A Framework for Incremental Deployment Strategies for Router-Assisted Services

Xinming He
Computer Science Department
University of Southern California
Los Angeles, CA, USA
Email: xhe@usc.edu

Christos Papadopoulos
Computer Science Department
University of Southern California
Los Angeles, CA, USA
Email: christos@isi.edu

Pavlin Radoslavov
International Computer Science Institute
Berkeley, CA, USA
Email: pavlin@icsi.berkeley.edu

***Abstract*— Incremental deployment of a new network service or protocol is typically a hard problem, especially when it has to be deployed in the routers. First, an incrementally deployable protocol is needed. Second, a study of the performance impact of incremental deployment should be carried out to evaluate 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. Unfortunately, to date there has been no systematic evaluation of incremental deployment for such services.**

Our research work is focused on the second aspect, namely the performance impact of incremental deployment of router-assisted services. We take the first step to define a framework for evaluating incrementally deployable services, which consists of three parts: (a) selection and classification of deployment strategies; (b) definition of performance metrics; and (c) systematic evaluation of deployment strategies. As a case study for our framework, we evaluate the performance of router-assisted reliable multicast protocols. Although our framework is still evolving, our results clearly demonstrate that the choice of a strategy has a substantial impact on performance, and thus affirms the need for systematic evaluation of incremental deployment.

Our case study includes two router-assisted reliable multicast protocols, namely PGM and LMS. We make several interesting observations: (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, but with other strategies up to 80% deployment may be needed to approach the same level; (b) our sensitivity analysis reveals relatively small variation in the results in most cases; and (c) the penalty associated with partial deployment is different for each of these protocols; PGM tends to impact the network, whereas LMS impacts the endpoints.

I. Introduction

As the Internet evolves, new services must be deployed in the routers. Such services are typically deployed gradually due to the large scale and inherent heterogeneity of the Internet, hence the network goes through extended periods when a service is in a state of partial deployment. During this state, the performance and utility of the service may suffer. Thoughtful deployment strategies may tip the scale towards success, whereas careless strategies may cause failure of the new service.

However, selecting the proper deployment strategy is a hard problem. To date, there has been no systematic methodology or *framework* 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 take a first step to define a framework for evaluating incremental deployment strategies for router-assisted services. Our framework consists of the following three parts: (a) selection and classification of deployment strategies; (b) definition of the metrics to measure performance; and (c) systematic evaluation of deployment strategies. We use the following guidelines in defining our framework. 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 in the performance measurement. Finally, our evaluation of deployment strategies is done over a large-scale (over 27,000 nodes), mapped Internet topology to avoid potential artifacts from topology generators.

Our case study is incremental deployment of *router-assisted reliable multicast*. Our work has yielded two main contributions. The first contribution is the definition of the framework itself, which may be adapted and reused in other case studies. The second contribution is the results of our case study of router-assisted reliable multicast. Our results 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 reliable multicast router-assist schemes, namely PGM [1] and LMS [2], because they have specified 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], [3]. We are simply interested in how performance of these protocols changes with deployment.

Our study helps to understand the general behavior of router-assisted protocols under partial deployment, and the specific issues faced by each protocol. PGM and LMS differ significantly in their operation and thus behave differently under partial deployment. For example, in PGM routers aggregate NAKs and guide retransmissions where needed, whereas LMS defers these actions to the receivers with minimal assistance from the routers; additionally, in PGM retransmissions typically emanate from the sender, whereas in LMS they come from the receivers. Evaluating these protocols in the same framework helps distinguish their features more clearly.

An earlier work has studied the performance of LMS under incremental deployment [4], but in a more limited setting. Our current work is more thorough and contains several significant improvements, including: (a) this work defines a framework for systematic study of incremental deployment; (b) it studies both LMS and PGM under incremental deployment, whereas the previous work studied only LMS; (c) we use a mapped Internet topology of over 27 thousand nodes obtained by a topology mapping software [5], whereas the earlier work used much smaller topologies (about 400 nodes) generated by the GT-ITM [6] topology generator; (d) it investigates more realistic deployment strategies, such as strategies based on router fanout and AS size.

Our case study reveals some interesting results. First, of the four evaluation metrics we use, only one or two show a strong impact due to partial deployment. Second, for certain 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 nearly full benefits of router-assist can be realized very early in the deployment phase. Third, there is a significant difference among deployment strategies, with 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 on 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 framework. Section III includes an overview of the data recovery mechanisms in PGM and LMS, and the definition the evaluation metrics. Simulation results are presented in Section IV, followed by sensitivity discussion in Section V. Section VI reviews related work, and Section VII concludes the paper.

II. Incremental Deployment Strategies

In our framework, we classify deployment strategies into two main categories: *Network-Aware* and *Multicast-Tree-Aware*. Within these two categories, strategies can be subdivided further based on factors such as Autonomous System size, network connectivity, core-to-stub and stub-to-core, proximity to the sender, proximity to the receivers, and multicast tree connectivity.

