A Comparison of Incremental Deployment Strategies for Router-Assisted Reliable Multicast

Xinming He, Christos Papadopoulos, Pavlin Radoslavov, Ramesh Govindan

Abstract—**Several reliable multicast schemes using router assistance have been proposed recently, which not only promise performance gains, but also simplify applications. Since they require network assistance, the effectiveness of such schemes degrades if the deployment of the required network services is sparse, meaning that an effective deployment strategy is critical.**

In this paper we study the performance of two such schemes, namely PGM and LMS, under six different incremental deployment strategies, including the Fanout-in-the-Multicast-Tree, Distance-from-the-Sender, Fanout-on-the-Network, AS, AS-Border-Router, and Random deployments. Based on the simulation results on some real-world topologies, including a router-level Internet topology of 27,646 nodes, we find that by employing the right deployment strategies, we can get significant gains for both PGM and LMS even under sparse deployment.

Keywords— **reliable multicast, router assistance, incremental deployment**

I. Introduction

The problem of reliable multicast has been studied extensively during the past 10 years [1], [2], [3], [4], [5], [6], [7], [8]. Reliable multicast is essential to applications like distributed computing, software updates, distributed caching, etc., and is considered a crucial element in the evolution of multicast.

Recently, several reliable multicast schemes employing router assistance have been proposed [2], [9], [10], [7], [11], [6], [12]. Router-assistance provides two advantages over end-to-end schemes: (a) it helps to efficiently build a hierarchy congruent with the underlying multicast routing tree, and (b) it provides controls for fine-grain multicast of retransmissions.

Recognizing the potential benefits of router assistance, efforts are underway at IETF to standardize router-assist services for reliable multicast. The Generic Router Assist (GRA) [13] architecture defines a limited set of pre-configured router services that allow applications to take advantage of information distributed across the network. GRA defines services that use such information to assist reliable multicast with operations like aggregating Naks and targeting the delivery of retransmissions to receivers that requested them.

Any new network service, including router-assist services, must be able to be deployed in an incremental fashion on the Internet, due to the scale and inherent heterogeneity of the Internet. Two of the router-assist schemes, namely PGM [9] and LMS [2], have stepped up to the challenge and added incremental deployment methods to their specification. Although both are router assisted reliable multicast schemes, PGM and LMS differ significantly in their operations. For example, PGM does Nak aggregation and retransmission targeting at the routers, where LMS defers these actions to the repliers with minimal assistance from the routers; in PGM retransmissions typically emanate from the sender, whereas in LMS they come from repliers in most cases. We believe these schemes reflect two diametrically opposed approaches in the router-assist solution space, with most other schemes falling somewhere in between. Thus, by studying these two schemes we cover a reasonable part of the solution space.

Why is it important to study performance under incremental deployment? Ideally, the performance of router-assist schemes should be acceptable even under sparse deployment. Schemes that only achieve good performance at near-full deployment are far less likely to be adopted than schemes that offer a clearly demonstrable benefit even at sparse deployment.

With our study we hope to provide clues to help network planners determine the best deployment strategies that suit their need, the lower threshold of deployment where the benefit justifies the cost, and the upper threshold where more deployment provides diminishing returns.

In an earlier work, we have studied the performance of LMS under incremental deployment [14], but under a limited setting. The current work improves upon the earlier work in several important aspects:

- This work measures the performance of both LMS and PGM under incremental deployment.
- In this work we use a real Internet topology of 27,646 nodes, discovered through topology mapping software [15], whereas the early work used small topologies (about 400 nodes) generated by GT-ITM [16].
- This work investigates the performance of PGM and LMS under more realistic deployment strategies, like Fanout-on-the-Network deployment, AS deployment, and

AS-Border-Router deployment, as well as the Random deployment, Distance-from-Sender deployment, and the Fanout-in-Tree deployment.

In our work, we measure the performance of PGM and LMS in term of their network overhead, and the implosion problem. Based on the simulation results, the Fanout-in-the-Multicast-Tree deployment proves to be one of the best deployment strategies for both PGM and LMS. Using this strategy, the performance of PGM and LMS can match the performance under full deployment even at relatively sparse deployment levels. The Fanout-on-the-Network strategy is a good approximation of the Fanout-in-Multicast-Tree strategy in performance, and it is unrelated with individual multicast tree. The Distance-from-the-Sender strategy performs better in PGM than in LMS, because of the different data recovery mechanisms in PGM and LMS. Similarly the effectiveness of the AS and AS-Border-Router strategies in PGM and LMS differs. The Random deployment turns out to be the worst scheme under most situations.

The rest of the paper is organized as follows. In Section II we present the details of the data recovery mechanisms in PGM and LMS, and describe the evaluation metrics. Section III describes the six incremental deployment strategies. Section IV presents and discusses the simulation results on real-network topologies. Related work and conclusion are in Section V and Section VI respectively.

II. Router-Assisted Reliable Multicast Schemes

In this paper, we focus on the following two router-assisted reliable multicast schemes. The first one is the Pragmatic General Multicast (PGM) [9], while the second is the Lightweight Multicast Service (LMS) [2]. For simplicity, we do not consider the operations related to the loss of a control packet (either a Nak or a Ncf).

