Incremental Deployment of a Router-assisted Reliable Multicast Scheme

Christos Papadopoulos
University of Southern California
941 W. 37th Place, SAL 238
Los Angeles, CA 90089
christos@isi.edu

Emmanouil Laliotis
University of Southern California
941 W. 37th Place
Los Angeles, CA 90089
laliotis@usc.edu

ABSTRACT

Recently, several schemes have been proposed that address the problem of scalable reliable multicast using router assistance. These include LMS, PGM, AIM and OTERS, among others. While these schemes (arguably) achieve significantly better performance than non-assisted schemes, a thorny issue remains: are these schemes incrementally deployable? More specifically, what is the performance impact of partial deployment to the application and the network? What is the best strategy to deploy such schemes? These questions, while very important, have not been addressed yet.

In this paper we investigate the performance of Light-weight Multicast Services (LMS) under various deployment strategies. Our results are preliminary, mostly due to lack of established strategies for incremental deployment of such services. First, we propose a method to incrementally deploy LMS which has minimal impact on router processing and state. Then, we investigate the performance of LMS for various basic deployment strategies, including random, core, stub, and border router deployment, and for different topologies, including random and transit-stub topologies. We also examine node v.s. path deployment. Our evaluation is carried out using the ns simulator and topologies generated with GT-ITM. Our results show that there are significant differences among the various deployment strategies with random being the worst. Finally, we discuss our observations in the context of other router-assisted schemes and identify common issues.

Keywords

Multicast, reliable multicast, incremental deployment.

1. INTRODUCTION

Influenced by the success of the unicast IP model, IP Multicast has adopted a simple "broadcast within a group" delivery service. While this service model is robust and flexible, its inherent broadcast nature coupled with group anonymity inhibit the scalability of some important higher level services like reliable multicast. In the past ten years many solutions have been proposed which attempted to solve such problems at the application layer (SRM[10], RMTP[11], etc.).However, these solutions either do not scale well, or are inflexible (e.g., they employ static hierarchies). Recently, a new breed of solutions has emerged which extends the multicast service model by utilizing some assistance from the network (LMS[2], PGM[5], AIM[3], OTERS[8], etc.). These solutions have been shown to perform significantly better than pure application-level solutions. Unfortunately, the thorny issue of deployment of such schemes still remains unresolved.

The Internet today has evolved into a huge commercial network that bears almost no resemblance to the Internet of the last decade. The Internet's size coupled with a highly distributed administration model, makes changes in the network layer very difficult to deploy. However, given a strong enough need, such changes may still be possible, as witnessed by several ongoing efforts to introduce quality of service (Diffserv[12]), reservations (RSVP[13]) and even a new network protocol (IPv6[15]). However, it has become increasingly clear that any such change in the network *must* be accompanied by a viable incremental deployment plan. It is widely believed that any proposed scheme that cannot be incrementally deployed will most likely never be adopted.

The issue of deployment is especially complicated in reliable multicast schemes, due to the strong coupling of deployment and efficiency. Reliable multicast schemes leverage off the location of the routers in the network to gain an edge over application-level schemes. For example, routers double as request aggregators at branching points to eliminate implosion; or routers are used as launching pads to initiate retransmissions that precisely target the appropriate subset of receivers. The efficiency of these operations hinges on the ability of each scheme to identify routers near the loss. With partial deployment this ability diminishes, and with it the efficiency of router-assisted schemes. An instance of this problem is shown in Figure 1. Here a loss inside the non-enhanced cloud cannot be dealt with efficiently. The reason is that lack of aggregation points inside the cloud cause many requests to be directed to the next available aggregation point near the loss (R0 in this example), which may be located far from the loss; and since this point must also used as a launching pad for the retransmission, duplicates may be delivered to several unsuspecting receivers.

Perhaps the easiest way to deploy router-assisted schemes is to bundle them with other new services like IPv6 on experimental overlays (e.g., 6-Bone [14]). In such a case there is no incremental deployment issue to consider. However, such deployment is of limited benefit to existing applications, until they also move to the

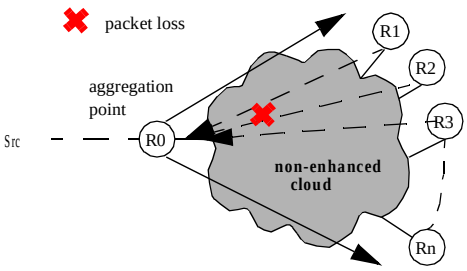


Figure 1: Loss of efficiency due to partial deployment

new overlay. Thus, there is a strong incentive to devise deployment methods that work on existing networks.

Evaluating the performance of a router-assisted reliable multicast scheme under partial deployment is very complex, and to the best of our knowledge has not been addressed before[1]. In this paper we explore the performance characteristics of incremental deployment of one reliable multicast scheme, namely LMS. We discuss the relevance of our findings to other schemes in our conclusions. Our aim is not to propose a comprehensive deployment plan, but rather to better understand the factors that affect performance under partial deployment.

While it may appear premature to investigate incremental deployment of router-assisted multicast services when IP multicast itself is not yet universally deployed, we note that such services significantly enhance the value of IP Multicast and may therefore strengthen the case for its deployment.

This paper has two parts. In the first part we propose enhancements to LMS to address incremental deployment, an issue which was lightly touched in the original description. In the second part, we present simulation results of incremental deployment of LMS. However,

The paper is organized as follows. In Section 2 we give a brief overview of LMS; in Section 3 we describe how LMS can be incrementally deployed; in Section 4 we describe the deployment strategies we study and the metrics we use; in Section 5 we present our simulation results; and finally in Section 6 we discuss the results.