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.

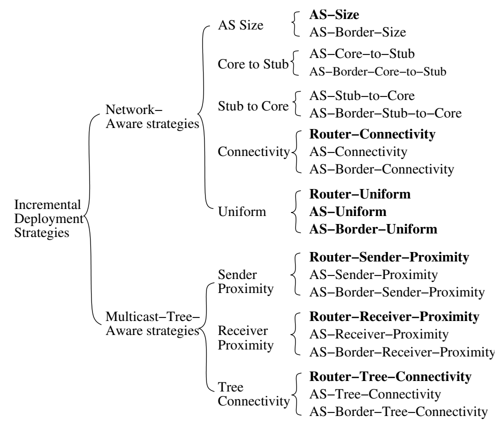

Fig. 1. Classification of Incremental Deployment Strategies, Strategies studied in this paper are in bold

A. Network-Aware Deployment Strategies

Network-Aware strategies utilize information about the general network structure, such as router-level and AS-level structure. However, for multicast services, these strategies are oblivious to the structure of multicast trees. 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 smaller ASs, according to their size. Such strategies assume that large ISPs will be the first to deploy the new service, 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 towards stub ASs. Conversely, with the Stub-to-Core strategies a service is first deployed at stub ASs, and then pushed towards 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. Deployment is faster because of the small number of those ISPs. 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 have studied, and they capture the scenario where deployment happens without any coordination.

Results from such strategies can form the baseline for evaluating other strategies.

B. Multicast-Tree-Aware Deployment Strategies

For multicast services, the knowledge of the multicast tree structure can play a significant role. Compared with the Network-Aware deployment strategies, Multicast-Tree-Aware strategies use information about multicast tree structure. While foregoing generality, such deployments should be studied for two reasons: (a) there are cases where 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. Deployment starts near the sender. Such strategies may be adopted when the sender charges for the new 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 in 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 according to the number of outgoing interfaces in their (Source, Group) multicast routing state, 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. These strategies can also be used to calibrate other strategies.

In our case study of router-assist reliable multicast, we have carried out intensive investigation on all the strategies, except the Core-to-Stub and Stub-to-Core strategies due to the lack of detailed data on the Internet AS-level hierarchy. Due to space limitation, we present only the results of eight strategies (these are 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 (requests and/or retransmissions). These messages may swamp the sender, the network, or the receivers. Exposure occurs when a retransmitted packet is delivered more than once to some 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 aspects: (a) ensuring the congruency between the data recovery process and the underlying multicast tree structure, and (b) allowing fine-grain multicast that helps to reduce the transmission on unwanted links. 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 reference [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 uses a random back-off interval, and then unicasts a NAK to its upstream PGM neighbor. Upon receiving a NAK, a PGM router creates the repair state that 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 propagated only to the downstream PGM neighbors, and are used also to suppress other pending NAKs. The router in turn unicasts a NAK to its upstream PGM neighbor, which is again followed by an NCF, and the process repeats until the NAK reaches the sender.

After the sender receives a NAK, it multicasts a repair packet. 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]

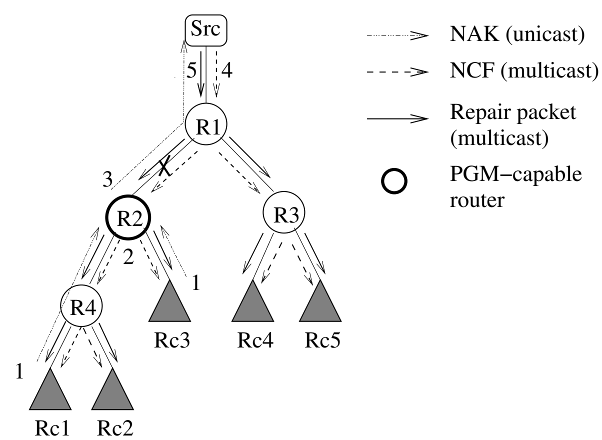

Fig. 2. PGM example

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

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] sketches an incremental deployment methodology. Here we expand on that methodology to define a slightly modified version of LMS that supports incremental deployment.

Similar to PGM, in LMS the sender periodically multicasts SPMs, so LMS routers and receivers may 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 candidate-repliers, based on cost 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: (a) if the NAK is originated from that router's replier, or there is no replier, the router unicasts a NAK to its upstream LMS neighbor; (b) if the NAK is originated from a non-replier, the router is the *turning point* for that NAK, therefore it inserts its own address and the interface the NAK arrives on before unicasting it 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, some other upstream replier has originated the repair, so any local repair state 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).