A. PGM

PGM includes the following basic operations:

- *Source Path State Establishment*: the source will periodically multicast out a Source Path Message (SPMs) to establish source path state in the network. When forwarding a SPM to its downstream nodes, a PGM router will include its own address into the SPM. In this way, a PGM router or receiver can know the address of its upstream PGM router.
- *Nak Generation*: upon detecting a packet loss, a receiver will set a back-off timer. When the timer expires, the receiver will generate a Nak for the lost packet and unicast it to its upstream PGM router.
- *Nak Aggregation and Suppression*: when receiving a Nak, a PGM router will add the interface from which the Nak arrives to the repair interface list for the lost packet. For Nak suppression, the PGM router will immediately multicast a Nak confirmation (Ncf) packet along that interface. When a receiver receives the Ncf before its timer expires, it will cancel the timer, and no Nak will be generated. When a downstream PGM router receives the Ncf, it will stop the propagation of the Ncf, and meanwhile it will refrain from forwarding any new Nak for the lost packet to the upstream PGM router. For Nak Aggregation, the PGM router will not forward a Nak upward if it has forwarded a Nak for the same lost packet upward before.
- *Retransmission of the Data Packet*: When the sender receives a Nak, it will first multicast a Ncf out, just like what a PGM router does. Then it will multicast the repair packet along the interface from which the Nak arrives. When a PGM router receives the repair packet, it will multicast the packet along all interfaces in the repair interface list for the packet. A router that is not PGM capable will simply forward the multicast packet (including the Ncf and the Repair Packet) on all downstream interfaces.

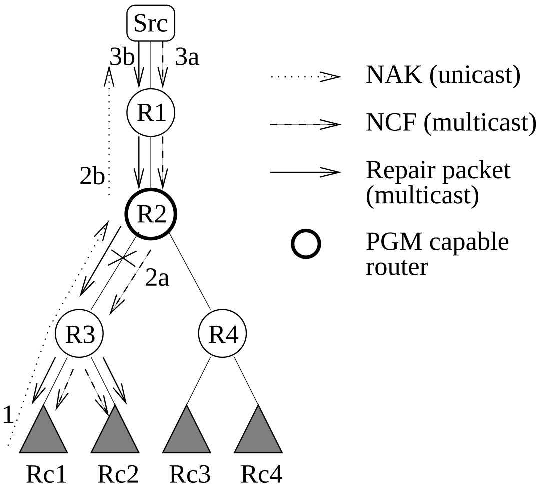

Fig. 1. PGM example

For example, on Figure 1, only R2 is PGM capable. Suppose a packet is lost between R2 and R3. Upon detecting the lost, Rc1 and Rc2 will set their timer. Here we assume that Rc1's timer goes off first. So at step 1, Rc1 will originate a Nak and unicast it to R2. When receiving this Nak, R2 will first multicast a Ncf along the interface to R3 at step 2a. This Ncf will reach Rc1 and Rc2. Suppose the Ncf arrives before Rc2's timer expires. Then Rc2 will cancel its timer, and no Nak will be generated from Rc2. After sending out the Ncf, R2 will also forward the Nak to the Source at step 2b. Upon receiving the Nak, the source will first multicast a Ncf out at step 3a. R2 will stop this Ncf from going further down the multicast tree. At step 3b, the source multicasts the repair packet along the interface to

R1, and R1 forwards it to R2. Then R2 forwards it to R3, and finally it reaches Rc1 and Rc2.

B. LMS

The original LMS scheme [2] has to be modified to deal with incremental deployment [14], because the original assumption that the replier link is always connected with a LMS capable router is no longer true in incremental deployment. In this paper, we assume the following procedures for LMS under incremental deployment.

- *Source Path State Establishment*: Similar to PGM, the sender in LMS will periodically multicast a SPM packet to establish the source path state. Each LMS node (including the receiver and the LMS router) will then know the address of its upstream LMS router.
- *Replier Selection*: each receiver will unicast a Replier Ready Message (RRM), which essentially says that it wants to be a replier, to its upstream LMS router. Those LMS routers will then unicast the RRM to their own upstream LMS routers. A LMS router will choose a replier based on the distance between itself and the replier. Unlike the original LMS scheme, in which the router simply records its link which can lead to the replier, here the LMS router has to record the address of the sender of the RRM, which can be either the replier or a downstream LMS router that leads to the replier. For convenience, we call this address the replying address for the LMS router.
- *Nak Forwarding*: When a receiver detects a packet loss, it will unicast a Nak to its upstream LMS router. In the Nak packet, the receiver should list its own address as the Nak originator address. Upon receiving the Nak, a LMS router will forward the Nak as below. If it has a replier, and the Nak comes from an address other than its upstream LMS router or its replying address, then this router is the turning point for the Nak. It will turn the Nak to its replying address, and in the Nak it will include its own address and the interface from which the Nak arrive (the subcast interface). Otherwise, if it has a replier, and the Nak comes from its upstream LMS router, then it simply unicasts the Nak to its own replying address without any modification to the Nak. If it does not have a replier, or the Nak comes from its replying address, then the router should forward the Nak to its upstream LMS router. In this way, a Nak will eventually reach either a replier or the sender.
- *Unicasting a Repair Packet*: When the sender receives a Nak, or when a replier receives a Nak and it has the requested data, the sender or the replier will then unicast a repair packet to the originator of the Nak. The originator of the Nak is the receiver who first sends out the Nak, and its address is specified in the Nak.
- *Subcasting a Repair Packet*: When a replier receives a Nak and it does not have the requested data, the replier will add the address of the turning point and the subcast interface to the subcast list for the lost packet. After it receives the repair packet through unicast (from the sender or another replier who has the data), it will unicast a subcast request packet containing the data to the turning point, and ask the turning point to multicast the repair packet on the subcast interfaces that are listed in the subcast request packet. If the replier receives the repair packet through multicast, then it should delete the subcast list for the lost packet, because in this case, an upstream replier has taken care of the repair work. So we see the recovery in LMS takes place in a two-stage process in most packet losses. The first stage is the sender or an upstream replier unicasts the repair packet to a replier. In the second stage, the replier sends the subcast request to the turning point, and ask the turning point to multicast the repair packet on the subcast interfaces. This two-stage recovery process can successfully reduce the amount of data traffic in sparse deployment because for most packet losses, the multicast of the repair packet is restricted only in the subtree that witnessed the packet loss.