2. BRIEF DESCRIPTION OF LMS

*Light-weight Multicast Services (*LMS) is a highly scalable and flexible router-assisted scheme created to address the problem of scalable reliable multicast. LMS, however, has other applications besides reliability which we do not investigate in this paper. In this section we give a brief overview of LMS. A more detailed description can be found in [9].

Most of the problems associated with reliable multicast, namely implosion and exposure[2], minimizing recovery latency and adapting to dynamic groups, could be efficiently solved if routers performed transport level operations. Such a solution, however, not only violates the end-to-end principle but also scales poorly. LMS attempts to solve both these problems while maintaining the efficiency of a router-based solution by allowing routers to select surrogates at the network edges which perform transport level processing on the routers' behalf, as shown in Figure 2. In order to

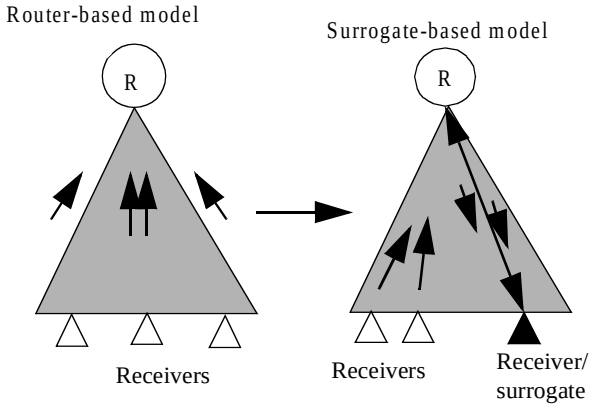


Figure 2: The LMS surrogate model

push transport processing to the edges, LMS defines a set of forwarding services at the routers that help steer control messages to the surrogates. These forwarding services are very simple and have been shown to have processing overhead on par with regular multicast packet forwarding[9]. In addition, since the forwarding is done at the network layer, these services do not need to peek into higher layers and thus avoid layer violation. Routers enhanced with LMS, in addition to their regular duties perform the following additional tasks:

1. **Replier (surrogate) selection:** each router with two or more downstream interfaces associated with a particular <S, G> entry, marks *one* of these interfaces as the *replier interface*. Intuitively, a replier interface is the interface the router presumes to lead towards an edge surrogate willing to perform transport level operations on behalf of this router. Surrogates are required to send advertisements to the network, and the replier interface is selected by comparing replier advertisements received from multiple interfaces. By default, each router selects the upstream interface as its replier interface while waiting for advertisements. Replier state is *soft state*, meaning that repliers have to refresh it periodically. This provides robustness and guards against replier and link failures.
2. **NAK forwarding:** this is the first of the two new forwarding services LMS introduces to the routers. NAKs are processed hop-by-hop and forwarded as follows: if a NAK arrives at the router from the replier interface, the NAK is forwarded upstream; if the NAK arrives either from the upstream inter-

1.PGM includes mechanisms that account for incremental deployment, but the performance of PGM under these conditions has not been evaluated yet.

2.Implosion occurs when a large number of redundant messages (e.g., NAKs) overwhelm the network, the sender or the receivers. Exposure happens when receivers receive duplicates which result from inaccurate targeting of recovery-related messages (e.g., retransmissions).

face or a non-replier interface, the NAK is forwarded to the replier interface. In the latter case (a NAK arriving on a non-replier interface), the router becomes the *turning point* for that NAK. The router inserts in the NAK the address of the incoming and outgoing interfaces, which together form the *turning point information*, which is carried by NAKs to the replier.

3. **Directed multicast (dmcast):** this is the second forwarding service introduced by LMS. It is used to deliver retransmissions and works as follows: a replier who is about to retransmit data to satisfy a previously received NAK creates a multicast packet containing the data and addresses it to the group. The replier encapsulates the multicast packet into a unicast packet, which is unicast to the turning point router. Included in the packet is the remaining information received with the NAK, namely the address of the interface the NAK originally arrived at the turning point router. When the router receives the unicast packet, it decapsulates the multicast packet and multicasts it on the specified interface. The result is that the entire subtree rooted at that interface receives the retransmission.

A sequence of recovery steps using LMS is depicted in Figure 3.

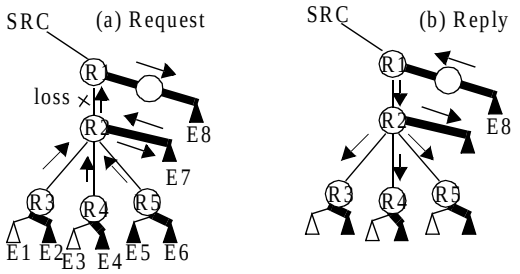


Figure 3: LMS Summary

Assume that loss occurs on the link between R1 and R2. Replier links are in bold. Endpoints E1 through E7 detect the loss. Following detection of loss the following events take place:

- E7 sends a NAK, which R2 forwards to R1 because E7 lies on R2's replier link.
- E1 sends a NAK which is forwarded by R3 to E2. Similarly, NAKs from E3 and E5 are forwarded to E4 and E6 by R4 and R5, respectively.
- The NAK from E2 is forwarded to R2, because E2 is on R3's replier link. Similarly, the NAK from E4 and E6 are also forwarded to R2.
- R2 forwards NAKs from E2, E4 and E6, to E7, which ignores the NAKs since it does not have the data (but has requested it).
- The NAK from E7 reaches R1.
- R1 forwards the NAK towards E8, which has the requested data.
- E8 creates a multicast message containing the reply. E8 encapsulates the message in a unicast message and sends it to R1 (the request's turning point).
- R1 decapsulates the multicast message and multicasts it on the link leading to R2. From that point on, all downstream routers and endpoints treat the reply as a regular multicast message coming from the source.