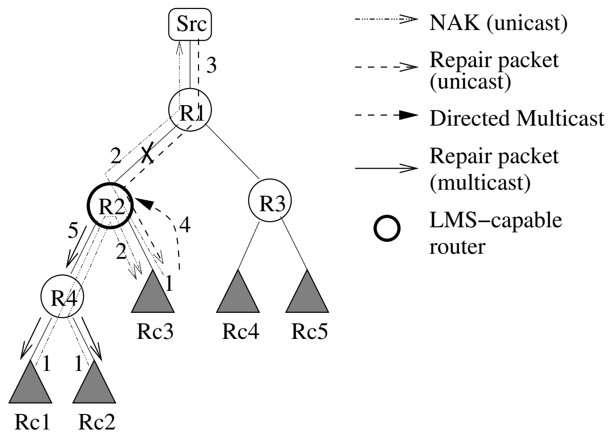

Fig. 3. LMS example

The following example illustrates data recovery in LMS. In Figure 3, assume that only $R2$ is LMS-capable, and it has selected $Rc3$ as its replier. Suppose 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

For both PGM and LMS, sparse deployment may result in an increase in exposure and/or implosion. Moreover, it may increase recovery latency. In this paper, we choose to focus on metrics that capture the most important problems in reliable multicast: implosion and exposure, and leave other aspects like recover latency for future investigation. We select the following metrics:

- *Average Normalized Data Overhead.* The data traffic network overhead is defined as the ratio of the amount of network resources used by the repair packet (in number of links traversed by the repair packet), and the size of the subtree (in number of links) that did not receive the packet. In the ideal case, the normalized data overhead would be 1.0 (*i.e.,* under full deployment and when the node right above the lossy link sends a single multicast packet to the loss subtree). We assume same packet loss probability on all links in the multicast tree. Therefore, the Average Normalized Data Overhead is determined by averaging the normalized value of measured overhead across packet loss on all links:

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

where $Data(l)$ is the number of links traversed by the repair packet for a single packet loss on link l, $Subtree(l)$ is the size of the subtree that did not receive the data, and $NumberOfLinks$ is the total number of links in the multicast tree.

- *Average Normalized Control Overhead.* Similar to the Average Normalized Data Overhead, the Average Normalized Control Overhead is defined as the ratio of the amount of network resources used by the control packets (NAKs and NCFs) and the size of the subtree that did not receive the data. We consider a ratio of 1.0 to be optimal, even though theoretically this is not the lowest ratio. Similar to the data overhead, control overhead is calculated as follows:

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

where $Control(l)$ is the total amount of control traffic that is generated when the packet is lost on link l, and $Subtree(l)$ and $NumberOfLinks$ are defined as before.
- *Maximum Average NAKs.* This is the maximum of the average NAKs received by any node. It is a measure of the worst case *sustained* implosion at any node, and is calculated as follows:

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

where node i can be the sender, a receiver, or a router, and $NAKs(l,i)$ is the number of NAKs received by node i when the packet loss is on link l. $NumberOfLinks$ is defined as before.
- *Maximum Peak NAKs*. Maximum Peak NAKs is the maximum number of NAKs received by a node during a single packet loss. This is a measure of the worst case *instantaneous* implosion at any node, and is calculated as follows:

$$MaxPeakNAKs = \max_i(\max_l NAKs(l,i))$$

where node i and $NAKs(l,i)$ are defined as before.

IV. Simulation Results

A. Simulation Setup

To evaluate the impact of different deployment strategies, we ran numerous simulations on a router-level Internet topology of 27,646 nodes [5], [7]. In our simulations we varied the group size, and used different receiver and sender placement models. In this section, we present the simulation results with the receiver and the sender being randomly placed on the network, and with the group size being 5% of the network size. Results for other setups are presented in Section V, where we study the sensitivity of those results.

For each set of parameters, we ran at least 50 simulations, using a different randomization seed for sender and receiver placement. We present the results averaged across all 50 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 participating in the multicast tree are deployed. Note that the AS-Border strategy usually can not reach full deployment as it only deploys border routers.

B. Simulation Assumptions

To reduce the complexity of the simulations, we make the following assumptions.

- *Control or repair packets are not lost.* While in reality recovery 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, not the protocol's ability to handle multiple losses. For the same reason we also assume that recovery periods do not overlap (i.e., recovery for a particular loss is completed before another loss occurs).
- *Uniform link loss probability.* We are not aware of any loss models for multicast traffic. To avoid making our own (possibly flawed) assumptions, we assume that all links have equal loss probability.
- *Optimal NAK suppression in PGM.* PGM employs heuristics for dynamically adjusting the back-off interval to eliminate NAK implosion. This interval depends on the population and topological distribution of the receivers. Since these heuristics are likely to be improved during operational experience, in our simulations we assume that the back-off intervals are well-tuned to achieve optimal NAK suppression; that is, no more than one NAK arrives along each downstream interface of a PGM node. Thus, our results approach the lower bound in control overhead. Note that this assumption has no impact on the data overhead.