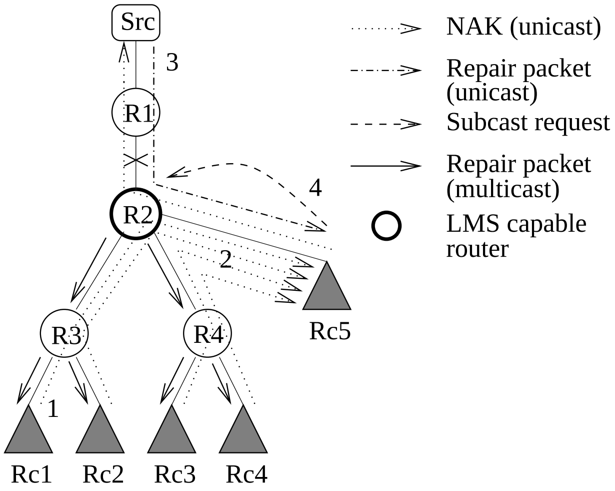

Fig. 2. LMS example

The data recovery in LMS can be illustrated through the example in Figure 2. In this example, only R2 is LMS capable, and it chooses Rc5 as its replier. Suppose a packet is lost in the link between R1 and R2. Upon detecting the packet loss, Rc1, Rc2, Rc3, Rc4, and Rc5 will all originate a Nak and unicast it to R2 at step 1. At step 2, R2 will turn the Naks sent by Rc1, Rc2, Rc3 and Rc4 to its replying address Rc5, and forward the Nak sent by Rc5 to the Source. When turning the Naks to Rc5, R2 will attach its own address and the subcast interface into the Nak. When Rc5 receives the Naks, it will add R2 and the subcast interfaces into the subcast list. At step 3, the Source receives the Nak, and unicasts a repair packet to Rc5. After getting the repair packet, Rc5 will unicast a subcast Request to R2, and ask R2 to multicast the repair packet on the interface to R3, and the interface to R4, at step 4. Eventually Rc1, Rc2, Rc3, Rc4 will get the repair packet.

C. Metric Space

There are a number of indices to measure the performance of PGM and LMS. In this paper, we focus on those for the network overhead and the implosion problem.

- *Normalized Data traffic overhead.* It is defined as the ratio of the amount of network resources used to transmit the repair packets (in term of the number of transmissions of the repair packet), and the size of the subtree (in number of links) that did not receive the data. In the ideal case the data network overhead will be 1.0, e.g., when the node right above the lost link has the data and it sends a single multicast packet down the subtree that did observe the loss. The formula to it across all link losses is:

$$NormDataOverhead = \frac{\sum_{links(l)} \frac{Data(l)}{Subtree(l)}}{NumberOfLinks}$$

where $Data(l)$ is the amount of data traffic that will be generated when the packet loss is on link l, $Subtree(l)$ is the size of the subtree (in term of number of links) that did not receive the data when the packet loss is on link l, i.e., the subtree below (and including) link l, and $NumberOfLinks$ is the total number of links in the topology. We assume that a data packet has an equal loss opportunity on any link.

- *Normalized Control traffic overhead.* Similar to the normalized data traffic overhead, the normalized control overhead is defined as the ratio of the amount of network resources used by the control packets (the Naks and Ncfs), and the size of the subtree that did not receive the data. We consider ratio of 1.0 as optimal, even though this is not the theoretically lowest ratio. For example, in LMS, if the node right above the lost link was a replier, the control overhead will be 1.0 if there was exactly one Nak sent over all of the links of the subtree below the lost link before the replier receive a Nak. Similar to the data overhead, the formula we use to compute the average control overhead across all link losses is:

$$NormControlOverhead = \frac{\sum_{links(l)} \frac{Control(l)}{Subtree(l)}}{NumberOfLinks}$$

where $Control(l)$ is the total amount of control traffic that will be generated in the network when the packet loss is on link l.

- *Maximum Averaged Naks.* This is the maximum value of the Averaged Naks among the sender, receivers, and PGM/LMS routers. The Averaged Naks of a node is defined as the total number of Naks received by the node for all link losses, divided by the total number of links in the multicast tree. This measurement represents the worst implosion problem witnessed by the sender, a receiver, or a PGM/LMS router over a long time. The formula to compute across all link losses is:

$$MaxAverageNaks = \max_i\left(\frac{\sum_{links(l)} Naks(l,i)}{NumberOfLinks}\right)$$

where node i is either the sender, a receiver, or a PGM/LMS router, $Naks(l,i)$ is the number of Naks received by node i when the packet loss is on link l.

- *Maximum Peak Naks.* This is the maximum value of Peak Naks among the sender, receivers, and all PGM/LMS routers. The Peak Naks of a node is defined as the maximum number of Naks received by the node in the recovery process of any single packet loss. This measurement represents the worst short time implosion problem at the sender, a receiver, or a PGM/LMS router. For the same multicast tree, the higher this value, the worse the implosion problem. The formula to compute it is:

$$MaxPeakNaks = \max_i(\max_l Naks(l,i))$$

D. Examples of Measuring PGM and LMS

We illustrate how to calculate the four metrics using the same examples in Figure 1 and Figure 2. In the PGM example where a loss happens between R2 and R3, the amount of Nak traffic is 2 + 2 = 4 (a Nak from Rc1 to R2, and a Nak from R2 to the source). The amount of Ncf traffic is 3 + 2 = 5. The normalized control overhead for this loss will be (4 + 5) / 3 = 3 (the size of the subtree that did not receive the data is 3). The amount of data traffic for this loss is 5. So the normalized data overhead will be 5 / 3 = 1.67. In this example, no matter on which link the packet is lost, the source will only receive one Nak. So both the Averaged Naks and the Peak Naks of the source will be 1. Suppose a Ncf is always successful in suppressing the Naks from other receivers, then the Peak Naks for R2 will be 2. The Averaged Naks for R2 will be (4 * 1 + 2 * 1 + 2 * 2) / 8 = 1.25 (the total number of links is 8). The amount of Naks received by Rc1, Rc2, Rc3, and Rc4 is always zero. So the Maximum Averaged Naks among the sender, receivers, and PGM routers will be 1.25, and the Maximum Peak Naks is 2. For the LMS example in Figure 2 where the loss happens between R1 and R2, the amount of Nak traffic is 4 * 3 + 3 = 15. So the normalized control overhead is 15 / 8 = 1.875 (the size of the subtree that did not receive the data is 8). The normalized data traffic is (3 + 1 + 6) / 8 = 1.25. The number of Naks received by the source in any link loss will be either 1 or 0. The Peak Naks of R2 across all link losses is 5 (like the loss between R1 and R2). Its Averaged Naks is (4 * 1 + 2 * 2 + 1 + 2 * 5) / 9 = 2.11 (the total number of links is 9). The Peak Naks of Rc5 across all link losses is 4 (like the loss on the link between R1 and R2). Its Averaged Naks is (4 * 1 + 2 * 2 + 2 * 4) / 9 = 1.78. So the Maximum Averaged Naks among the sender, receivers, and LMS routers is 2.11, and the Maximum Peak Naks is 5.