To summarize then, LMS employs three important concepts: replier selection, steering of requests to repliers and establishing turning points, and directed multicast. These concepts work together to enable receivers to construct an efficient recovery mechanism.

2.1. Exposure

While the example above showed a scenario where the retransmissions reached precisely the receivers that lost the original data, it is possible to target replies incorrectly and expose receivers to duplicates. An example is shown in Figure 4. Here a request from

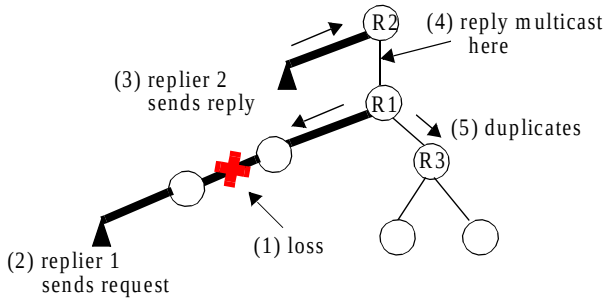


Figure 4: Exposure in LMS

replier 1 reaches replier 2 which sends a directed multicast to R2, which in turn multicasts the reply on the downstream link leading to R1, causing duplicates on the subtree rooted at R3. The exposure resulting from this problem has been shown to be very low on the average [9] and does not impact correctness. However, it may lead to excessive duplicates if a replier branch is particularly lossy. With LMS this problem is mitigated by selecting the branch that advertises the least loss. For example, the path to replier 1 was lossy, R1 would select a replier from the branch leading to R3, if this branch advertises lower loss.

A way to eliminate exposure completely[1] at the cost of adding an extra hop to the retransmission is shown in Figure 5. To eliminate exposure, NAKs include the requestor's address and specify that the reply should be unicast to the requestor rather than the turning point. Then, only if the requestor receives more requests a DMCAST is sent to the remaining part of the subtree. This can be done by the requestor sending a dmcast to its turning point router requesting a multicast to all downstream links. Note that the delay introduced by the additional hop may be small if the unicast path from the requestor to the replier is shorter than the multicast path.

3. INCREMENTAL DEPLOYMENT OF LMS

In this section we describe the changes required in LMS to facilitate incremental deployment. We describe a procedure that creates a LMS overlay (a virtual subgraph containing only LMS-

1.With this method the replier will receive its own retransmission which technically counts as a duplicate.

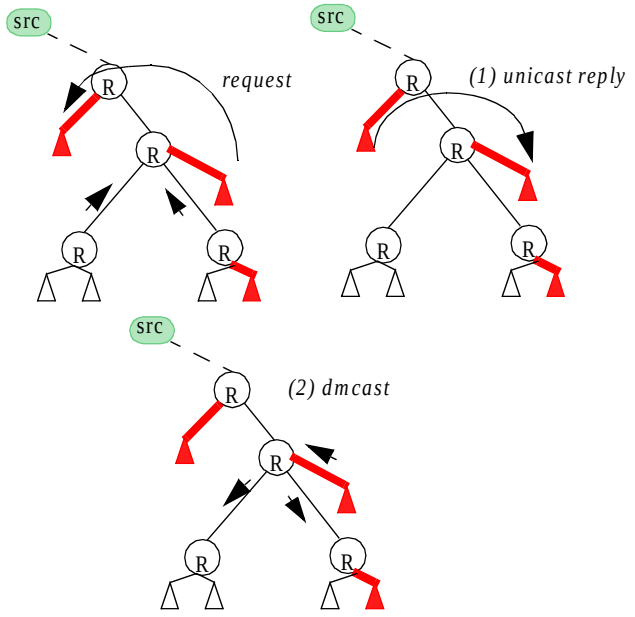


Figure 5: Using a two-step DMCAST to eliminate exposure

capable nodes) and some changes in the original procedures for replier selection, NAK and dmcast handling necessary to cope with incremental deployment. Our assumptions are as follows: without loss of generality, we assume the existence of a single-source multicast group formed using a routing protocol that produces either a source-based tree (e.g., DVMRP or PIM-DM), or a unidirectional shared tree (e.g., PIM-SM). We define a partial LMS deployment as a deployment where the source and *all* the receivers in the group are LMS enabled, but only *a subset* of the on-tree routers support LMS. We do not study the case where some of the receivers or the source are not LMS-enabled.

3.1. Establishing the LMS Overlay

To establish the LMS overlay each source periodically multicasts a control message called a *Source Path Message*[1] (SPM) to the group. When transmitted by the source, a SPM carries the IP address and port number of the source. SPMs also carry the IP Router Alert option [16] which forces *all* routers (LMS or otherwise) along the multicast tree to examine them. Non-LMS routers ignore SPMs and allow them to continue downstream without any further action. LMS routers, upon receiving a SPM take the following actions: first, the address of the upstream LMS node (which may potentially located across a non-LMS cloud) is recorded and filed with the appropriate <S, G> entry. Then, LMS routers insert their own IP address[2] in the SPM overwriting the existing entry, and re-insert the SPM in the multicast tree. After passing through all intermediate routers, SPMs finally reach the receivers. At that point each LMS router and LMS receiver has established a pointer to its upstream LMS hop along the reverse *multicast* path to the source. This process is depicted in Figure 6.

