Strategies.* With the Core-to-Stub strategies, a service is first deployed at the Internet core ASs, and then pushed toward stub ASs. Conversely, with the Stub-to-Core strategies a service is first deployed at stub ASs, and then pushed toward the core. These approaches hold different views as to how deployment will proceed. The core-first strategies assume that backbone ISPs see significant added value from the new service and support it immediately. The stub-first strategies assume that smaller, agile ISPs move quickly to adopt the new service, leading to rich deployment at the edges.
- *Connectivity Strategies.* With these strategies, routers or ASs with the highest network connectivity (fanout) get deployed first, followed by less-connected routers or ASs. The intuition behind such strategies is that better connectivity leads to higher probability that the service will touch many flows.
- *Uniform Strategies.* Uniform strategies select deployment units (routers, border routers, full AS) with uniform

probability. Uniform strategies are the simplest of the strategies we study and they capture the scenario where deployment happens without any coordination. Results from such strategies form the baseline for comparing other strategies.

B. Multicast-Tree-Aware Deployment Strategies

For multicast services knowledge of the multicast tree structure is very important. Compared with Network-Aware deployment strategies, Multicast-Tree-Aware strategies use information about multicast tree structure. While foregoing generality, such strategies should be studied for two reasons: (a) often the basic structure of the multicast tree is known a priori (*e.g.,* large content distribution networks, where the internal structure of the tree remains largely unchanged over the time); and (b) such strategies are expected to have better performance as they can take advantage of the extra information and help calibrate other strategies. In this paper we study the following Multicast-Tree-Aware deployment strategies:

- *Sender-Proximity Strategies.* With these strategies a service is deployed in routers or ASs based on their distance from the sender starting near the sender and moving toward the leaves. Such strategies may be adopted when the sender charges the receivers for its service, in which case there is a strong incentive to optimize performance in the home network. An example might be a video-on-demand service.
- *Receiver-Proximity Strategies.* These strategies deploy a service at routers or ASs based on their distance from the receivers. The rationale behind such strategies is that receivers independently exert influence on their ISPs to deploy the service. An example might be a new caching service.
- *Tree-connectivity Strategies.* With these strategies a service is deployed in routers or ASs based on their connectivity in the multicast tree. For example, routers are sorted based on the number of their downstream interfaces and the new service is deployed in the routers with the largest fanout first. Similar to network connectivity, such strategies are expected to extract the maximum benefit because they touch the denser parts of a multicast tree first.

In our case study of router-assist reliable multicast we have carried out extensive investigation on all the strategies except the Core-to-Stub and Stub-to-Core strategies due to the lack of AS tier information in our Internet topology map. Due to space limitations, we present only the results of eight strategies (highlighted in bold in Figure 1). In our discussion of the results, we include results from other strategies where appropriate.

III. Router-Assisted Reliable Multicast Schemes

The key design challenge for reliable multicast is the scalable recovery of packet losses. The main impediments to scale are *implosion* and *exposure.* Implosion occurs when a packet loss triggers redundant data recovery messages. These messages may swamp the sender, the network, or the receivers. Exposure occurs when a retransmitted packet is delivered to unwanted receivers, wasting network and receiver resources.

Router-assisted reliable multicast schemes use assistance from the network to overcome these problems. Such assistance comes in two forms: (a) ensuring congruency between the data recovery tree and the underlying multicast tree, and (b) allowing fine-grain multicast that helps direct retransmissions only where needed. In this paper, we consider two router-assisted reliable multicast schemes, PGM [1], and LMS [2].

A. PGM

Below is a brief introduction to the basic operation of PGM. A more detailed description can be found in the PGM specification [1].

In PGM, the sender periodically multicasts Source Path Messages (SPMs). Those messages are processed hop-by-hop by PGM-capable routers, and are used by each receiver or PGM router to learn the address of its upstream PGM neighbor. When a receiver detects a packet loss, it observes a back-off period and then unicasts a NAK to its upstream PGM neighbor. Upon receiving a NAK, a PGM router creates repair state, which includes the sequence number of the lost packet and the interface the NAK was received on. In addition, the PGM router acknowledges the NAK by multicasting a NAK Confirmation (NCF) on the interface the NAK was received on. NCFs are also used to suppress other pending NAKs. The router in turn unicasts a NAK to its upstream PGM neighbor if it has not done so for the lost packet. This is again followed by a NCF from the upstream PGM neighbor. This process repeats until the NAK reaches the sender.

After the sender receives a NAK, it multicasts a repair packet. The sender may delay the transmission of the repair packet to avoid repeated retransmissions - we will describe it shortly. Non-PGM routers forward the repair packet as an ordinary multicast packet, but PGM routers forward it only on interfaces where a NAK was previously received. [1]

The sender may delay a repair packet to allow repair branches, grown by NAKs starting from the leaves, to reach the main tree. If a repair packet is sent too early (typically triggered by a nearby receiver) it may tear down the recovery tree before distant branches are grafted. Partial branches, however, will re-grow the repair tree back to the sender, triggering a repeated retransmission. Repeated retransmissions waste resources so PGM allows the sender to delay a repair packet up to twice the greatest propagation delay in the loss neighborhood. This delay is called the *sender holding time*. Setting this value is a task left to the network administrator.

PGM incorporates a dynamic adjustment of the NAK back-off interval. This is a per-interface parameter that helps suppress duplicate NAKs received along the interface. The NAK back-off interval is periodically advertised by the router to downstream receivers below the interface, which upon packet loss will observe a back-off time randomly selected between 0 and the advertised NAK back-off interval. Note that PGM routers do not observe a back-off time when forwarding NAKs

[1]Note that PGM permits Designated Local Repairers (DLRs) to retransmit missing data on behalf of the source. We do not consider DLRs in our study.

upstream. To set this value, each PGM-capable router monitors the number of duplicate NAKs received after each loss. The NAK back-off interval is doubled if one or more duplicate NAKs are received, and is cut in half if no duplicate NAKs were received during the last N losses. In addition, PGM requires that upper and lower bounds be established for the NAK back-off interval. PGM also includes a periodic poll mechanism to estimate the receiver population attached to an interface, which helps track membership changes.

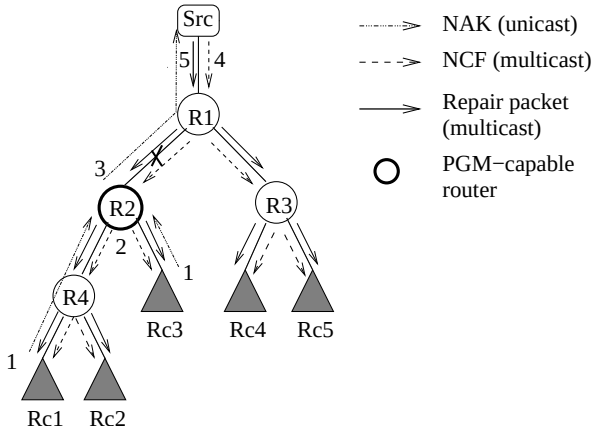

Fig. 2. An example for the PGM protocol

The example in Figure 2 illustrates data recovery in PGM under partial deployment. Assume that only $R2$ is PGM-capable, and that a packet is lost between $R1$ and $R2$. Upon detecting the loss, $Rc1$, $Rc2$, and $Rc3$ set their back-off timers. Suppose that $Rc1$'s timer expires first; therefore $Rc1$ unicasts a NAK to $R2$ (step 1). Upon receiving this NAK, $R2$ multicasts a NCF along the interface to $R4$ (step 2), and then sends a NAK to the source (step 3). Suppose the NCF reaches $Rc2$ before its timer expires; hence, it cancels the NAK from $Rc2$. Similar to $Rc1$, $Rc3$ also unicasts a NAK to $R2$, to which $R2$ responds with another NCF. However, $R2$ does not propagate another NAK to the source. When the source receives a NAK, it first multicasts a NCF (step 4), followed by a repair packet (step 5). Since $R1$ is not PGM-capable, both the NCF and the repair are forwarded to $R2$ and $R3$. $R2$ does not propagate the NCF, but propagates the repair, as dictated by the repair state created earlier by NAKs. $R3$, however, forwards both NCF and repair to $Rc4$ and $Rc5$, exposing them to unnecessary recovery messages.

Inefficiencies due to partial PGM deployment can arise for two reasons: (a) non-PGM routers forward all multicast packets, including NCFs and repair packets, along all downstream interfaces, which creates opportunities for exposure; and (b) sparse deployment may attract many downstream routers to bind with the same upstream router creating opportunities for implosion.