III. Incremental Deployment Strategies

There are many different strategies for incremental deployment of PGM and LMS. We can divide them roughly into two categories.

A. Multicast Tree Based Deployments

Deployments in this category are related with individual multicast tree. In this paper, we studied the following two multicast tree based deployments:

- *Fanout-in-the-Multicast-Tree deployment.* In this deployment, we first deploy PGM/LMS on the routers that have the largest number of downstream children in the multicast tree (fanout in the multicast tree), and then on the routers that has the second largest fanout in the multicast tree, and so on, until all routers in the multicast tree are enabled.
- *Distance-from-the-Sender deployment.* In this approach, we will first deploy PGM/LMS on the routers that are one hop away from the sender, and then on the routers that are two hops away, and so on.

B. Network Topology Based Deployments

Deployments in this category are purely related with the network topology, and have no relations with any individual multicast tree. They include:

- *Fanout-on-the-Network deployment.* In this deployment scheme, we will first deploy PGM/LMS on the routers that have the largest number of neighbors on the network topology (fanout on the network), and then on the routers with the second largest fanout on the network, and so on.
- *AS deployment.* In this approach, we first deploy PGM/LMS on all routers in the AS that has the largest number of routers, and then on all routers in the AS with the second largest number of routers, and so on.
- *AS-Border-Router deployment.* In this deployment, we first deploy PGM/LMS on all border routers in the AS that has the largest number of router, and then on all border routers in the AS with the second largest number of routers, and so on.
- *Random deployment.* In this deployment, we randomly choose routers from the network and deploy PGM/LMS on them.

IV. Simulation Results

A. Simulation Setup

In our simulation, we use a router-level Internet Core topology of 27,646 nodes [15], [17]. We assume a single-source multicast tree with the sender at the root of the tree. We randomly choose 5% routers (1382) from them. An extra node is added to the network as the sender, and an extra link is added to connect this sender with a router that is randomly chosen from the 1382 routers. Each of the rest 1381 routers is also connected with an additional node that acts as a receiver. So in the final topology, there are one sender, 1381 receivers, and 27646 routers. In the multicast group we use here, the average distance between the sender and a receiver is 9.28 hops. and the longest distance is 18 hops. The multicast tree consists of 9832 links, and 4917 nodes.

In addition, in our simulation, we set the receiver's back-off timer interval for PGM to be four times the worst round-trip-time (RTT) between the sender and any receiver. This is different with the dynamic adjustment scheme in the PGM draft, which set the back-off interval based on the number of Naks received in the PGM router and the number of its first PGM-hop children. As we can see from the discussions in the Simulation Results Sensitivity Section, this difference has small effect on the normalized data overhead and control overhead, and our results on Maximum Averaged Naks and Peak Naks can also provide useful insights on the figures with the dynamic adjustment scheme.

The results for Random deployment were obtained by averaging the results over 20 runs with different random seeds for selecting routers and also setting back-off timer in PGM. The results for other deployment strategies in PGM were obtained by averaging the results over 10 simulations with different seeds for the back-off timer. Results for other deployment strategies in LMS were obtained in one single run. In the graph, the X-axis represents the percentage of routers in the whole multicast tree that are PGM/LMS capable.

B. Simulation Results for PGM

Figure 3 shows the normalized data overhead for PGM under different deployment strategies. We can see that it starts with a very high value. The main reason for this is that at zero deployment, the repair packet will be multicasted to the whole subtree below one of the sender's in-

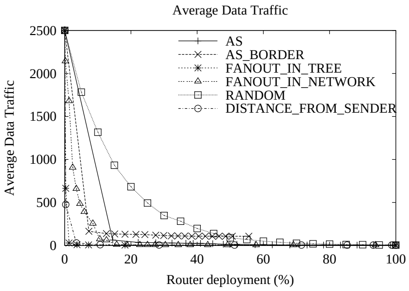

Fig. 3. PGM average data traffic

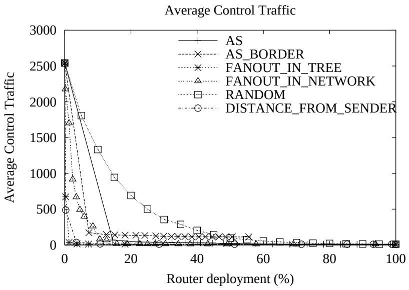

Fig. 4. PGM average control traffic

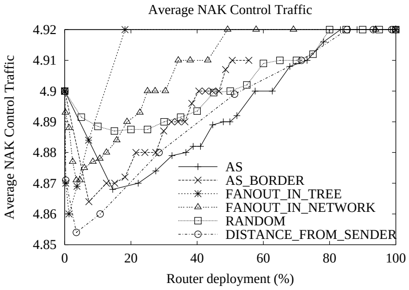

Fig. 5. PGM average NAK control traffic

terfaces, even if only one receiver in that subtree did not receive the packet.