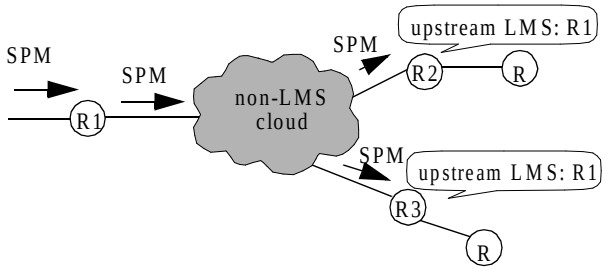


Figure 6: Establishing reverse paths using Source Path Messages (SPMs)

The overlay established using *reverse path pointers* is sufficient to enable NAKs to be forwarded upstream towards the source. In LMS, however, NAKs need to be turned around at the turning points and forwarded toward repliers. Reverse path pointers are clearly not sufficient; LMS requires to set-up a path towards repliers which occurs during replier selection, as described next.

3.2. Replier Selection

Replier selection in a LMS overlay is done as follows: initially receivers wait before sending the first replier advertisement until they receive at least one SPM. Upon reception of the first SPM, receivers begin periodic replier advertisements, but in contrast to full deployment, advertisements are unicast to the first upstream LMS router. In addition, the replier advertisement contains the address of the receiver. LMS routers that receive multiple advertisements select the best replier following the usual LMS rules. Each router then unicasts the selected replier's advertisement to the upstream LMS router, after replacing the replier address with its own address[3]. This process repeats until all routers have selected a replier, establishing *forward paths* towards repliers as shown in Figure 7. Once all LMS routers have selected repliers the LMS overlay is complete. The mechanisms for establishing reverse paths and replier selection were included in our simulations.

3.3. Handling Requests (NAKs)

With full LMS deployment it is beneficial for receivers to multicast NAKs in order to suppress other NAKs from other receivers on the same LAN. With partial deployment this is no longer viable as local routers may not be LMS-enabled[4]. To overcome this problem, LMS receivers *unicast* NAKs to the upstream LMS router; local NAK suppression is done via a separate

1. The term was originally introduced in the PGM draft. The term *path message*, was originally used in RSVP. The functionality of SPMs in LMS and PGM is similar.

2. A router has multiple IP addresses (one per interface). SPMs sent downstream carry the IP address of the interface they are forwarded on.

3. As before, the address inserted in the message is the address of the interface the message is transmitted on.

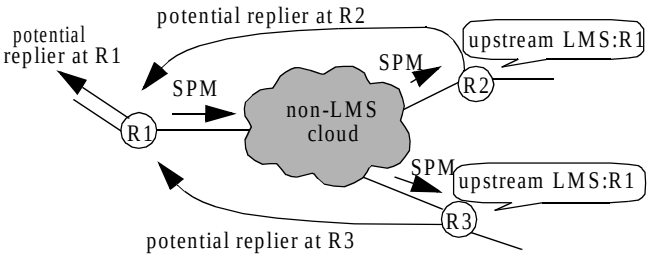


Figure 7: Establishing forward paths to repliers

local multicast. Upon reception of a NAK, the upstream LMS router applies the standard forwarding rules to determine how to handle the NAK. Once a decision is made, the router unicasts the NAK either to the upstream LMS hop or the previously selected replier hop. The router identifies NAKs coming from its replier based on the source address of the NAK, not the incoming interface. The turning point information inserted in a NAK, however, remains unchanged, and consists of the addresses of the NAK's incoming and outgoing interfaces.

A drawback of the above approach is that repliers are now vulnerable to NAK implosion. The problem is depicted in Figure 8

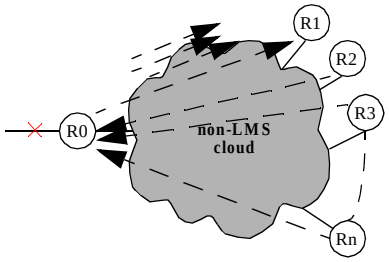


Figure 8: Replier experiences NAK implosion if turning point has large fan out

and occurs when a large number of LMS routers (or receivers) peer with the same upstream router. Note that this problem is similar to the problem that arises with routers with large fan-out. The solution to this problem was briefly touched in [9]. We repeat it here and expand on it with some further improvements.

To avoid implosion at the replier, the router may partition its links into subgroups and select a replier for every subgroup, as shown in the example in Figure 9. Here requests from links in subgroup *D* go to replier *d*, but requests from replier *d* go to replier *c*, requests from replier *c* go to replier *b* and so on. Requests from replier *a* are forwarded upstream. In this scenario, replier *a* is called the router's *primary* replier and repliers *b*, *c*, and *d* are called

4. A local multicast may not reach a LMS router. If receivers know that local routers are LMS-enabled, then they can multicast NAKs. Receivers can query local routers by sending a message to the all-routers multicast address.

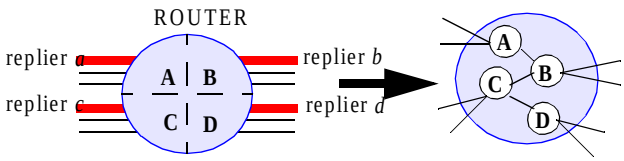


Figure 9: dealing with routers with large fan-out

auxiliary repliers. By partitioning links this way, the maximum number of requests a replier can receive is significantly reduced.