B. LMS

The original description of LMS [2] sketched an incremental deployment methodology. Here we refine that methodology and provide a more detailed incremental deployment specification.

Similar to PGM, in LMS the sender periodically multicasts SPMs to help LMS routers and receivers discover their upstream LMS neighbors. Lost packets are retransmitted by *repliers* which are simply group members willing to assist with the packet recovery process. Each LMS router selects a replier among its downstream candidates, based on some cost measurement, such as distance or loss rate. When a receiver detects a packet loss, it unicasts a NAK to its upstream LMS router. Upon receiving a NAK, the LMS router forwards the NAK according to the following rules: if the NAK was originated from its replier, the router forwards the NAK to its upstream LMS neighbor; otherwise, the router is the *turning point* for that NAK, therefore it inserts its own address and the interface the NAK arrived on before unicasting the NAK to the replier.

When a replier (or the sender) receives a NAK and has the requested data, the replier unicasts a repair packet directly to the NAK originator (we assume that the sender always has the repair data). If the replier does not have the requested data, the replier records the NAK turning point and waits for the repair packet. If the replier receives the repair packet via multicast, that means some other upstream replier has taken care of the repair process, hence any local repair state in this replier is purged. If the repair is received via unicast, the replier delivers the repair to each recorded turning point using *directed multicast*. A directed multicast consists of two phases: (a) a unicast of the repair packet to the turning point router, and (b) a multicast of the repair by the turning point router on the NAK's original incoming interface (contained in the repair).

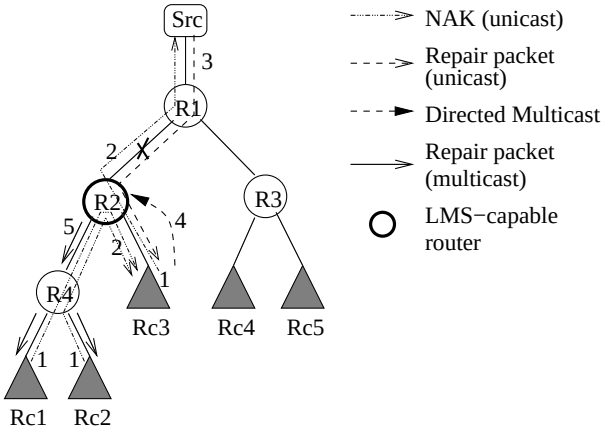

Fig. 3. An example for the LMS protocol

The example in Figure 3 illustrates data recovery in LMS. Assume that only $R2$ is LMS-capable, and it has selected $Rc3$ as its replier. Suppose that a packet is lost on the $R1 - R2$ link. Upon detecting the loss, $Rc1$, $Rc2$, and $Rc3$ each unicast a NAK to $R2$ (step 1). $R2$ forwards the NAKs from $Rc1$ and $Rc2$ to replier $Rc3$, but the NAK from $Rc3$ is forwarded to the source (step 2). The NAKs from $Rc1$ and $Rc2$ have the same turning point, therefore $Rc3$ records the first NAK and discards the other. After the source receives the NAK from $Rc3$ it unicasts a repair to $Rc3$ (step 3). $Rc3$ in turn sends a

directed multicast to $R2$ (step 4). The repair then is delivered via multicast to $Rc1$ and $Rc2$ (step 5).

Inefficiencies due to partial LMS deployment can arise for two reasons: (a) a turning point may be established higher in the tree than where loss occurred, leading to possible exposure; and (b) sparse deployment may result in a large number of NAKs forwarded to one replier or the sender, leading to implosion.

C. Metric Space

Incremental deployment may have a strong impact on the performance of network services and therefore metrics must be carefully defined to capture that impact. In our case study we select the following metrics that capture the two major obstacles to scalability in reliable multicast, *implosion* and *exposure*, in addition to data recovery latency. To calculate the metrics, we first define a packet loss sequence as $l_1, l_2, ..., l_N$, where l_i is the link where the i-th packet loss happens, and N is the total number of packet losses considered in the calculation. For simplicity, we assume the recovery process for different packet losses does not overlap.

- *Average Normalized Data Overhead.* Normalized Data Overhead is defined as the ratio of network resources used by repair packets (in terms of link hops) to recover a packet loss, and the size of the subtree (in number of links) that is affected by the packet loss. In the ideal case, the normalized data overhead would be 1.0 (*i.e.*, under full deployment and when the node right above the subtree sends a single multicast packet to the subtree). Average Normalized Data Overhead is calculated by averaging Normalized Data Overhead across all packet losses in the packet loss sequence:

$$AvgNormDataOverhead = \frac{\sum_i \frac{Data(i)}{Subtree(i)}}{N}$$

  where i is from 1 to N, $Data(i)$ is the number of links traversed by the repair packet to recover the i-th packet loss, $Subtree(i)$ is the size of the subtree that does not receive the packet due to the i-th packet loss.
- *Average Normalized Control Overhead.* Similar to Normalized Data Overhead, Normalized Control Overhead is defined as the ratio of the amount of network resources used by control packets (NAKs and NCFs) to recover a packet loss, and the size of the subtree that does not receive the packet. We consider a ratio of 1.0 to be optimal, even though theoretically this is not the lowest ratio. Similarly, Average Normalized Control Overhead is calculated as follows:

$$AvgNormControlOverhead = \frac{\sum_i \frac{Control(i)}{Subtree(i)}}{N}$$

  where i, N, and $Subtree(i)$ are defined as above, and $Control(i)$ is the total amount of control traffic that is generated to recover the i-th packet loss.
- *Maximum Average NAKs.* This is the maximum of the average number of NAKs received by a node across all packet losses. It is a measure of the worst case *sustained* implosion at any node in the multicast tree, and is calculated as follows:

$$MaxAvgNAKs = \max_m(\frac{\sum_i NAKs(m,i)}{N})$$

  where m is a node in the multicast tree (either the sender, a receiver, or a router), and $NAKs(m,i)$ is the number of NAKs received by node m for the i-th packet loss.
- *Maximum Peak NAKs.* Maximum Peak NAKs is the maximum number of NAKs received by a node for a single packet loss. This is a measure of the worst case *instantaneous* implosion at any node, and is calculated as follows:

$$MaxPeakNAKs = \max_m(\max_i NAKs(m,i))$$

  where i, m, and $NAKs(m,i)$ are defined as above.
- *Average Recovery Latency.* Recovery Latency is defined as the ratio of the data recovery time observed by a receiver r and the round-trip time from r to the sender. For example, latency of 0.5 means that the time for the receiver to recover the data is half of its round-trip time to the sender. We use the following formula to compute the Average Recovery Latency across all packet losses.

$$AvgRcvrLatency = \frac{\sum_i \sum_r \frac{RecoveryTime(r,i)}{RTT(r)}}{\sum_i Lossy(i)}$$

  where r is a receiver in the multicast tree, $RTT(r)$ is the Round Trip Time from r to the sender, $RecoveryTime(r,i)$ is r's recovery time for the i-th packet loss, and $Lossy(i)$ is the number of receivers that are affected by the i-th packet loss. The value calculated by this formula reflects the expected recovery latency for a receiver to recover a packet loss.

D. A Generalized Framework

Although we chose to focus on the reliable multicast service, our work contains the ingredients of a general framework for studying incremental deployment of general router-assisted services. For example, while our tree-aware deployment strategies are application specific, our network-aware strategies are applicable to many router services. Similarly, while the metrics we propose in Section III-C are specific to reliable multicast, we can define general metrics as follows.

- *Network Overhead.* Applicable to both data and control overhead, this metric can be defined as the number of links a specific packet traverses. Thus, the metric will capture the cost for the network to transmit a particular packet to its destination(s). Examples for reliable multicast that would map to this metric are AvgNormdataOverhead and AvgNormControlOverhead.
- *Node Overhead.* This metric generalizes implosion and exposure and captures the cost incurred at a node in terms of the number of unwanted messages received. Examples for reliable multicast that map to this metric are MaxAvgNAKs and MaxPeakNAKs.
- *Transaction Latency.* As a generalized metric, Transaction Latency refers to the time from when a packet requesting

an action is transmitted until another packet containing the response to the first packet arrives. An example for reliable multicast that maps to this metric is AvgRcvrLatency.