As more and more routers become PGM capable, the data overhead decreases, and eventually drop to 4.92 at full deployment. Among the six deployment strategies, the Fanout-in-Tree strategy and the Distance-from-Sender strategy are the best ones. In the Fanout-in-Tree strategy, the data overhead dropping to 9.31 at 3.7%, and to 4.92 at 18.2%. Clearly, by deploying PGM on routers with a large number of downstream nodes, we can target the retransmission more accurately to the receivers that need the packet.

The result of the Distance-from-Sender strategy is very close to the Fanout-in-Tree strategy, with 29.94 at 3.5% and 6.4 at 28.5%. There are two main reasons for this. First a router close to the sender is more likely to have a large number of children. Second, the size of the subtree rooted at a router close to the sender is usually larger than the size of subtree rooted at a router close to the receiver. So by deploying PGM on a router close the sender, we can reduce the number of repair packet transmission over unwanted links.

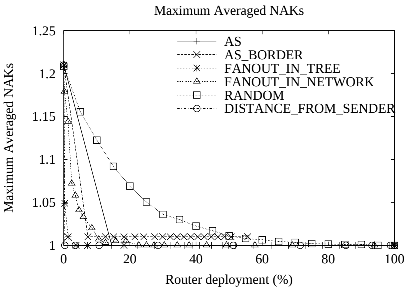

Fig. 6. PGM maximum averaged NAKs

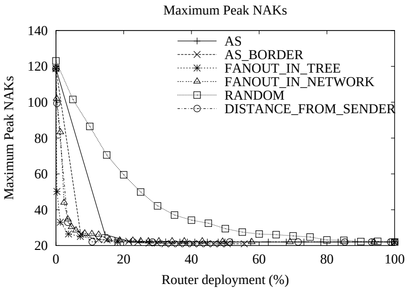

Fig. 7. PGM maximum peak NAKs

The good performance of the Fanout-on-Network strategy shown in the graph can be explained as the more neighbors a router has on the network, the more likely the chance of having a large number of children in the multicast tree. The performance of the AS-Border-Router deployment and the AS deployment depends to a large extend on the sender's location. When the AS which the sender belongs to is chosen to be deployed, we will see a significant drop in the data overhead, like in this simulation where the sender belongs to the largest AS, we see a big drop when the first AS is deployed (at 7.3% for AS-Border-Router and 14.5% for AS). By enabling routers in the same AS with the sender, we are enabling routers close to the sender.

The normalized control overhead shown in Figure 4 is quite similar with the normalized data overhead. This is due to the fact that the majority of the control overhead in sparse deployment comes from the Ncf packet, and the transmission of Ncf is similar with the repair packet. The exact number of Ncf transmissions is equal or slight higher than that of the repair packet, because a Ncf is not always successful in suppressing Naks, so sometimes multiple Nak packets for the same lost packet may arrive on

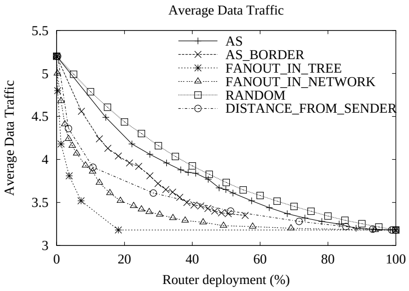

Fig. 8. LMS average data traffic

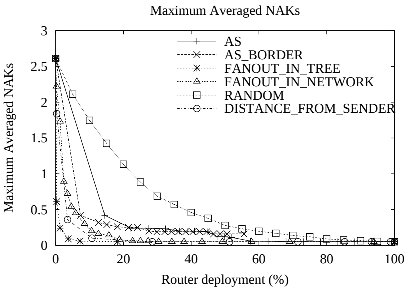

Fig. 10. LMS maximum averaged NAKs

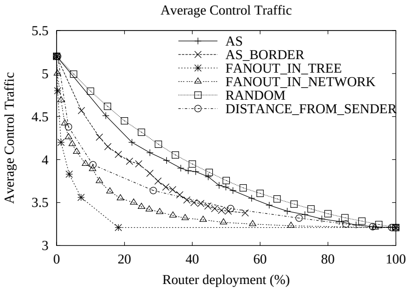

Fig. 9. LMS average control traffic

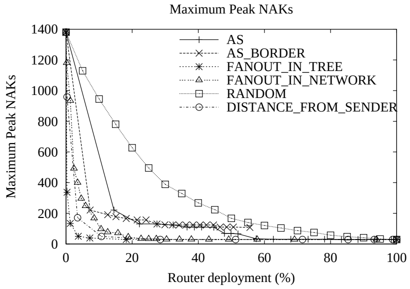

Fig. 11. LMS maximum peak NAKs

the same interface, causing multiple Ncf being multicasted over the link, whereas the repair packet is always transmitted over a link at most once.

We can see from Figure 5 that the normalized control overhead for Nak packet is very low, and stays within a narrow range. It is interesting to notice that in this graph, for all deployment schemes, the Nak control overhead first goes down and then goes up. This can be explained by the dual effect of deploying PGM on a router. When a router becomes PGM capable, it can reduce the number of Nak transmissions by aggregating Naks sent by downstream nodes. But on the other hand, enabling this router will stop the propagation of a Ncf sent by its upstream node, hence limit the Nak suppression benefit of Ncf. The first effect outweighs the second effect in sparse deployment, while the second effect becomes more obvious when the deployment goes high.

Figure 6 and Figure 7 show the Maximum Averaged Naks and Maximum Peak Naks witnessed by the sender, receivers, and PGM routers. Due to the Nak suppression benefit of Ncf, we see the number of Naks received by the sender is far below the number of receivers in zero deployment. At full deployment the Maximum Averaged Naks and Peak Naks drop to 1 and 22 respectively (22 is the largest fanout in the multicast tree). Here, we again see the Fanout-in-Tree strategy and the Distance-from-Sender strategy perform best, followed closely by the Fanout-on-Network deployment, and the AS deployment and AS-Border-Router deployment. Clearly, a router with a large fanout in the multicast tree can do a better job in Nak aggregation, and enabling routers close to the sender can ease the Nak implosion problem at the sender.