To maximize efficiency, once the router determines that *n* auxiliary repliers are needed the router selects the *n* best repliers from the replier advertisements it receives. The size of the auxiliary replier set (*n)* is no longer determined by the number of downstream interfaces, but is a function of the number of distinct replier advertisements that were received. To further increase efficiency, the router directs the first NAK from a non-replier peer to the primary replier as follows: we define a *quiet period* as a period during which the router has seen no NAKs for a particular <S, G> entry. Upon receiving a NAK after a sufficiently large quiet period (to be determined by each router), the router forwards the first NAK to the primary replier; if more NAKs arrive and the NAK arrival rate exceeds a specified value, the router distributes these NAKs to the appropriate auxiliary repliers with the restriction that a NAK is never sent back to the replier it came from.

The above solution eliminates the danger of implosion at the replier but not at the router, and may still allow transmission of multiple NAKs which wastes bandwidth. A solution that reduces the number of redundant NAKs is to use the replier's NAK to suppress NAKs from other receivers. To accomplish this, all receivers agree to observe a back-off interval after detecting loss and before sending a NAK. The LMS router, not only forwards the replier request upstream, but subcasts it to all downstream links which suppresses requests from other receivers[1]. The trade-off is that latency is increased due to the back-off interval observed by receivers. To speed up recovery some heuristics can be used, for example a receiver that knows with a certain confidence that it is a replier (e.g., it has initiated the most recent reply), it may keep its back-off interval short or forgo it altogether. Note that we did not evaluate this approach in our simulations.

3.4. Handling Retransmissions

There are two ways to handle retransmissions with incremental deployment: the first is to use the original LMS mechanism as is, which trades simplicity for exposure; the second is to target dmcasts better by performing individual dmcasts to downstream peers at the turning point, which reduces exposure at the expense of increased retransmission traffic. We describe the problems with both methods next.

With the original method, the replier unicasts the reply to the turning point router and the router multicasts it on the NAK's incoming interface. This may result in increased exposure as shown

1.We should be careful not to suppress all requests in cases where recovery is done in two steps as described in section 2.1

in Figure 10. In this example, loss occurred just upstream of R3,

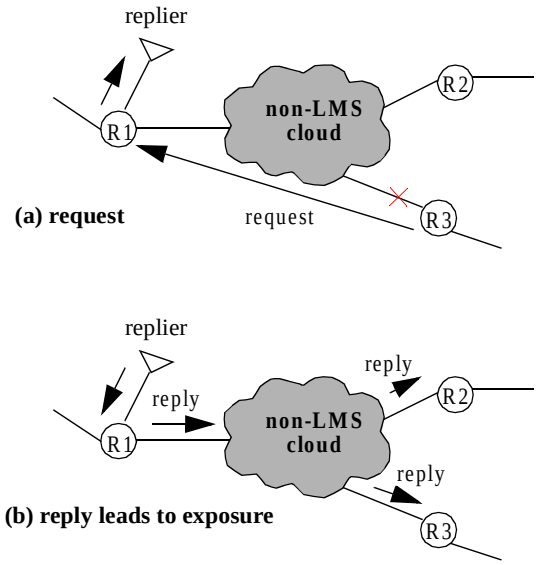


Figure 10: Increased exposure caused by non-LMS clouds

causing a request to be unicast from R3 to R1, where it is directed to R1's replier. The reply is multicast by R1 on the request's incoming link, thus reaching both R2 and R3. The problem is that with incremental deployment LMS loses its targeting ability when no turning points exist near the location of loss. When this happens, replies are multicast to a larger subtree than needed, leading to increased exposure. Note the dependence of this problem on the location of loss: if loss had occurred upstream of R1, there would be no exposure. This version of LMS is the one we use in our simulations.

The second solution is to allow dmcasts to be unicast to the first LMS router downstream the turning point, in our example R3. This can be done if each LMS router marks itself as the turning point when forwarding the replier's NAK upstream. In the above example, R3 would be the last router modifying the turning point information and the replier would unicast the reply to R3 instead of R1. The disadvantage of this approach is that while it eliminates exposure in the previous case where loss occurred just upstream of R3, if loss had occurred just downstream of R1 then the replier would have to perform individual dmcasts to R2 and R3. Thus, this approach is better suited for cases where routers have more information about topology, for example, if R3 knows that there is a large non-LMS cloud between itself and R1. A reason, however, not to use this approach is that it potentially sends multiple retransmissions over an already congested link.

A more general approach combines the above approaches and works as follows: a LMS router keeps an estimate of the number of peers on each downstream interface by counting replier advertisements. When NAKs are received, the router at the turning point inserts the address of the incoming interface, the address of the peer that sent the NAK (the router does not have to remember the peer address since it is carried in each unicast request), and the router's estimate of the number of peers on that interface. It is left up to the replier to decide how to respond based on this information. For example, the replier may chose to respond with a single dmcast at the turning point's interface if NAKs are received from many of the router's peers; or if the number of NAKs is small, the replier may send a separate dmcast to each requestor. The replier can make this decision by delaying the retransmission for a short time in order to gather a sufficient number of NAKs. These choices are shown in Figure 11.

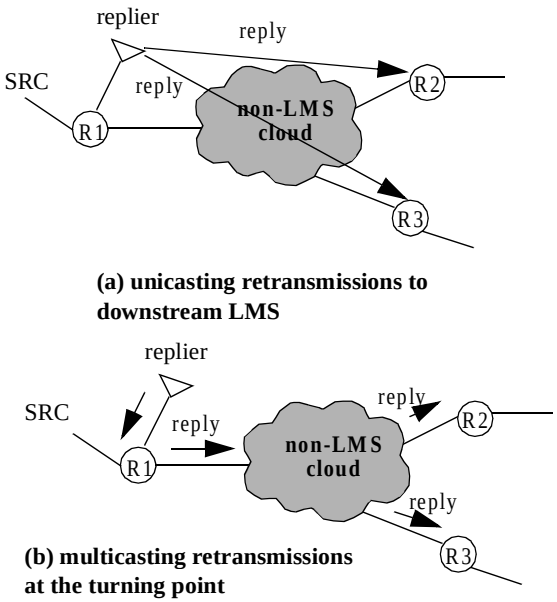


Figure 11: Replier options for delivering retransmissions

The important point of this discussion is that the approach we propose for incremental deployment of LMS does not require routers to make any decisions, but maintains the intelligence at the edges where it can be easily modified. The changes proposed at the routers to accommodate deployment are small and we do not expect them to significantly impact performance.