With a generalized framework, application specific deployment strategies still have a place in the study if they are deemed important, or for calibrating the performance of non-application specific strategies. For example, in our reliable multicast study we would still like to investigate tree-aware strategies for the reasons outlined in the Section II-B.

IV. Simulation Results

A. Simulation Assumptions

To reduce the complexity and have more tractable simulations we make the following assumptions.

- *Control or repair packets are not lost.* While in reality all packets may suffer loss, we only consider cases where recovery is successful on the first try. The reason is that we want to focus on overhead due to the various deployment strategies, instead of the protocol's ability to handle multiple losses and optimize the processing.
- *Random packet loss model.* We are not aware of any realistic loss models for multicast traffic. To avoid making our own (possibly flawed) assumptions, we randomly generate the packet loss sequence with uniform packet loss probability on each link in the multicast tree.

B. Simulation Setup

To evaluate the impact of different deployment strategies we ran numerous simulations on a mapped router-level Internet topology [13], [15]. The topology has 27,646 routers, 134,620 links, and 1,155 ASs. Table I summarizes some statistics of this mapped Internet topology.

TABLE I
STATISTICS OF THE MAPPED INTERNET TOPOLOGY

Property	Average	Std	Min	Max	Median
Router Fanout	4.87	6.06	2	91	3
AS Fanout	5.79	20.46	1	362	2
AS Size	23.93	134.58	1	3269	4

In our simulation we assume all paths are symmetric and each link has the same delay and cost per transmission. For simplicity, we only consider static multicast groups with the root of multicast tree as the single source of normal data traffic. In this section we present the simulation results with the sender and receivers being randomly attached to routers in the network, and with the group size being 5% of the network size. In Section V, we present results with different group sizes and different receiver and sender placement models.

For each set of parameters we average the results over 50 different multicast trees, each generated with a different randomization seed for sender and receiver placement. In addition, for each Uniform deployment strategy, we repeat the simulation 10 times for each multicast tree with a different seed for selecting routers for deployment. We present the results averaged across all simulations, along with the 95th percentile confidence interval. The Y-axis shows the metric being measured, and the X-axis shows the percentage of deployed routers *in the multicast tree*. For example, 50% deployment means that half the routers in the multicast tree are deployed. Another candidate for the X-axis is the deployment percentage *in the network*, *i.e.*, the percentage of deployed routers in the network. In Section V, we present the mapping between the two deployment percentages which can be used to infer the performance in terms of deployment percentage *in the network*.

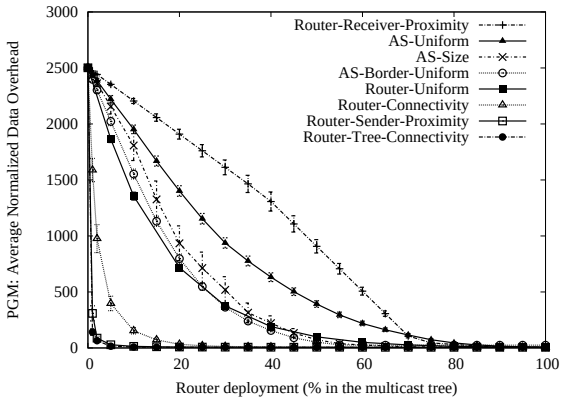

Fig. 4. PGM Average Normalized Data Overhead

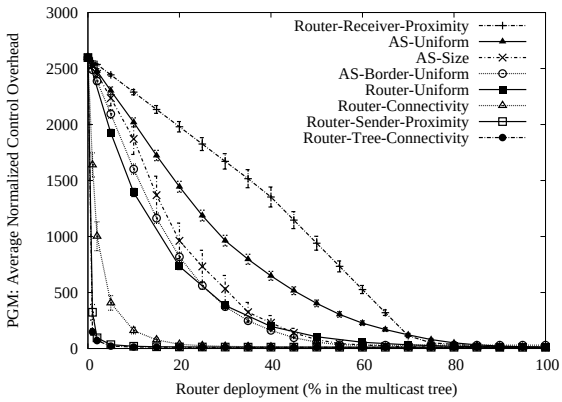

Fig. 5. PGM Average Normalized Control Overhead

The simulation is carried out via an event-driven simulator that numerically simulates packet-level actions by all nodes in the multicast tree. A packet loss sequence is first generated and fed to the simulator. The simulator then processes each packet loss sequentially. Since LMS does not have NAK back-off mechanism and the processing of a packet loss does not depend on the processing of previous packet losses, we evaluate its performance using a short packet loss sequence where each link in the multicast tree appears just once. For PGM, the processing of a packet loss depends on the processing of previous packet losses due to its dynamic adjustment of NAK back-off interval (see Section III-A). Hence, we evaluate its performance using a relatively long packet loss sequence where each link in the multicast tree randomly appears 20 times.

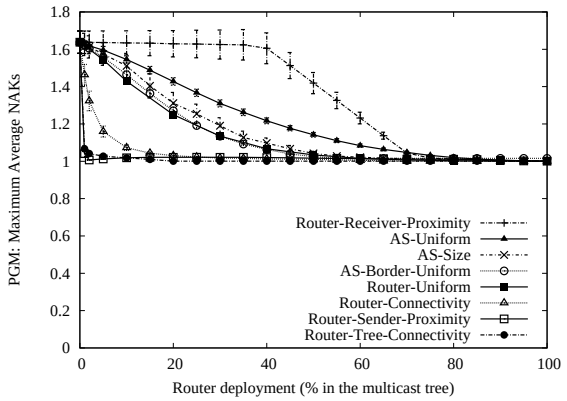

Fig. 6. PGM Maximum Average NAKs

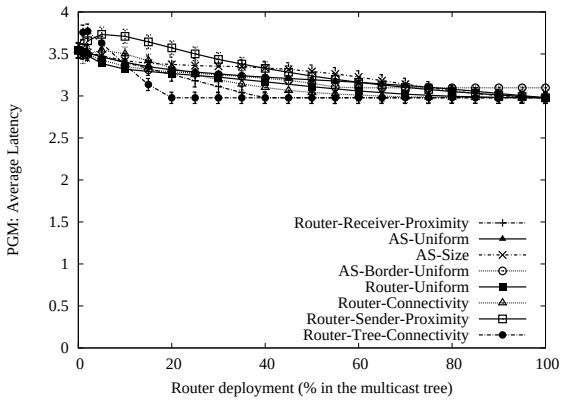

Fig. 8. PGM Average Recovery Latency

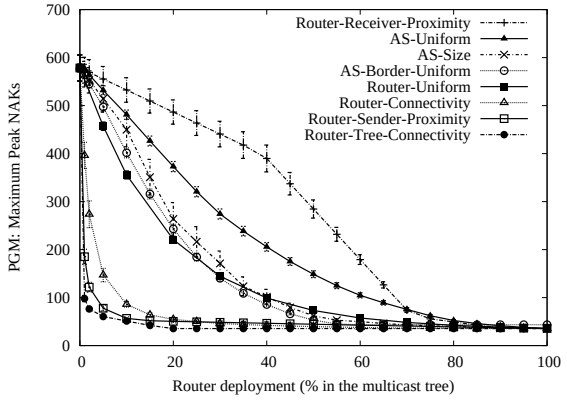

Fig. 7. PGM Maximum Peak NAKs

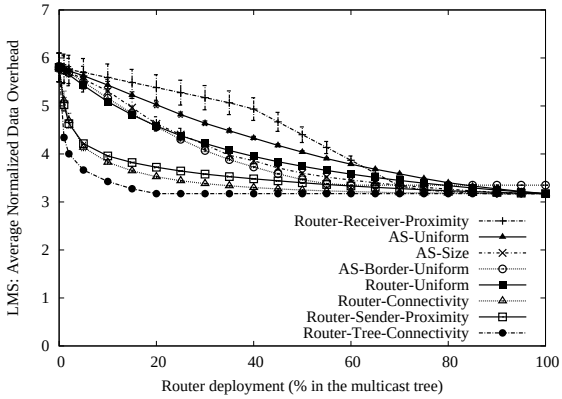

Fig. 9. LMS Average Normalized Data Overhead

In the simulation for LMS, each LMS router selects the nearest downstream receiver (in terms of hop distance) as its replier. For PGM, we set the upper bound for the NAK back-off interval to 128 $\times$ maxRTT (where maxRTT is the maximum RTT from the sender to any receiver), the lower bound is set to one maxRTT, and the number of rounds needed to reduce the NAK back-off interval (N) is set to five. In addition, the sender holding time is set to one maxRTT, which is the maximum value PGM recommends. We investigate the impact of these parameters in Section V.