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 the repair packets. In the absence of PGM routers to guide them to appropriate receivers, the repair packets flood the multicast tree.

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.

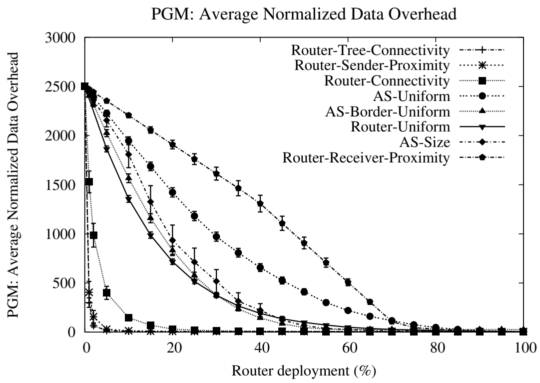

Fig. 4. PGM Average Normalized Data Overhead

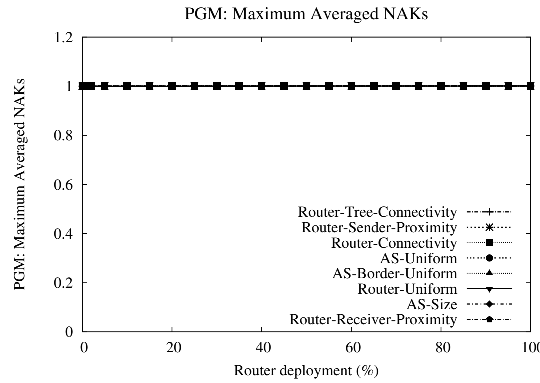

Fig. 6. PGM Maximum Averaged NAKs

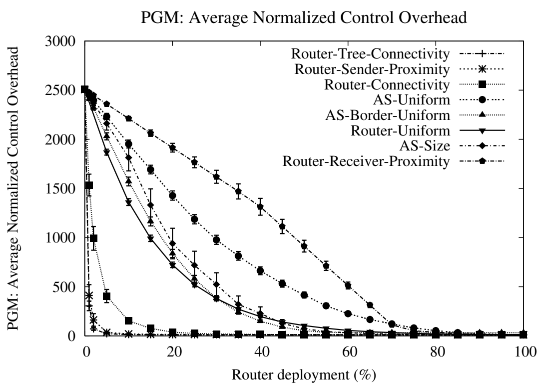

Fig. 5. PGM Average Normalized Control Overhead

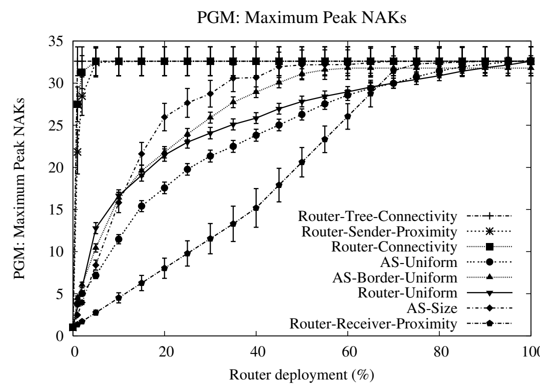

Fig. 7. PGM Maximum Peak NAKs

The second category includes Router-Uniform, AS-Uniform, AS-Border-Uniform, and AS-Size strategies. The third category contains the worst performer, the Router-Receiver-Proximity strategy.

In the first category, the Router-Tree-Connectivity and Router-Sender-Proximity strategies are both exceptional performers, achieving near full-deployment performance with only 5% of the routers deployed. The 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, the Router-Sender-Proximity strategy performs exceptionally well because a router near the sender is more likely to have a large fanout in the tree and large size subtrees, which plays an important role in the targeting of repair packets. Finally, the 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 the 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. The figure also shows that the AS-Size deployment strategy performs slightly worse than the Router-Uniform strategy when deployment level is lower 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 traverse exactly the same path as the repair packet under the optimal NAK suppression assumption.

Figures 6 and 7 show the Maximum Averaged NAKs and Maximum Peak NAKs in PGM respectively. We can see that with the optimal NAK suppression assumption, on average a router receives at most one NAK following a packet loss, and the maximum number of NAKs a router can possibly get after a packet loss is bounded by the downstream interfaces