4. METRICS AND DEPLOYMENT STRATEGIES

We used the same metrics to evaluate the performance of LMS during incremental deployment as the ones used for full deployment. In addition, we added another metric that captures the NAK overhead at the source and the repliers. As we described earlier, incremental deployment may lead to increased NAK traffic and potentially to implosion.

4.1. Metrics

Four metrics were used to evaluate the performance of LMS as a function of deployment. These are: (a) average normalized latency, (b) average exposure, (c) peak NAK load at the source, and (d) peak NAK load at repliers. The definitions of these metrics are shown in Figure 12.

Normalized Latency:	$\frac{\text{Recovery Latency}}{\text{RTT to Source}}$
Exposure:	$\frac{\text{Total Duplicates}}{\text{Number of receivers}}$ / **Total Drops**
Peak NAK load: at source	$\frac{\text{Max \# of requests at source}}{\text{Total \# of receivers}}$
Peak NAK load at receivers:	$\frac{\text{Max \# of requests at receivers}}{\text{Total \# of receivers}}$

Figure 12: Evaluation metrics

The first two, namely normalized latency and exposure, are the original metrics used in our previous evaluation of LMS. Exposure measures the success of local recovery in LMS, and can be thought of as the average number of duplicates a receiver sees as a result of a loss anywhere in the tree. The last two metrics capture the maximum number of NAKs as a function of the receiver size at the source and the repliers during incremental deployment.

4.2. Deployment Strategies

The choice of deployment strategy is a complicated one because it depends not only on technical factors but also on policy issues. Different Autonomous Systems (AS's) may adopt different deployment strategies based on economic factors, customer support, security, and different business models. In this work we limit our evaluation to simple deployment strategies in the hope of achieving generality. *We make absolutely no claims* that the simple deployment strategies we present here are sufficient for the deployment of router-assist schemes in real networks.

For this work we define a deployment strategy as the set of rules that specify which routers will be enabled with LMS. In our experiments we assume that deployment is progressive, meaning that we typically start with a low density of deployment and increase density according to the rules of the particular strategy until all routers are enabled. Although absolute numbers are important, here we are more concerned with identifying the *trends and characteristics* of a particular strategy. Next, the strategies we have chosen to study the performance of LMS are described. They are also summarized in Figure 13.

Random node: routers to be enabled with LMS are chosen with uniform probability. We investigate how performance is affected as the percentage of LMS routers increases. This is, admittedly, an unrealistic strategy, but worth studying because of its simplicity.

Random Path: a subset of receivers is chosen with uniform probability and LMS is deployed on the path from each receiver to the source. This strategy is probably not very realistic on wide area networks, but may be feasible within the same administrative domain. A case where this strategy may be used is intra-domain deployment, when an ISP offers LMS to a few high-paying customers first.

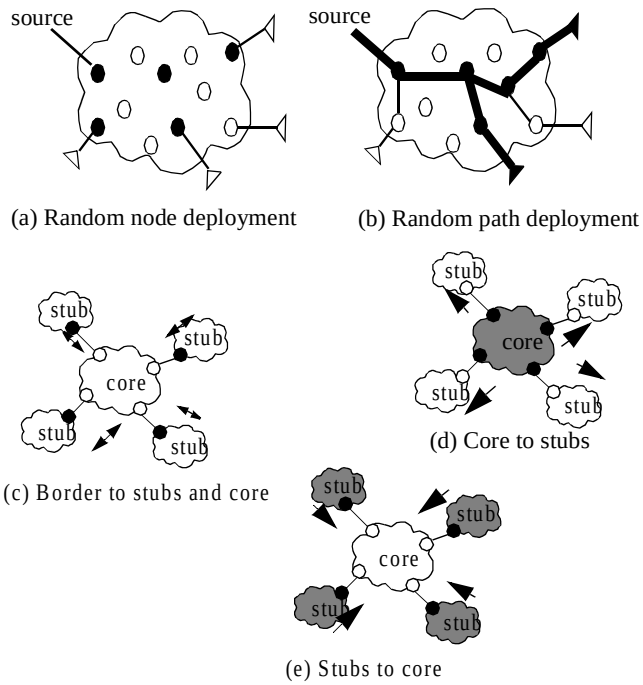


Figure 13: LMS deployment strategies

Border to stubs and core: this strategy applies to transit-stub topologies. We assume that in such topologies deployment will include at least the border routers and LMS is uniformly added to the remaining routers.

Core to stubs: this strategy also applies to transit-stub topologies. LMS starts with the core routers LMS-enabled and uniformly moves out to the stubs. This is probably an unrealistic strategy for the wide-area Internet, but may be adopted by small, or newly formed ISPs.

Stubs to core: this is another strategy that applies to transit-stub networks. This strategy is the opposite of the above: it assumes that all the stubs are LMS enabled, and deployment uniformly moves to the core. This strategy attempts to capture the case where customers deploy may LMS before their ISPs.