C. Simulation Results for PGM

Figure 4 shows the Average Normalized Data overhead for PGM with the eight deployment strategies described in Section II. The high data overhead at zero deployment is due to the multicast of repair packets. In the absence of PGM routers to guide them to appropriate receivers, any repair packet from the sender will flood the entire multicast tree, even when there is only one receiver experiencing the packet loss.

The results in Figure 4 indicate that the eight strategies can be roughly divided into three categories. The first category contains the best performers, Router-Tree-Connectivity, Router-Sender-Proximity, and Router-Connectivity strategies. The second category includes Router-Uniform, AS-Uniform, AS-Border-Uniform, and AS-Size strategies. The third category contains the worst performer, Router-Receiver-Proximity strategy.

In the first category, both Router-Tree-Connectivity strategy nd Router-Sender-Proximity strategy are exceptional performers, achieving near full-deployment performance with only 5% of the routers deployed. Router-Connectivity strategy achieves similar performance with about 20% of the routers deployed.

Intuitively, the overall good performance of these three strategies can be explained as follows. First, deploying PGM on routers with large tree fanout can achieve better targeting of repair packets, significantly reducing the transmission on unwanted links. Second, Router-Sender-Proximity strategy performs exceptionally well because a router near the sender is more likely to have a large tree fanout and large size subtrees, which play an important role in the targeting of repair packets. Finally, Router-Connectivity strategy also performs very well, because a router that has more neighbors in the network is more likely to have a large fanout in the multicast tree.

In the second category, we can see that both Router-Uniform strategy and AS-Border-Uniform strategy perform better than AS-Uniform strategy. We note that in general, strategies deploying on the router granularity and border router granularity perform better than their counterparts that deploy on the AS granularity. A possible explanation is that deploying on the AS granularity will deploy many adjacent routers in the multicast tree, and the marginal benefit of deploying a router adjacent to routers that have been deployed may not

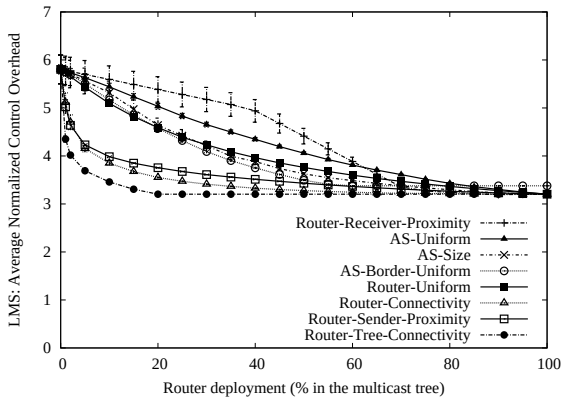

Fig. 10. LMS Average Normalized Control Overhead

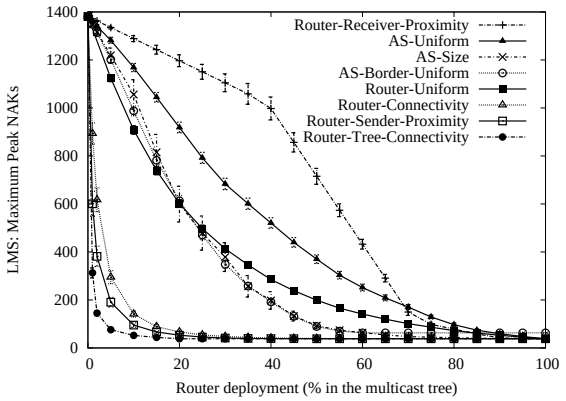

Fig. 12. LMS Maximum Peak NAKs

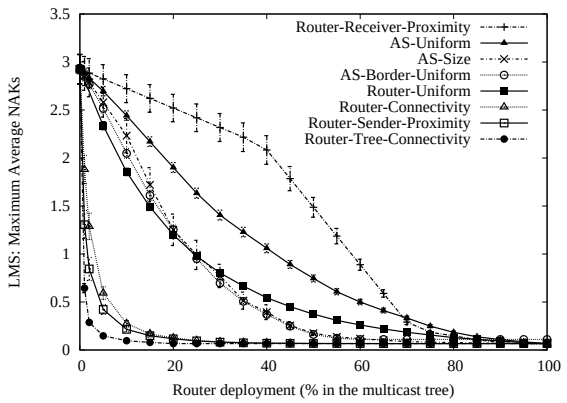

Fig. 11. LMS Maximum Average NAKs

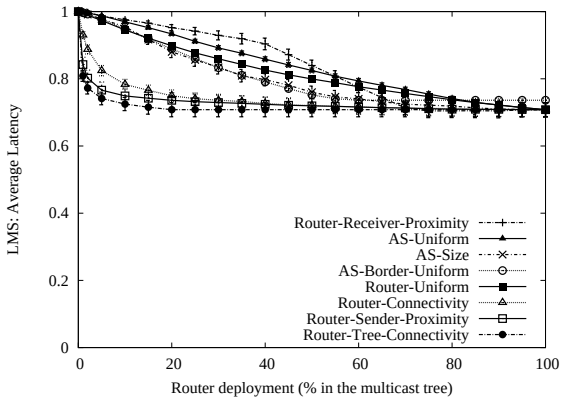

Fig. 13. LMS Average Recovery Latency

be as good as deploying a router that is a few hops away from any deployed router because the latter controls another region in the tree. The figure also shows that the AS-Size deployment strategy performs slightly worse than the Router-Uniform strategy when deployment level is less than 45%, but it is still better than the AS-Uniform strategy.

The third category contains the worst performer, the Receiver-Proximity strategy. This confirms our intuition that routers close to receivers generally play a less significant role since they are unlikely to have a large number of downstream receivers.

Figure 5 shows the normalized control overhead for PGM. The results are virtually identical with the normalized data overhead. This is not surprising since the majority of the control overhead in sparse deployment comes from the NCFs, which are transmitted in a similar fashion as the repair packet.

Figures 6 and 7 show the Maximum Average NAKs and Maximum Peak NAKs in PGM respectively. Maximum Average NAKs are very low, showing that PGM's NAK suppression mechanism works very well, on the average. The Maximum Peak NAKs, however, reveal that NAK storms are possible at low deployment, and the eight deployment strategies again divide themselves into the same three groups.

Figure 8 shows the average recovery latency in PGM. In addition to propagation delay, the other main contributors to latency are the NAK back-off interval and the sender holding time. Unlike exposure and implosion, the three-group classification is not visible here. The range of recovery latency is less than one RTT (between 2.97 and 3.77), so deployment seems to have a small effect.

D. Simulation Results for LMS

Figure 9 and Figure 10 show the Average Normalized Data Overhead and Control Overhead for LMS. Compared with PGM, both types of overhead start with sharply lower values in LMS. This is not surprising, because at zero deployment, all control packets and repair packets in LMS will be transmitted through unicast between the sender and the affected receivers, while in PGM all repair packets and most control packets (namely, NCFs) will flood the entire multicast tree via multicast.

Interestingly, the results for eight deployment strategies are grouped in the same three categories as observed with the PGM results. The only notable difference is that Router-Connectivity and Router-Sender-Proximity have swapped places. The reason is that in LMS repairs are typically sent by repliers, not by the sender; therefore routers near the sender become less important. The trends for the second and third categories are similar to PGM.