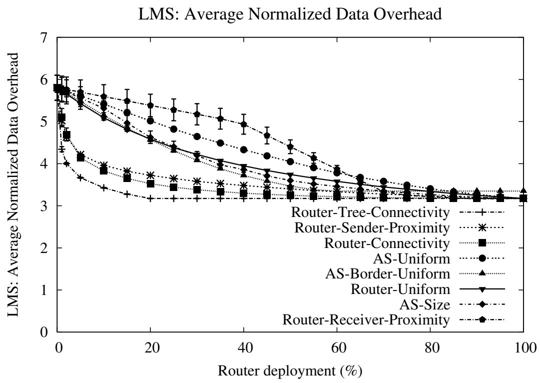

Fig. 8. LMS Average Normalized Data Overhead

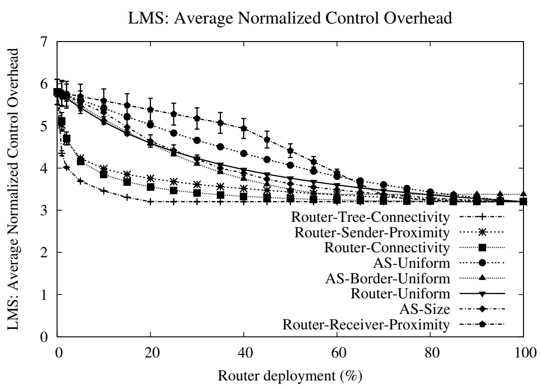

Fig. 9. LMS Average Normalized Control Overhead

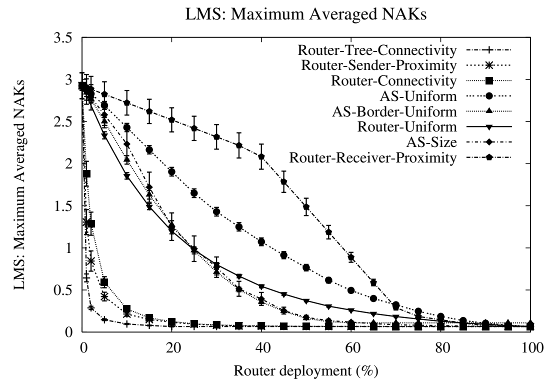

Fig. 10. LMS Maximum Averaged NAKs

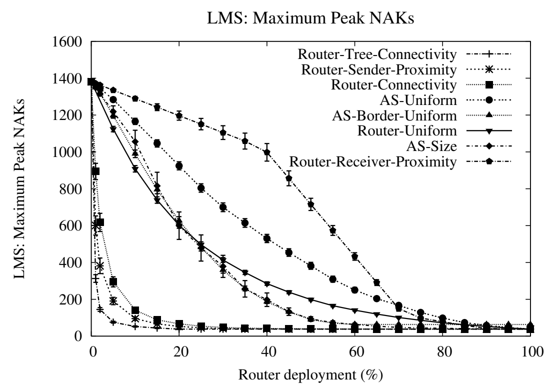

Fig. 11. LMS Maximum Peak NAKs

of the router. In our simulations, the maximum downstream interfaces a router may have is approximately 33 across the 50 different multicast groups.

D. *Simulation Results for LMS*

Figure 8 and Figure 9 show the Average Normalized Data Overhead and Control Overhead for LMS. Compared with PGM, both types of overhead start with much lower values in LMS. 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 because in LMS in most cases repliers send the repairs, not the sender, therefore routers near the sender become less important. The trends for the second and third categories are similar to PGM.

Figure 10 shows the maximum averaged NAKs in LMS. While higher than PGM, this overhead also appears to be negligible.

Figure 11 shows the Maximum Peak NAKs 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. 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.

In summary, the deployment strategies we study exhibit similar behavior for both PGM and LMS. However, the impact of incremental deployment appears very different. In PGM the impact can be felt in terms of data and control overhead, because in the absence of PGM routers, NCFs and repairs cannot be scope-limited, therefore they are multicasted to a large 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*.

V. Simulation Results Sensitivity

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

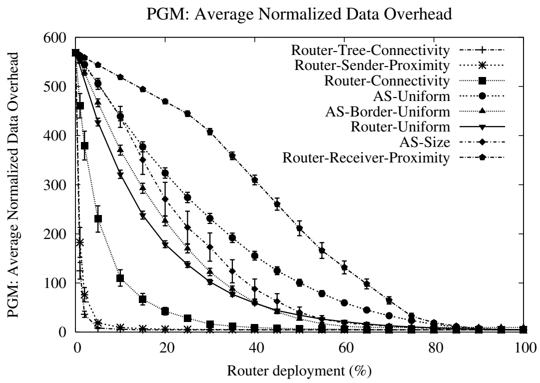

Fig. 12. Receiver population sensitivity: PGM Average Normalized Data Overhead with 1% multicast group size

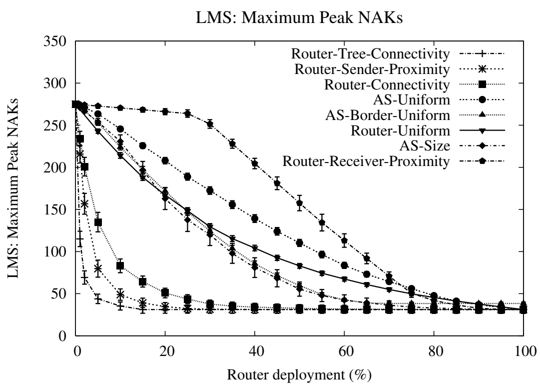

Fig. 13. Receiver population sensitivity: LMS Maximum Peak NAKs with 1% multicast group size