C. *Simulation Results for LMS*

Figure 8 and Figure 9 show the normalized data overhead and control overhead for LMS, with the control overhead slightly higher than the data overhead in most cases. We can see that both of them start with 5.2, a pretty low value compared with the value for PGM. They decrease as the percentage of router deployment goes up. Like in PGM, here the Fanout-in-Tree strategy also has an outstanding performance, with both data overhead and control overhead reaching their lowest values at 18.2%. By deploying LMS on a router with a large number of children, a large amount of Naks sent from downstream nodes can be turned to the replier by this router. And the replier will carry out the recovery process locally if it has the data, without bothering an upstream replier or the sender. Even if the replier does not have the data, there will be only one Nak sent upward by the router.

Unlike in PGM, here the Distance-from-Sender strategy lags far behind the Fanout-in-Tree strategy, and is only close to the Fanout-on-Network strategy. In other words, the benefit of deploying LMS on routers close to the sender is not so significant as in PGM, in terms of reducing network overhead, even though they are still good candidates

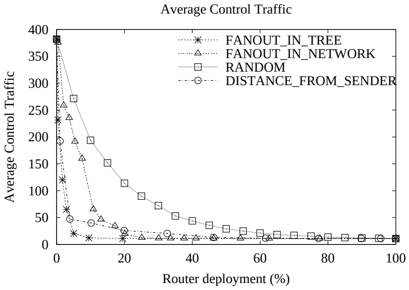

Fig. 12. Topology sensitivity: PGM average control overhead in Mbone

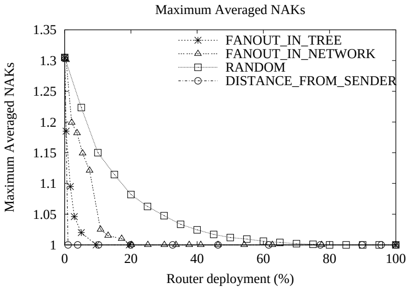

Fig. 13. Topology sensitivity: PGM maximum averaged NAKs in Mbone

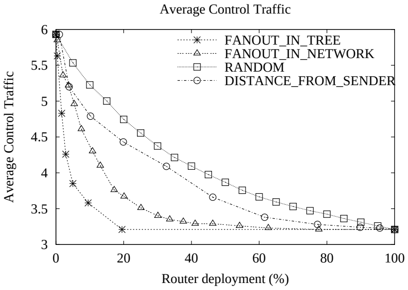

Fig. 14. Topology sensitivity: LMS average control overhead in Mbone

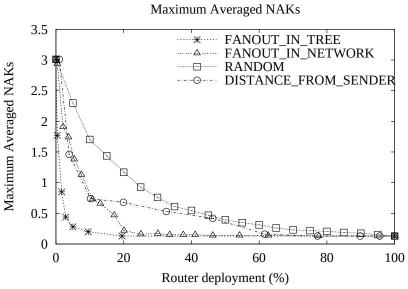

Fig. 15. Topology sensitivity: LMS maximum averaged NAKs in Mbone

as their chances of having large amount of children are high. This is due to the different data recovery mechanism used in LMS. In LMS, Naks are turned to a close replier to exploit the benefit of recovering the data locally from the replier, while in PGM, the sender is responsible to emanate a repair packet whenever a loss happens. So LMS is not only concerned on whether the router can steer large amount of Naks to a close replier, but also on whether the router is close the receivers and its replier. Similarly, in LMS, the location of the sender plays a less important role in affecting network overhead, so we just see a gradual decrease in the network overhead as the deployment goes up under both the AS and AS-Border-Router strategies. But they are still better than the Random deployment. From Figure 11, we see the Maximum Peak Naks in LMS starts with a very high value, equal to the number of receivers. This is because at zero deployment, all Naks and Repair Packets are all sent through unicast, which results in implosion at the sender. This problem is eased as more and more routers become LMS capable. Still, the Fanout-in-Tree strategy performs best. And the Fanout-on-Network strategy is also pretty good. Contrary to its role in reducing network overhead, here we see the router's distance from the sender could do more in reducing the Maximum Peak Naks. In the graph, we see the Distance-from-Sender deployment is pretty good. For the AS-Border-Router deployment and AS deployment, there are also big drop when the AS where the sender resides is chosen to be deployed, which is similar with the case in PGM. By enabling the routers close to the sender in LMS, these routers can divert Nak packets to nearby receivers and reduce the Naks seen by the sender.

The implosion problem in LMS for long time period looks better than in short-time period, as we can see from the Maximum Averaged Naks in Figure 10. The trend of different deployment schemes resembles the trend in Figure 11. We should notice that it is possible that LMS also adopt the back-off timer mechanism to ease the implosion problem, like in PGM, but that would result in an increase in the network overhead and recovery latency.

D. Simulation Results Sensitivity

Clearly a number of factors could affect our simulation results. They include the network topology, the choice of the multicast group, and the Nak back-off timer interval. But as we can see below, even though the exact value of the simulation results may change, the overall trend of the six deployment strategies in both PGM and LMS remains the same.

- *Network Topology Sensitivity*. Figures 12, 13, 14 and 15 show the control overhead and maximum averaged Naks of PGM and LMS on the Mbone topology with 4387 nodes [17]. The multicast group we use here has 208 receivers. The average distance from the sender to a receiver