Figure 11 shows the maximum average NAKs in LMS. While initially higher than PGM, this overhead also appears to be negligible. Figure 12 shows the Maximum Peak NAKs

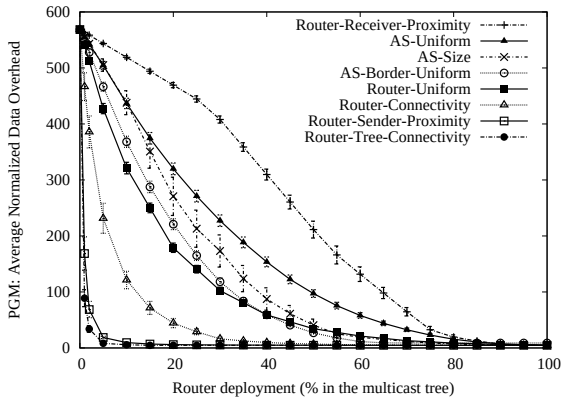

Fig. 14. Receiver population sensitivity: PGM Average Normalized Data Overhead with 1% multicast group size

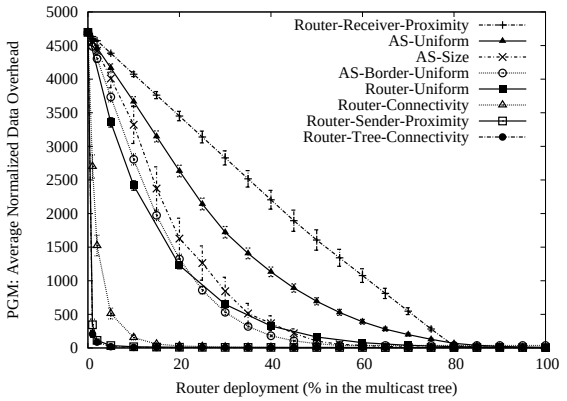

Fig. 16. Receiver population sensitivity: PGM Average Normalized Data Overhead with 10% multicast group size

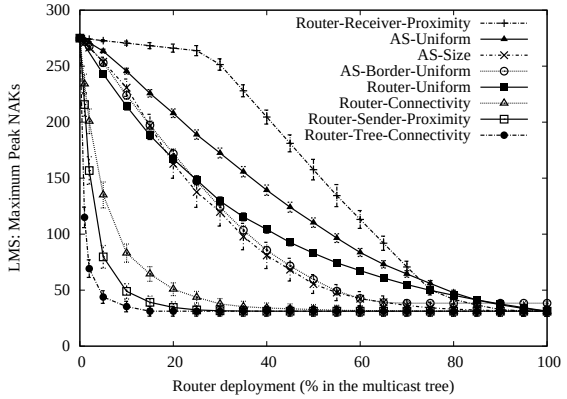

Fig. 15. Receiver population sensitivity: LMS Maximum Peak NAKs with 1% multicast group size

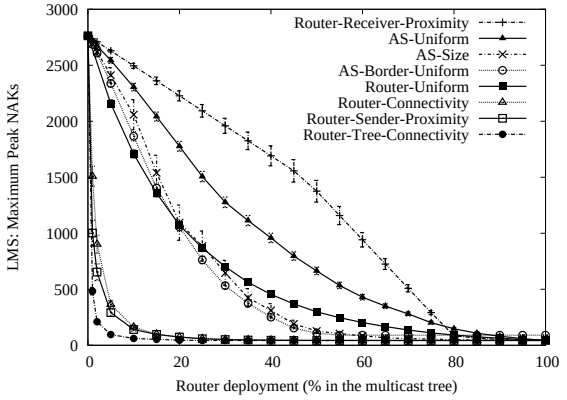

Fig. 17. Receiver population sensitivity: LMS Maximum Peak NAKs with 10% multicast group size

for LMS. The figure reveals the most significant cost of incremental deployment of LMS. Since LMS does not have a suppression mechanism like PGM, at zero deployment all NAKs go to the sender, resulting in NAK implosion at the sender. As deployment increases, the three best deployment schemes quickly reduce this overhead, and by 20% deployment a node receives about the same number of NAKs as with full deployment.

Figure 13 shows the Average Recovery Latency for LMS. There are no surprises here. The differentiation between strategies seems more pronounced than PGM, but all strategies in general offer recovery times of under one RTT.

In summary, the deployment strategies we study exhibit similar trends for both PGM and LMS, except perhaps for the mild differences in recovery latency. The impact of incremental deployment, however, appears very different between the two protocols. In PGM the impact is felt in terms of data and control overhead because at low deployment NCFs and repairs cannot be targeted well, and thus reach a larger part of the multicast tree. In LMS the impact comes in the form of NAK storms pounding individual endpoints, most notably the sender. Thus, we observe that with PGM, incremental deployment impacts the *network*, where with LMS it impacts the *endpoints*. Recovery latency is slightly higher in PGM due to the additional timers and timer management mechanisms.

V. Sensitivity

In this section we explore the sensitivity of our simulation results to factors such as multicast group size, receiver placement, sender placement, and the impact of the back-off timer interval in PGM. Due to space limitation, we focus on the Average Normalized Data Overhead for PGM and the Maximum Peak NAKs for LMS, since these two metrics are affected the most by partial deployment for PGM and LMS, respectively.

A. Multicast Group Size

Figure 14 and Figure 16 show results for PGM when the multicast group size is 1% and 10% of the network size, respectively. The results for LMS are in Figure 15 and Figure 17. The receivers and the sender are again chosen at random. We make the following observations: (a) for both protocols the zero-deployment overhead is, as expected, proportional to the group size; (b) for both protocols, increasing group size seems to enlarge the gaps between the three strategy categories; and, (c) Router-Receiver-Proximity strategy seems to be impacted more by group size than other strategies. In general, however, the overall behavior seen earlier appears to persist.

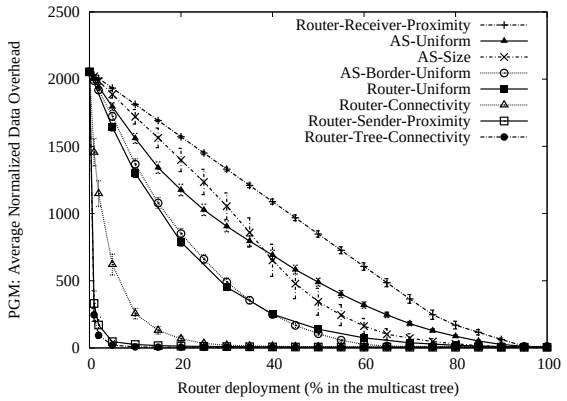

Fig. 18. Receiver placement sensitivity: PGM Average Normalized Data Overhead with the extreme Affinity receiver placement

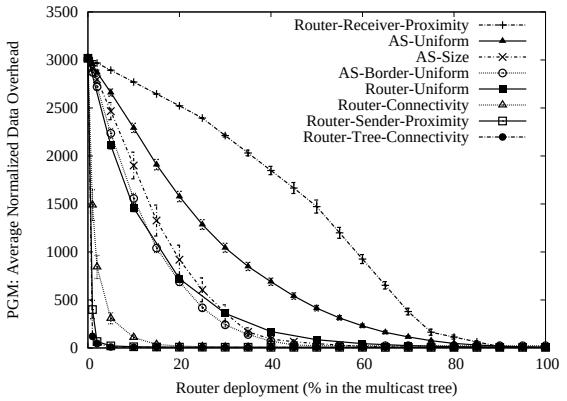

Fig. 20. Receiver placement sensitivity: PGM Average Normalized Data Overhead with the extreme Disaffinity receiver placement

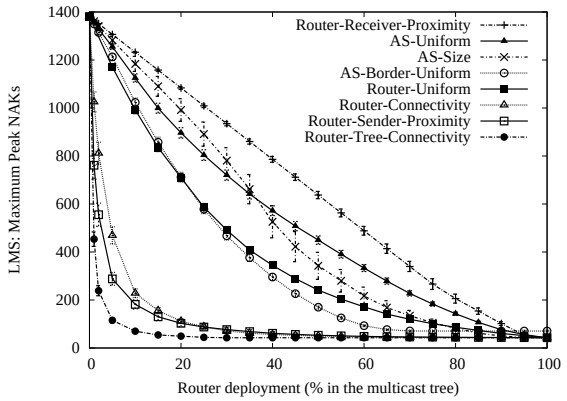

Fig. 19. Receiver placement sensitivity: LMS Maximum Peak NAKs with the extreme Affinity receiver placement

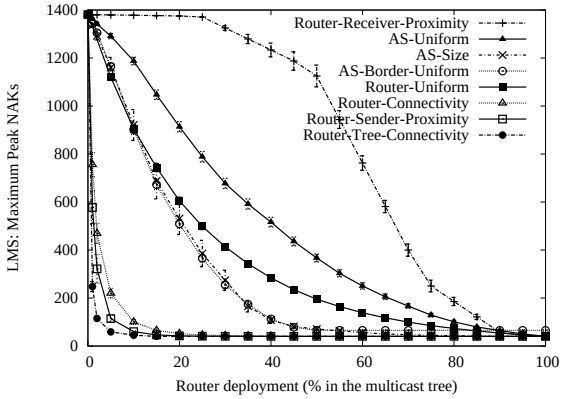

Fig. 21. Receiver placement sensitivity: LMS Maximum Peak NAKs with the extreme Disaffinity receiver placement

B. Receiver Placement

To study the impact of receiver placement, we consider two extreme receiver placement models as defined in [16], namely *extreme affinity* and *extreme disaffinity*. The extreme affinity model places receivers as close to each other as possible, while the extreme disaffinity model places receivers as far away from each other as possible. The particular algorithm for receiver selection we use is given in [17] and is summarized below. We first randomly select one node among all nodes. Then, we assign to each node n_i that is not selected yet the probability $p_i = \frac{\alpha}{w_i^\beta}$, where w_i is the closest distance between node n_i and a node that is already selected, α is calculated such that $\sum_{n_i} p_i = 1$, and β is the parameter that defines the degree of affinity and disaffinity. The probability is recomputed after a new node is selected. Similar to [17], we use $\beta = 15$ and $\beta = -15$ for extreme affinity and disaffinity respectively.