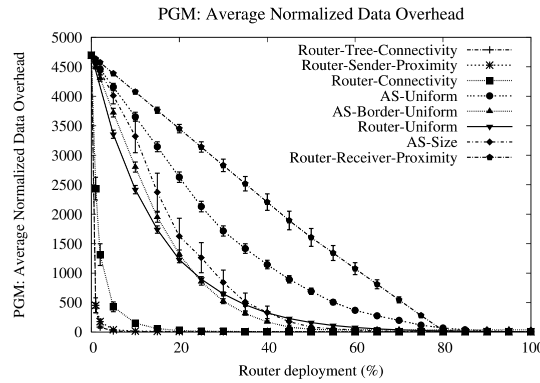

Fig. 14. Receiver population sensitivity: PGM Average Normalized Data Overhead with 10% multicast group size

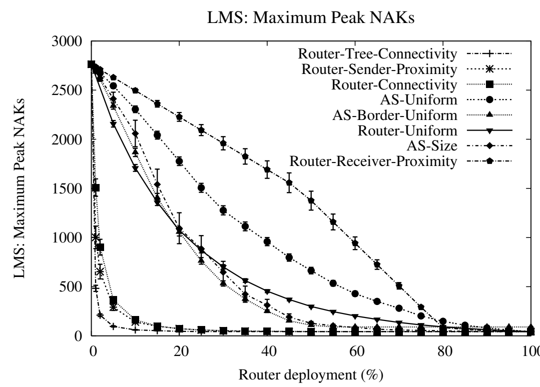

Fig. 15. Receiver population sensitivity: LMS Maximum Peak NAKs with 10% multicast group size

A. *Multicast Group Size*

Figure 12 and Figure 14 show the PGM results when the multicast group size is 1% and 10% of the network size, respectively. The results for LMS are in Figure 13 and Figure 15. The receivers and the sender are again chosen at random. From those results we can make the following observations: (a) for both protocols the zero-deployment overhead is, as expected, proportional to the group size; (b) for both protocols, smaller group size seems to decrease the gaps between the three strategy categories; and, (c) Multicast-Tree-Aware strategies (such as Router-Tree-Connectivity and Router-Sender-Proximity) seem to be impacted by group size more than Network-Aware strategies. In general, however, the overall behavior seen earlier appears to persist.

B. *Receiver Placement*

To study the impact of receiver placement, we consider two extreme receiver placement models as defined in [8], namely *extreme affinity* and *extreme disaffinity*. The extreme affinity model selects the receivers as close to each other as possible, while the extreme disaffinity model selects the receivers as far away from each other as possible. The particular algorithm for receiver selection we use is given in [9] 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 [9], we use $\beta = 15$ and $\beta = -15$ for extreme affinity and disaffinity respectively.

Figure 16 and Figure 17 show the results for extreme affinity for each protocol. Figure 18 and Figure 19 show the results for extreme disaffinity. In all these settings, the multicast group size is 5%, and the sender is placed on 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. As a result, there are fewer routers in the multicast tree, and routers close to the receivers have higher fanout, which gives them more control over the data recovery. In contrast, the AS-Size strategy

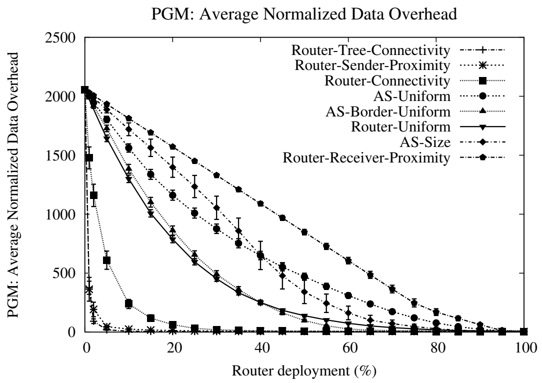

Fig. 16. Receiver placement sensitivity: PGM Average Normalized Data Overhead with the extreme Affinity receiver placement

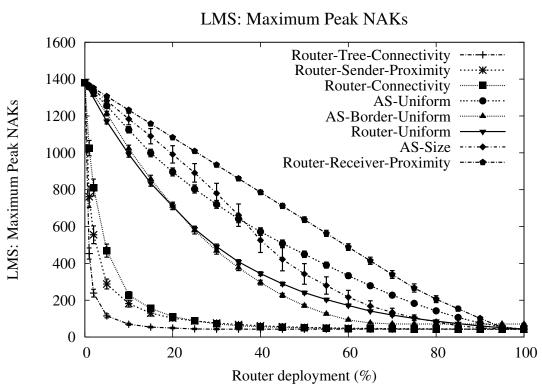

Fig. 17. Receiver placement sensitivity: LMS Maximum Peak NAKs with the extreme Affinity receiver placement

appears to be worse in the extreme affinity model as opposed to the extreme disaffinity model. A possible explanation is that in the extreme affinity model all receivers may be clustered in some 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 routers on the topology. The sender location, however, plays a significant role for the AS-Size deployment. Figure 20 and Figure 21 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 routers. The reason for improvement is that by deploying routers in the largest AS at the beginning, we are enabling routers close to the sender, and getting the same kind of benefits as the Router-Sender-Proximity strategy.