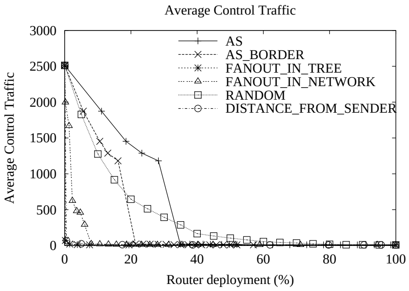

Fig. 16. Multicast group sensitivity: PGM average control overhead

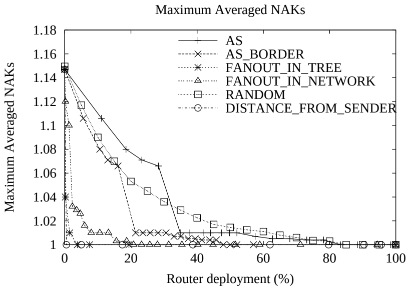

Fig. 17. Multicast group sensitivity: PGM maximum averaged Naks

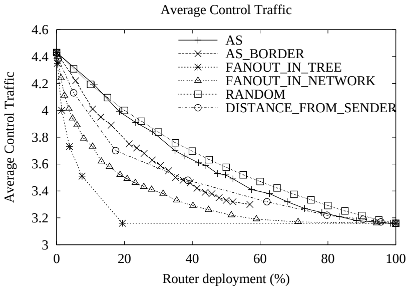

Fig. 18. Multicast group sensitivity: LMS average control overhead

is 10.49, and the longest distance is 18. There are 1450 links and 726 nodes in the multicast tree. Due to the lack of AS information on the Mbone topology, we only show the results for the four deployment strategies with no relations to AS. We can see clearly that overall trend of these strategies remains the same. The same observation applies to the data overhead and Maximum Peak Naks.

• *Multicast group.* This includes the number of receivers, the location of the sender, and the multicast tree structure. It is possible that we can end up with very different results if we use some extreme examples. For example, the Fanout-in-Tree strategy will have no meaning in a binary multicast tree with all receivers at the leaves, as every router in the tree has the 2 downstream children. We have run the simulation over dozens of randomly generated

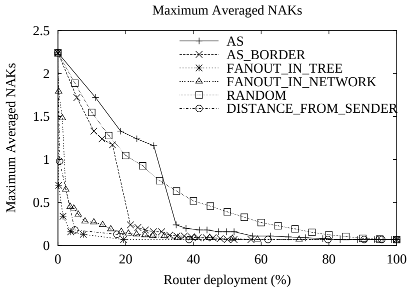

Fig. 19. Multicast group sensitivity: LMS maximum averaged Naks

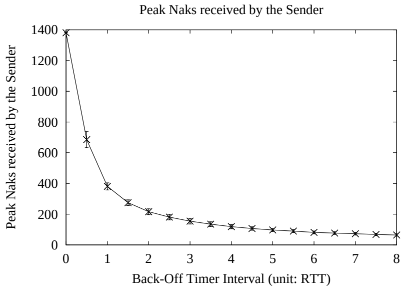

Fig. 20. Back-off interval sensitivity: PGM peak Naks at the sender

multicast groups with different number of receivers (ranging from 1% to 10%), different sender locations, and different tree structures, and found the overall trend remains the same. Figure 16, Figure 17, Figure 18 and Figure 19 show the control overhead and Maximum Averaged Naks of PGM and LMS on a different multicast group with the 1381 receivers on the router-level Internet topology. In this multicast group, the sender is located in a small AS. The average distance from the sender to a receiver is 7.78, and the longest distance is 17. The multicast tree has 9620 links and 4811 nodes. We see the performance of PGM in AS-Border-Router and AS strategies are close to or even worse than the Random deployment, before PGM is deployed in the AS where the sender resides at 21.4% and 34.9% respectively. At these two point, we see a big drop in both the control overhead and the Maximum averaged Naks. In LMS, we still see a graduate decrease in control overhead for the AS-Border-Router and AS deployments, but a significant drop in Maximum averaged Naks for them at at 21.4% and 34.9% respectively.

• *Nak back-off timer interval.* The Nak back-off timer interval has an important role on how successfully a Ncf can suppress the generation of Naks in PGM. Figure 20 shows its effect on Peak Naks at the sender in zero deployment in PGM. The unit in the X-axis is the worst RTT between the sender and any receiver. The results were obtained by averaged over 10 simulations on the same multicast group as the one we use for the main simulation. The graph includes the 95% confidence interval, even though in most cases this confidence interval is very small to be noticed. From this graph, we see as the interval goes up, its influence on the peak Naks is diminishing. In our work, we choose the 4 * RTT as the Nak back-off timer interval. Even though this is different with the dynamic adjustment in the PGM draft which is based on the number of Naks received in the PGM router and the number of its first PGM-hop children, we believe that this setting is reasonable, and our simulation results can provide helpful insights on the real figure with the dynamic adjustment. First, as we can see in the graph, 4*RTT is big enough to get a good representation. Second, in the real world, there is always a discrepancy between the adjusted value and the optimal value, as the number of Naks received in the PGM router can vary significantly across the time.

V. Related Work

Reliable multicast is an area that has been actively studied in the past, and a variety of schemes have been proposed. The proposed solutions can be classified into two types: schemes that require router support (also known as router-assisted hierarchical schemes), and schemes that do not require such support (also known as application-level hierarchical schemes).

The Reliable Multicast Transport Protocol (RMTP) [3] is an example of an application-level hierarchical scheme. In RMTP the receivers are organized into a manual hierarchy, and then the children unicast their acknowledgments to the parents. Data that is not acknowledged is retransmitted by using either unicast from a parent to a child, or by multicast over a local group. The Tree-based Multicast Transport Protocol (TMTP) [8] is another example of a scheme that does not require router support. Unlike RMTP, the hierarchy in TMTP is dynamically created by using expanding ring search. Other application-level schemes are Scalable Reliable Multicast(SRP) [1], LGMP [18], and Tree-based Reliable Multicast Protocol (TRAM) [5]. Application-level hierarchical schemes have the advantage that their deployment does not depend on router support, but only on the support from the participants, therefore the work in this paper does not apply to such schemes.

Router-assisted schemes have the advantage that their recovery hierarchy is more congruent to the underlying multicast tree. Examples of such schemes are PGM [9] and LMS [2] which are studied in this paper. Search Party [10] is a scheme heavily inspired by LMS that uses *randomcast* to forward a request at random instead of toward a preselected replier. Addressable Internet Multicast (AIM) [7] is a scheme that requires routers to assign per-multicast group labels to all participating routers. These labels are used to redirect requests to the nearest upstream member. In Active Error Recovery (AER) [11], routers that have repair servers attached periodically announce their existence to the downstream routers and receivers. OTERS [6] and Tracer [12] use the help of the mtrace [19] utility to build the hierarchy. These schemes require support from the routers in order to operate correctly and efficiently.