Figure 18 and Figure 19 show the results for extreme affinity for each protocol. Figure 20 and Figure 21 show the results for extreme disaffinity. In all these settings, the multicast group size is 5%, and the sender is placed in the network at random. From these results we can see that the receiver placement algorithm has more significant impact on the Router-Receiver-Proximity and AS-Size strategies than on others. In the case of extreme disaffinity placement, the Router-Receiver-Proximity strategy performs far worse than others, while in the case of extreme affinity that strategy is closer to the second worst strategy. The reason is that in the extreme affinity model receivers are placed close to each other and as a result, there are fewer routers in the multicast tree. Routers close to the receivers have higher fanout, which gives them more control over the data recovery. In contrast, the AS-Size strategy appears to be worse in the extreme affinity model compared to the extreme disaffinity model. A possible explanation is that in the extreme affinity model a large portion of receivers may be clustered in small ASs, in which case deploying the largest AS offers small benefit.

C. Sender Placement

In the previous experiments, the sender location is uniformly selected among all nodes on the topology. The sender location, however, plays a significant role for the AS-Size deployment. Figure 22 and Figure 23 show the results with the sender being placed in the largest AS. We see that the AS-Size strategy performs much better compared with the scenario where the sender is uniformly selected among all network nodes. The reason for the improvement is that by deploying routers in the largest AS at the beginning, we hit routers close to the sender, providing similar benefits to the Router-Sender-Proximity deployment strategy.

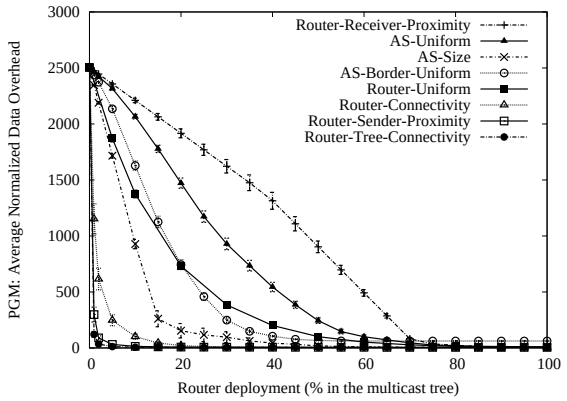

Fig. 22. Receiver placement sensitivity: PGM Average Normalized Data Overhead with the Largest-AS sender placement

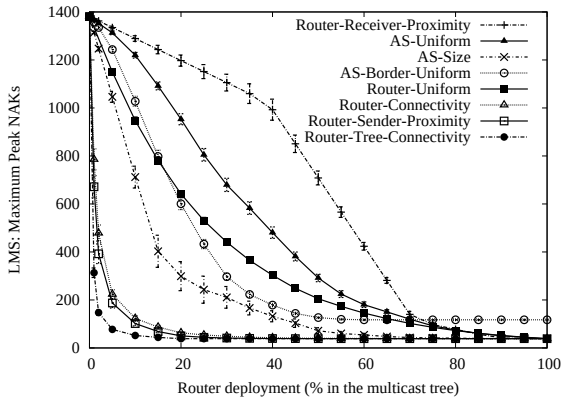

Fig. 23. Receiver placement sensitivity: LMS Maximum Peak NAKs with the Largest-AS sender placement

D. Deployment Percentage on the Network

For all previous results, we investigated the performance of different deployment strategies in terms of deployment percentage *in the multicast tree*. Another perspective is to use the percentage of deployed routers *in the network* to represent the deployment cost (the X-axis). In this subsection, we present the relationship between these two router deployment percentages for each deployment strategy. With this relationship, we can infer the performance in terms of deployment percentage in the network from the performance in terms of deployment percentage in the multicast tree.

We obtain the relationship between these two deployment percentages by recording the mapping between their values at all deployment instances and averaging it across all simulations with the same setting. Figure 24 shows the average mapping along with the 95th percentile confidence interval for the setting with 5% group size and random receiver and sender placement. The X-axis is the deployment percentage in the network, and the Y-axis is the deployment percentage in the multicast tree.

The graph shows that for all Tree-Aware strategies the deployment percentage in the multicast tree is about 7.85 times the deployment percentage in the network on average. For example, deploying 0.64% routers on the network according to Router-Tree-Connectivity strategy is equivalent to deploy

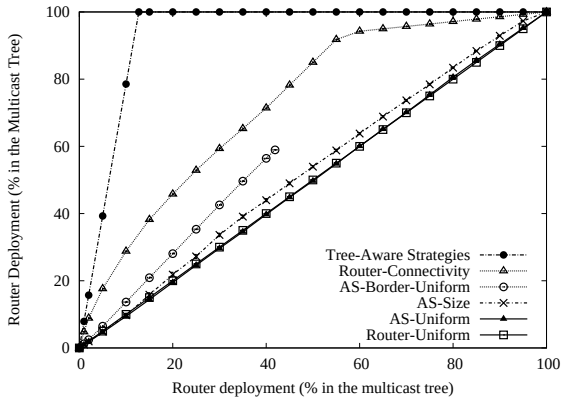

Fig. 24. Mapping between deployment percentage in the network and deployment percentage in multicast tree with 5% group size

5% routers in the multicast tree, which is almost as good as full deployment for both PGM and LMS.

This mapping can be explained as follows. For all deployment strategies, we have the following equation.

$$\frac{PercentageInTree}{PercentageInNetwork} = \frac{Prob(R\ in\ tree \mid R\ is\ deployed)}{Prob(R\ in\ tree)}$$

where R is a router in the network, $Prob(R\ in\ tree)$ is its probability being in the multicast tree, and $Prob(R\ in\ tree \mid R\ is\ deployed)$ is its probability being in the multicast tree given that R has been deployed.

Since all Multicast-Tree-Aware strategies only deploy routers in the multicast tree, $Prob(R\ in\ tree \mid R\ is\ deployed)$ is always 1. $Prob(R\ in\ tree)$ is equal to the ratio of the average multicast tree size and the network size (both in number of routers). For the 50 multicast trees used in the simulation, average multicast tree size is about 1/7.85 of the network size. So the ratio of the two deployment percentages would be 7.85.

For Router-Connectivity strategy, the graph shows the deployment percentage in the network is mapped to a much higher deployment percentage in the multicast tree at the beginning. For example, a 5% deployment in the network is mapped to nearly 18% deployment in the multicast tree. This is very encouraging as it means deploying 5% routers in the network according to Router-Connectivity strategy can yield the same benefit as deploying 18% routers in the multicast tree, which approaches full deployment performance. This reason behind this mapping is because Router-Connectivity strategy deploys routers with the largest fanout in the network first, and such routers are more likely to be included in the multicast tree than average routers. So for Router-Connectivity strategy, a deployment percentage in the network corresponds to a much higher deployment percentage in the multicast tree initially, and as more and more routers get deployed, these two percentages will converge at the full deployment.