D. PGM NAK Back-off Timer Interval

The NAK back-off timer interval in PGM is an important parameter because it controls the effectiveness of NCFs in suppressing NAKs. In the previous experiments we assumed

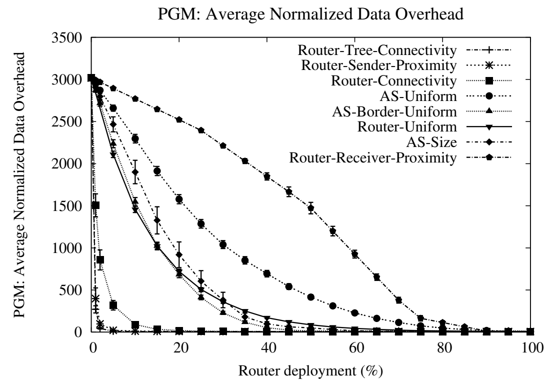

Fig. 18. Receiver placement sensitivity: PGM Average Normalized Data Overhead with the extreme Disaffinity receiver placement

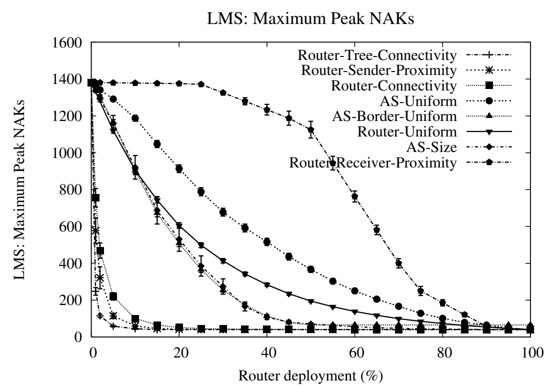

Fig. 19. Receiver placement sensitivity: LMS Maximum Peak NAKs with the extreme Disaffinity receiver placement

optimal NAK suppression for PGM. That gives us a lower bound on the PGM overhead. A good estimate of the back-off interval, however, is important to avoid undue impact on recovery latency. PGM proposes an algorithm to adjust the back-off interval based on implosion. Evaluating this algorithm is beyond the scope of our work. Instead, we are interested in how quickly implosion subsides with deployment given a reasonable estimate of the back-off interval. Our evaluation includes two steps. First, we estimate the back-off interval under the worst case scenario of zero deployment. Second, we study how implosion subsides as deployment increases. Figure 22 shows the results of the first step, namely NAK implosion as a function of the back-off interval (measured in the unit of the worst round-trip time between any receiver and the sender). From this graph we somewhat arbitrarily select the value of 4*RTT as a compromise, and using this value we run simulations for all deployment strategies. We note that the actual value is less important because we are only interested in how different deployment strategies affect implosion. Our results show that the merits of all strategies remain virtually unchanged, as shown in Figure 23.

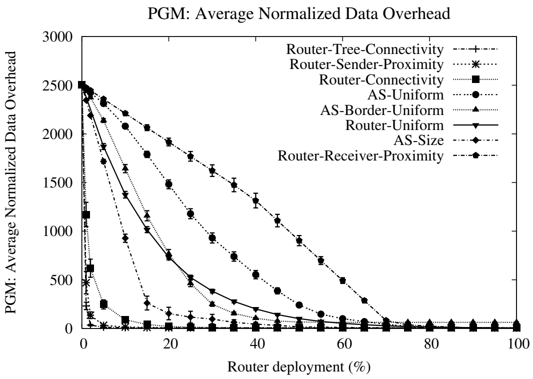

Fig. 20. Receiver placement sensitivity: PGM Average Normalized Data Overhead with the Largest-AS sender placement

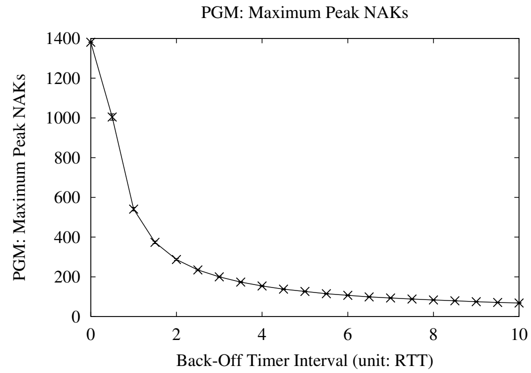

Fig. 22. PGM back-off timer interval sensitivity: PGM Maximum Peak NAKs with different back-off timer intervals