The only two works we are aware of that study the impact of incremental deployment for reliable multicast are [14] and [20]. The first one investigates the performance of LMS under various deployment schemes; however it does not consider PGM or any other schemes. The topologies for that study are generated by the GT-ITM [16] topology generator and are quite small (around 400 nodes). In this paper we use a real Internet router-level topology with 27646 nodes. This topology is bigger and more realistic than the generated topologies, and it also contains AS information so that we can study deployment schemes related with AS.

Active Reliable Multicast(ARM) [20] utilizes soft-state storage within the network for NACK suppression and to limit the data retransmission only to receivers that have observed losses. Incremental deployment strategies are studied within the context of ARM. The authors have found that significant benefits can be obtained when only 50% of the routers are ARM-capable. Based on their results, the authors also suggest that the same benefit can be obtained even with a much smaller set of ARM-capable routers if they are placed at strategic locations, but how to find those locations is suggested as a future work.

VI. Conclusions

In this paper, we present the simulation results for six different incremental deployment strategies for both PGM and LMS in real-world topologies. From the results, we see that at sparse deployment, PGM could suffers from the huge network overhead, while LMS could suffer from the implosion problem. By choosing the right deployment schemes, we can make significant improvement on their performance.

Among the six strategies, the Fanout-in-Multicast-Tree deployment proves to be one of the best strategies for both PGM and LMS. Using this strategy, the performance of PGM and LMS can match the performance in full deployment even at relatively sparse deployment levels. The Fanout-on-the-Network strategy is a good approximation of the Fanout-in-Multicast-Tree strategy in performance for both PGM and LMS, and it is unrelated with individual multicast tree. The results also show that the Distance-from-Sender scheme has a different impact on PGM and LMS. In PGM, it is also one of the best strategies. In LMS, its influence on the Maximum averaged Naks and peak Naks is greater than on the average data overhead and control overhead. The performance of the AS strategy and the AS-Border-Router strategy is related to the sender's location. There is a significant improvement on all four metrics for PGM and the last two metrics for LMS, when the AS where the sender resides is chosen to deploy PGM/LMS. Meanwhile, for the average data overhead and control overhead in LMS, the AS strategy and the AS-Border-Router strategy shows a more gradual improvement as deployment percentage goes up. The Random deployment scheme provides a comparison for the performance of other strategies, and in most cases is the worst scheme.

The findings in this paper can be explained by the data recovery mechanisms of PGM and LMS. In both PGM and LMS, a router with a large number of downstream children is always a good candidate for deployment. When such router is enabled with PGM, a large number of Naks sent by the downstream receivers can be aggregated at this router, and this router can help target the repair packet to the right receivers more accurately. In LMS, a router with large fanout in the multicast tree is a good choice for incremental deployment, as such router can turn a large number of Naks sent by downstream receivers to the replier, and let the replier to carry out local recovery without bothering an upstream replier or the sender.

The Distance-from-sender strategy proves to be good in the incremental deployment of PGM, as a router close to the sender plays an important role in reducing the number of transmissions of the repair packets and Ncf packets over unwanted links in PGM. It also has a good impact on the Maximum averaged Naks and peak Naks in LMS, as the routers close to the sender can divert Nak packets to nearby receivers, instead of sending them to the sender. But it does not perform well in terms of data overhead and control overhead in LMS, as LMS cares not only the number of Naks a router can steer to the replier, but also the distance from the router and the receivers (repliers).

References

[1] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, November 1997.

[2] Christos Papadopoulos, Guru Parulkar, and George 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] 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.

[4] H. Holbrook, S. Singhal, and D. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation," in *Proceedings of the ACM SIGCOMM'95*, Cambridge, MA, USA, August 1995, pp. 328–341.

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

[6] 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.

[7] 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.

[8] Rajendra Yavatkar, James Griffoen, and Madhu 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.

[9] Tony Speakman, Dino Farinacci, Steven Lin, Alex Tweedly, Nidhi Bhaskar, Richard Edmonstone, Kelly Morse Johnson, Rajitha Sumanasekera, Lorenzo Vicisano, Jon Crowcroft, Jim Gemmell, Dan Leshchiner, Michael Luby, Todd L. Montgomery, and Luigi Rizzo, "PGM Reliable Transport Protocol Specification," *Internet Draft, draft-speakman-pgm-spec-06.txt*, February 2001, Work in progress.

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

[11] Sneha K. Kasera, Supratik Bhattacharyya, Mark Keaton, Diane Kiwior, Jim Kurose, Don Towsley, and Steve Zabele, "Scalable Fair Reliable Multicast Using Active Services," *IEEE Network Magazine (Special Issue on Multicast)*, January/February 2000.

[12] Brian Neil Levine, Sanjoy 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.

[13] Brad Cain, Tony Speakman, and Don Towsley, "Generic Router Assist (GRA) Building Block Motivation and Architecture," *Internet Draft, draft-ietf-rmt-gra-arch-02.txt*, p. Work in progress, July 2001.

[14] Christos Papadopoulos and Emanouil 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.

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

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

[17] USC/ISI, "The SCAN Project," http://www.isi.edu/scan/.

[18] M. Hofmann, "Home page of the Local Group Concept (LGC)," http://www.telematik.informatik.uni-karlsruhe.de/˜hofmann/lgc/.

[19] Steve Casner and Ajit Thyagarajan, *mtrace(8): Tool to print multicast path from a source to a receiver*, UNIX manual page.

[20] Li wei Lehman, Stephen J. Garland, and David L. Tennenhouse, "Active Reliable Multicast," in *Proceedings of the IEEE Infocom'98*, San Francisco, USA, March 1998.