In the graph, we see that AS-Border-Uniform strategy also has a relatively steep slope, suggesting that AS border routers are more likely to be in the multicast tree than average routers. So the performance of AS-Border-Uniform strategy in terms of deployment percentage in the network will also be better than its performance in terms of deployment percentage in the

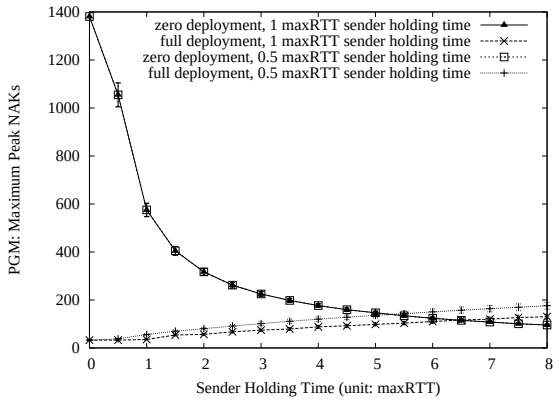

Fig. 25. PGM back-off interval sensitivity: impact of the lower bound for the NAK back-off interval on Maximum Peak NAKs

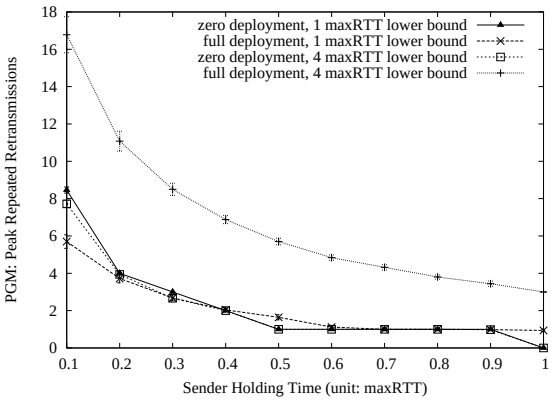

Fig. 27. PGM back-off interval sensitivity: impact of the sender holding time on Peak Repeated Retransmissions

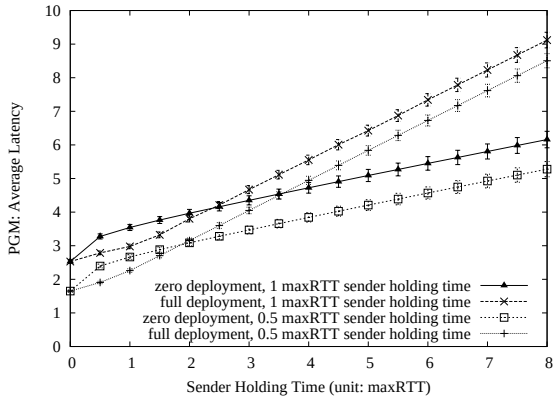

Fig. 26. PGM back-off interval sensitivity: impact of the lower bound for the NAK back-off interval on Recovery Latency

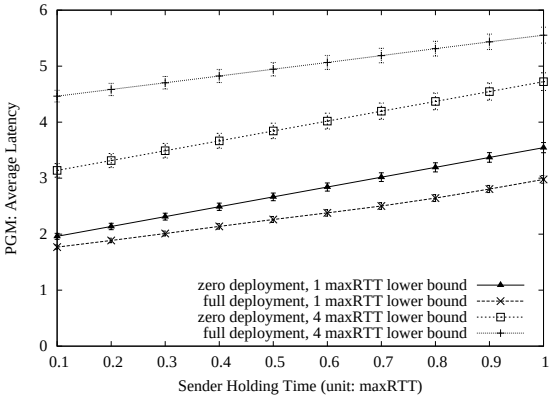

Fig. 28. PGM back-off interval sensitivity: impact of the sender holding time on Recovery Latency

multicast tree. Note that the mapping stops at 41.9% in the network when all border routers in the network are deployed. For both AS-Uniform strategy and Router-Uniform strategy, the deployment percentage in the network is almost identical to the deployment percentage in the multicast tree, while for AS-Size strategy, deployment percentage in the network is mapped to a slightly higher deployment percentage in the multicast tree.

Besides the mapping for the setting with 5% group size and random receiver and sender placement, we also obtained the mapping under different group size and different receiver and sender placement. Although there is some variation with different setting, the trend is similar with Multicast-Tree-Aware strategies having the steepest initial slope, followed by Router-Connectivity strategy and AS-Border-Uniform strategy. The other three strategies, AS-Size, AS-Uniform, and Router-Uniform, keep almost identical mapping between the two deployment percentages.

E. PGM NAK Back-off Interval

The NAK back-off interval plays an important role in the performance of PGM because it controls the trade-off between NAK suppression and recovery latency. Recall that there are three main parameters in the dynamic adjustment of the NAK back-off interval: the upper bound, the lower bound, and the number of measurements for reducing NAK back-off interval. Our simulation results show that among the three parameters, the lower bound plays a substantially more important role in the performance of PGM.

Figure 25 shows the impact of the lower bound on the Maximum Peak NAKs in PGM. Note that the two graphs at zero deployment coincide. At zero deployment increasing the lower bound is very beneficial, because the maximum peak NAKs drop quickly, virtually unaffected by the sender's holding time.

At full deployment, however, we observe the opposite effect. The maximum peak NAKs increase steadily as we increase the lower bound, and the values with a shorter sender holding time are higher than corresponding values with a longer sender holding time. While this may be counter-intuitive since one might expect that larger back-off intervals would suppress more NAKs, at full deployment each receiver has to send out a NAK in order to get a repair packet, because all routers are now PGM capable and will only forward the repair packet on the interface where a NAK was previously received. Therefore, the larger the NAK back-off interval, the more likely that a NAK will be originated after a repair packet has been sent out by the sender, which will wipe out the repair state among PGM routers. This results in significant increase in the number of repeated retransmissions at full deployment. It also means that

more NAKs will reach the sender. Obviously, the shorter the sender holding time, the more likely that a repair packet will be sent out prematurely, leading more repeated retransmissions and more NAKs on the sender.

Figure 26 shows the impact of increasing the lower bound of the NAK back-off interval on recovery latency. The impact is stronger under full deployment, but luckily this is when NAK suppression is needed the least. Thus, we conclude that the lower bound of the NAK back-off interval should be kept relatively high (between three to five maxRTT according to our results) at low deployment levels, and reduced as deployment level increases.

F. PGM Sender Holding Time

Recall that a repeated retransmission occurs when the sender sends a repair packet before the NAK state in the routers was fully established. We can reduce the number of repeated retransmissions by increasing the sender holding time. Figure 27 shows this effect under different deployment levels and values of the lower bound for the NAK back-off interval. We can see that regardless of the deployment level and the lower bound of the NAK back-off interval, the number of peak repeated retransmissions decreases when the sender holding time increases. However, increasing the sender holding time also increases the average recovery latency, which is shown in Figure 28. Therefore, selecting a proper sender holding time is a trade-off between average recovery latency and repeated retransmissions.

VI. Related Work

In recent years, there have a number of proposals for new network services that put additional functions to network routers. For example, PGM [1] and LMS [2] are two protocols that rely on router assistance to achieve reliable multicast. GIA [3] introduces a scalable architecture for routers to support global IP-anycast service. Another router-assisted new network service, Concast (many-to-one communication) has been studied in [4], [5] Two network layers primitives, Packet Reflection and Path Painting, have been proposed in [9] to support overlay networks. [10] presents a generic router-based building block that allows packets to create and manipulate small amounts of temporary state at routers via short, predefined computations. In security area, SAVE [6] designs a protocol for routers to validate the source address of IP packets.

While most proposals have sketched mechanisms to make the protocol work with partial deployment of router support, the subject of incremental deployment strategies has typically received little attention. Few has carried out systematic evaluation of the protocol's utility under various incremental deployment strategies. In addition to the earlier work on incremental deployment of LMS [12] described in Section I, there are three other studies that have investigated the impact of partial deployment on the performance of new router-assisted services. The first is in the context of Active Reliable Multicast(ARM) [18]. ARM argues that significant benefits can be obtained even when only 50% of the routers are ARM-capable. Further, the authors suggest that significant benefits can be obtained even with a much smaller set of ARM-capable routers if strategically located, but they do not investigate what these strategies are. The second study [8] proposes the use of router-based distributed packet filtering to counter DDos attacks. The study presents a deployment strategy that decreases the number of spoofable addresses while minimizing the percentage of routers performing the filtering. The third one [5] compares the performance of Concast under three scenarios, no router support, full router support, and support at egress routers. Simulation results suggest deploying all egress routers can yield significant benefits.

Previous work on reliable multicast can be divided into two broad categories: (a) *end-to-end*, and (b) *router-assist* schemes. End-to-end schemes do not depend on router support, therefore they are much easier to deploy. Those schemes include RMTP [19], TMTP [20], SRM [21], TRAM [22], to name a few.