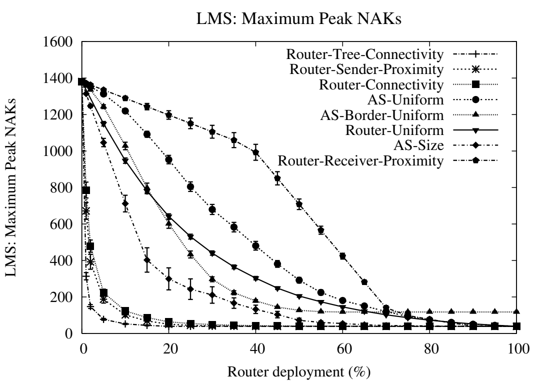

Fig. 21. Receiver placement sensitivity: LMS Maximum Peak NAKs with the Largest-AS sender placement

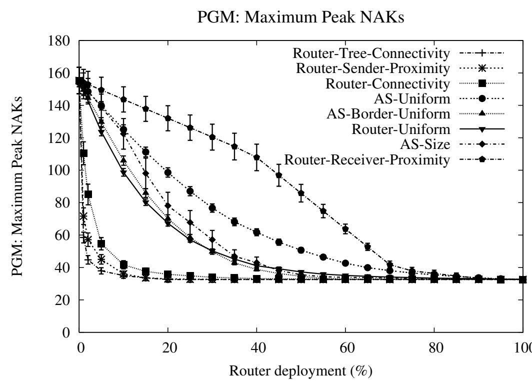

Fig. 23. PGM back-off timer interval sensitivity: PGM Maximum Peak NAKs with 4 * worst RTT as the back-off timer interval

VI. Related Work

While there exists a large body of work that proposes new network services, the subject of incremental deployment has typically received less attention. Most proposals sketch incremental deployment plans, but very few carry out systematic evaluations.

In addition to the earlier work on incremental deployment of LMS [4] described in Section I, the other related work we are aware of is a study in the context of Active Reliable Multicast(ARM) [10]. 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.

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 [11], TMTP [12], SRM [13], TRAM [14], to name a few.

Router-assist schemes scale better than end-to-end schemes because they (a) help efficiently build a hierarchy congruent with the underlying multicast routing tree, (b) provide controls for fine-grain multicast of retransmissions, and (c) simplify applications by eliminating the need for heuristics to maintain topology-related state, such as the setup and management of local groups.

In addition to PGM [1] and LMS [2], which are the subjects of this paper, other router-assist schemes include the following: Search Party [15] is a scheme inspired by LMS, which adds robustness by using *randomcast* to distribute a request randomly among receivers rather than just a single replier. Addressable Internet Multicast (AIM) [16] is a scheme that assigns labels to routers on a per-multicast group basis, and routes requests and repairs based on these labels. OTERS [17] and Tracer [18] both employ mtrace utility to build congruent hierarchies. Finally, Active Error Recovery (AER) [19] is a scheme 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 diffcult 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 in the past to systematically study deployment strategies and their impact on performance.

In this paper we took a first step to define a framework for evaluating incremental deployment of router services. Given the lack of information about how deployment occurs in the real world, our framework adopted a blend of plausible and representative deployment strategies and a mix of network-specific and host-specific performance metrics. Our case study was router-assisted reliable multicast. We considered two protocols, namely PGM and LMS, and used numerical simulation to evaluate their performance under partial deployment. In our evaluation 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 value of the PGM back-off timer interval.

Our results show significant difference between deployment strategies, suggesting that our framework is capable of capturing and characterizing their performance, clearly demonstrating careful study of incremental deployment can no longer be ignored. Our framework 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, needing 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 such services.

One of our future works is to further improve our simulation of the PGM protocol, including its dynamic adjustment of the NAK back-off timer interval and the proper setting of related parameters. Although this work has provided a lower bound on the network overhead and the implosion problem in PGM, we believe further improvement is needed to better understand the PGM performance. Another on-going work is to study possible improvements to PGM and LMS protocols so that they can deal better with sparse deployment. As we have seen, when there is little assistance from the network, LMS performs poorly in terms of NAK implosion, while PGM incurs high network overhead. A combination of these two protocols may provide possible solutions to solve these problems. Finally, we are planning to apply our framework to study the incremental deployment of other router-assisted services, such as security services that rely on the support of routers.

ACKNOWLEDGMENT

This paper has benefited greatly from the valuable suggestions of Dr. Ramesh Govindan. The authors would also like to thank the anonymous referees for their constructive comments.

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] 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.

[4] 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.

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

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

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.