5. SIMULATION RESULTS

Our simulations used the ns simulator [6] and topologies generated by GT-ITM [7]. We used two types of topologies: flat topologies of 250 nodes and uniform edge connectivity of 10% which yields an average fan out of approximately 25 (we will refer to these as **random topologies**), and hierarchical, **transit-stub topologies** of 250 nodes, consisting of 1 transit domain with 25 nodes and edge probability of 60%, each of which has one stub domain of 9 nodes attached. Edge probability for stubs is 42% and the average fan-out of transit-stub topologies was 4.55. We generated 10 random and 10 transit-stub topologies. Onto these

topologies we attached 100 receivers, distributed uniformly. Receiver and sender nodes were always LMS enabled.

In all simulations we enabled with LMS on-tree routers only. We used deployment levels of 10%, 20%, 40%, 60%, 80%, 90% and 100% (full deployment). For each deployment level, we ran 10 simulations on each of the 10 different topologies, meaning that each experiment was a result of 700 simulation runs (10 iterations x 10 topologies x 7 deployment levels) and each point in the graph is the average of 100 simulation runs.

The optimizations described in Section 3 to better target retransmissions and reduce NAK count were *not* used in our simulations. The only changes made to the basic LMS mechanism are the ones that allow the creation of the LMS overlay (i.e., discover reverse paths to LMS nodes and forward paths to repliers). Replier allocation was static: each router selected the nearest downstream replier and maintained the same replier throughout each experiment. We modeled uniform link loss by dropping a packet on every link (one link at a time), and measuring the following: (a) the resulting latency for each affected receiver to recover the lost packet, (b) the resulting exposure, (c) the peak NAK load at the source, and (d) the peak NAK load at the receivers. Most of our experiments were performed with a group size of 100 receivers; some additional experiments were performed using different numbers of receivers to explore the impact of group size on performance. The results did not vary significantly.

5.1. Experiment 1: Random and Random Path Deployment

Figures 14 and 15 show the results for random node and random path deployment on random and transit-stub topologies. We observe that recovery latency varies with deployment, with improvements reaching up to 40% between sparse and full deployment. Exposure, however, varies dramatically depending on deployment level and strategy. With the random node strategy, exposure at sparse deployment is around 80%, and reduces linearly as deployment increases. With the random path strategy exposure improves significantly. Recall that with random path deployment a percentage of receivers is chosen uniformly and LMS is enabled on all the routers on the path from each chosen receiver to the source. The difference between random path and random node strategies is not surprising given that random path deploys LMS faster and more intelligently (i.e., on nodes directly on the path to the source).

Looking at the peak NAK load at the source we note that with the random node strategy the load is high at low deployment, and decreases linearly with increasing deployment. The peak NAK load at the receivers, however, behaves in an interesting manner: it starts low, peaks at some intermediate value, and then drops as deployment approaches 100%. Our explanation for this behavior is as follows: with sparse deployment the source attracts the vast majority of NAKs, many of them being duplicates. As deployment increases more repliers appear, which divert NAKs away from the source. Finally, with dense deployment most NAKs are captured by repliers and are evenly distributed among the increasing number of repliers, which in turn reduces the average load at each replier.

Finally, we note that the results between random and transit-stub topologies exhibit similar trends. The important conclusion from this experiment is that strategies that deploy LMS on random nodes

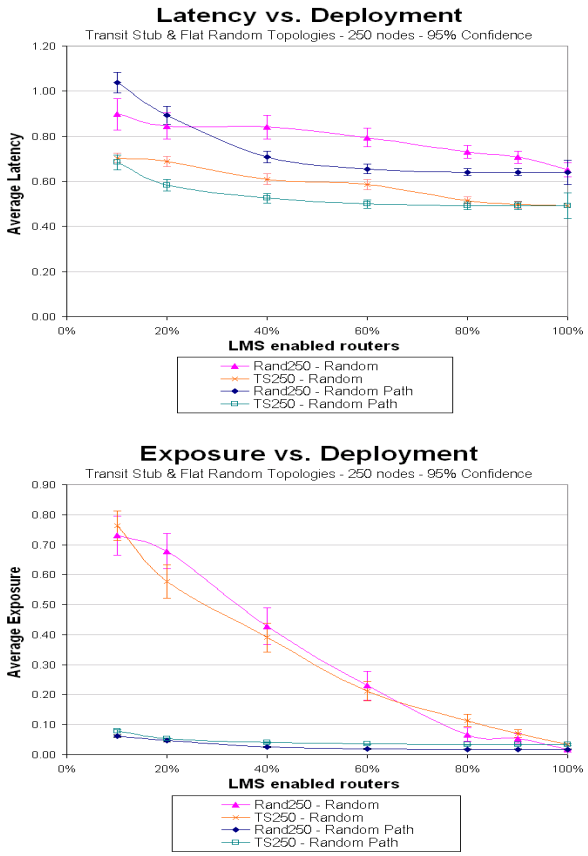


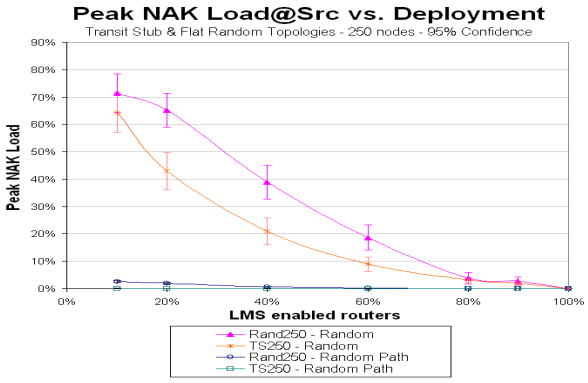
Figure 14: Random node and random path deployment: latency and exposure

are not very efficient. Strategies that deploy LMS on complete paths perform much better.