In addition to PGM [1] and LMS [2], which are the subjects of this paper, other router-assist schemes include the following: Search Party [23] is inspired by LMS, and adds robustness by using *randomcast* to distribute a request randomly among receivers rather than just a single replier. Addressable Internet Multicast (AIM) [24] assigns labels to routers on a per-multicast group basis, and routes requests and repairs based on these labels. OTERS [25] and Tracer [26] both employ mtrace utility to build congruent hierarchies. Finally, Active Error Recovery (AER) [27] is targeted towards an active networks environment. For further references on previous work on both end-to-end and router-assist reliable multicast see [2].

VII. Conclusions and Future Work

Adopting a new service in the Internet is difficult without a viable incremental deployment plan. If a service does not follow the appropriate deployment strategy, an otherwise robust service may fail. Unfortunately, little work has been done to systematically study deployment strategies and their impact on performance.

In this paper, we defined a methodology for evaluating incremental deployment of router-assist reliable multicast. Given the lack of information about how deployment occurs in the real world, our study adopted a blend of plausible and canonical deployment strategies and a mix of performance metrics that capture both implosion and exposure. In our study we considered two protocols, namely PGM and LMS, and used numerical simulation to evaluate their performance under partial deployment. We investigated a variety of deployment strategies over a large real-world router-level Internet topology. Such study is needed not only to determine which deployment strategy is better, but also to investigate the level of deployment necessary to reach acceptable performance. In addition, we carried out a sensitivity analysis to determine the impact of factors such as multicast group size, receiver and sender placement, and the selection of the PGM back-off timer interval.

Clearly, our study ignores many real-life factors that affect router deployment. Such factors include upgrade schedules for

both hardware and software, economic considerations, peering relationships, router and network capacities, user demand, and others. While these are important considerations, modeling them is hard, especially since they vary across ISPs. Thus, we focus on investigating simple deployment strategies that are, however, easy to understand.

Our results show significant difference among various deployment strategies, suggesting that our methodology is capable of capturing and characterizing their performance and clearly demonstrating that careful study of incremental deployment can not be ignored. Our study also identifies different types of overhead during deployment, namely *network* and *end-point* overhead, and demonstrates that it is important to provide metrics and methodology to capture both.

Results from our case study are very encouraging. Some strategies are clear winners, requiring only a small percentage of the routers to be deployed for near-optimal performance. The impact of deployment strategies varies significantly, with the best allowing both protocols to approach full-deployment performance with as little as 5% of the routers deployed, and others needing upwards of 80% deployment to reach the same level of performance. Clearly, deployment strategies do have a strong impact on these protocols. Thus, our study has produced useful information for network planners contemplating the deployment of such services.

As future work we plan to extend our work to a framework for more generic router-assisted network services beyond reliable multicast, such as Anycast, source address validation ([6]), network filtering against DDoS attacks [7], and other end-to-end services([9], [10]). Although each individual service may have its unique requirement for incremental deployment and evaluation metrics, we believe our current work can be easily applied to study the incremental deployment strategies for other services. In addition, we are looking at possible improvements to PGM and LMS protocols to improve performance under partial deployment, as our study reveals the significant penalty associated with each protocol under some deployment strategies.

ACKNOWLEDGMENTS

This paper has benefited greatly from the valuable feedback from Dr. Ramesh Govindan.

REFERENCES

[1] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. L. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano, "PGM Reliable Transport Protocol Specification," *Request For Comments (RFC) 3208*, December 2001, http://www.ietf.org/rfc/rfc3208.txt?number=3208.

[2] C. Papadopoulos, G. Parulkar, and G. Varghese, "An Error Control Scheme for Large-Scale Multicast Applications," in *Proceedings of the IEEE Infocom'98*, San Francisco, USA, March 1998, pp. 1188–1196.

[3] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)," in *Proceedings of the ACM SIGCOMM'2000*, Stockholm, Sweden, August 2000.

[4] K. L. Calvert, J. Griffioen, A. Sehgal, and S. Wen, "Concast: Design and Implementation of a New Network Service," in *Proceedings of the 7th IEEE International Conference on Network Protocols (ICNP'99)*, October 1999.

[5] K. L. Calvert, J. Griffioen, B. Mullins, A. Sehgal, and S. Wen, "Concast: Design and Implementation of an Active Network Service," *IEEE Journal of Selected Areas in Communications*, 2001.

[6] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "SAVE: Source Address Validity Enforcement Protocol," in *Proceedings of the IEEE Infocom 2002*, New York, NY, USA, June 2002.

[7] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Framework for Classifying Denial of Service Attacks," in *Proceedings of the ACM SIGCOMM'2003*, Karlsruhe, Germany, August 2003.

[8] K. Park and H. Lee, "On the Effectiveness of RouteBased Packet Filtering for Distributed DoS Attack Prevention in PowerLaw Internets," in *Proceedings of the ACM SIGCOMM'2001*, San Diego, California, USA, August 2001.

[9] J. JANNOTTI, "Network layer support for overlay networks," June 2002.

[10] K. L. Calvert, J. Griffioen, and S. Wen, "Lightweight network support for scalable end-to-end services," in *Proceedings of the ACM SIGCOMM'2002*, Pittsburgh, PA, August 2002.

[11] P. Radoslavov, C. Papadopoulos, R. Govindan, and D. Estrin, "A Comparison of Application-Level and Router-Assisted Hierarchical Schemes for Reliable Multicast," in *Proceedings of the IEEE Infocom 2001*, Anchorage, Alaska, USA, April 2001.

[12] C. Papadopoulos and E. Laliotis, "Incremental Deployment of a Router-assisted Reliable Multicast Scheme," in *Proceedings of Networked Group Communications (NGC2000)*, Stanford University, Palo Alto, CA, USA, November 2000.

[13] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *Proceedings of the IEEE Infocom 2000*, Tel-Aviv, Israel, March 2000.

[14] K. L. Calvert, M. B. Doar, and E. W. Zegura, "Modeling Internet Topology," *IEEE Communications Magazine*, June 1997.

[15] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-Informed Internet Replica Placement ," in *Proceedings of the Sixth International Workshop on Web Caching and Content Distribution*, Boston, Massachusetts, USA, June 2001.

[16] G. Phillips, S. Shenker, and H. Tangmunarunkit, "Scaling of Multicast Trees: Comments on the Chuang-Sirbu scaling law," in *Proceedings of the ACM SIGCOMM'99*, Cambridge, Massachusetts, USA, August 1999.

[17] T. Wong and R. Katz, "An Analysis of Multicast Forwarding State Scalability," in *Proceedings of the 8th IEEE International Conference on Network Protocols (ICNP 2000)*, Osaka, Japan, November 2000.

[18] L. Lehman, S. J. Garland, and D. L. Tennenhouse, "Active Reliable Multicast," in *Proceedings of the IEEE Infocom'98*, San Francisco, USA, March 1998.

[19] J. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," in *Proceedings of the IEEE Infocom'96*, San Francisco, USA, March 1996, pp. 1414–1424.

[20] R. Yavatkar, J. Griffoen, and M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications," in *Proceedings of the Third International Conference on Multimedia '95*, San Francisco, CA, USA, November 1995.

[21] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, November 1997.

[22] D. Chiu, S. Hurst, M. Kadansky, and J. Wesley, "TRAM: A Tree-based Reliable Multicast Protocol," Sun Microsystems, Tech. Rep. Sun Technical Report SML TR-98-66, July 1998.

[23] A. M. Costello and S. McCanne, "Search Party: Using Randomcast for Reliable Multicast with Local Recovery," in *Proceedings of IEEE Infocom'99*, New York, USA, March 1999.

[24] B. Levine and J. J. Garcia-Luna-Aceves, "Improving Internet Multicast with Routing Labels," in *Proceedings of the 5th IEEE International Conference on Network Protocols (ICNP'97)*, Atlanta, GA, USA, October 1997.

[25] D. Li and D. R. Cheriton, "OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol," in *Proceedings of the 6th IEEE International Conference on Network Protocols (ICNP'98)*, October 1998, pp. 237–245.

[26] B. N. Levine, S. Paul, and J. J. Garcia-Luna-Aceves, "Organizing Multicast Receivers Deterministically According to Packet-Loss Correlation," in *Proceedings of the 6th ACM International Conference on Multimedia*, September 1998, pp. 201–210.

[27] S. K. Kasera, S. Bhattacharyya, M. Keaton, D. Kiwior, J. Kurose, D. Towsley, and S. Zabele, "Scalable Fair Reliable Multicast Using Active Services," *IEEE Network Magazine (Special Issue on Multicast)*, January/February 2000.