5.2. Experiment 2: Core, Border Router and Stub Deployment

Our next experiment compares three different deployment strategies, namely core-to-stubs, border-to-core-and-stubs and stubs-to-core. The results are shown in Figures 16 and 17. With the core-to-stubs strategy we begin with all the core nodes LMS-enabled (which is about 10% of all the nodes in our case) and continue deploying LMS uniformly among the remaining routers. With the border-to-stubs-and-core strategy, we begin by enabling the border routers only (i.e., routers connecting the stubs to one of the core nodes) and then uniformly deploy LMS to the remaining routers. Finally, with the stubs-to-core strategy, we begin by fully deploying the service in all the stubs (including the border routers) and we then uniformly deploy LMS among the core routers. We include results from the random-node strategy from the previous experiment for the sake of comparison. Note that deployment levels for different strategies do not correspond to different number of routers.

We observe that with the exception of latency, performance is generally better with core deployment. Exposure reduces



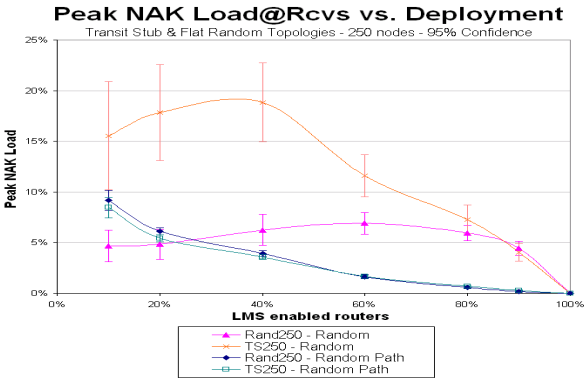
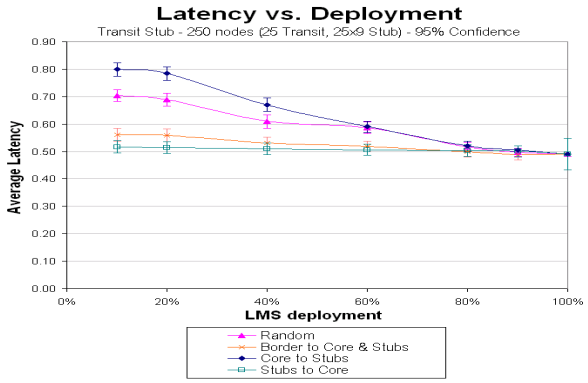


Figure 15: Random node and random path deployment: Peak NACK load at source and receivers



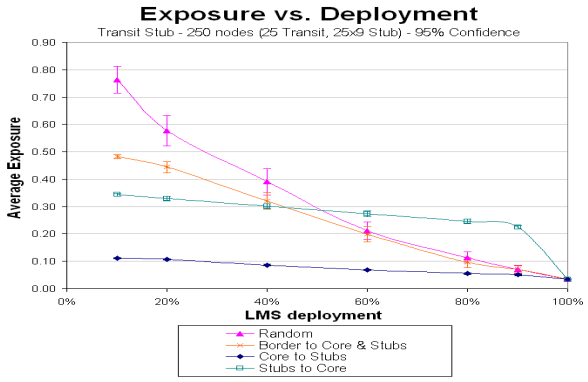


Figure 16: Random topologies, random path: latency and exposure

significantly compared to random deployment when only border routers are enabled, and reduces even more when the stubs are fully enabled. We do not yet fully understand the reasons for the cross-over at about 40% deployment between border and stub strategies. We are in the process of investigating this result.

6. DISCUSSION

Incremental deployment is a crucial element of the success of router-assisted schemes. Schemes whose performance suffers due to lack of a comprehensive plan for incremental deployment are likely to fail to be adopted in the wide area Internet. Despite its importance, this problem has not yet been addressed and has largely been ignored by most of the proposed router-assisted schemes.

In this paper we have taken the first step towards addressing the problem of incremental deployment. We have investigated the performance of LMS, a recently proposed router-assisted reliable multicast scheme, under different deployment strategies. We conducted performance evaluations using three different metrics, namely recovery latency, exposure and peak NAK load at repliers. The first two are the metrics used in the original evaluation of LMS. The third becomes significant due to the mechanics of incremental deployment.

Since the issue of incremental deployment has never been addressed before, there are no established strategies. Therefore, our results are preliminary because they are not based on deployment strategies which have been proven to be viable. However, they show that partial deployment has a significant impact on the selected performance metrics and varies significantly depending on deployment strategy. Latency is affected the least, because even with no LMS deployment losses can still be recovered from the source (at the expense of implosion). Partial deployment has a strong impact on exposure and peak NAK load. Not surprisingly, deployment strategies that deploy LMS on randomly selected routers performed poorly, suggesting that LMS does not perform well with many small pockets of deployment. On the other hand, strategies that deploy LMS on paths from selected receivers to the source performed very well, suggesting that LMS performs best if deployed in contiguous regions. In addition, deploying LMS on only a fraction of the paths seems sufficient to elevate performance to a level close to that of full deployment. A possible reason is that this strategy favors routers closer to the source.

Perhaps not surprisingly, the peak NAK load at either the source or the repliers is quite high with random deployment strategies. Note, however, that high NAK load occurs only when loss affects a large number of receivers and typically affects a very small number of repliers (or just the source). With deployment strategies other than random, the peak NAK load is significantly lower, but does depend on the group size. Thus, it appears that for very large groups a NAK suppression mechanism like the one described in Section 3 may be desirable.

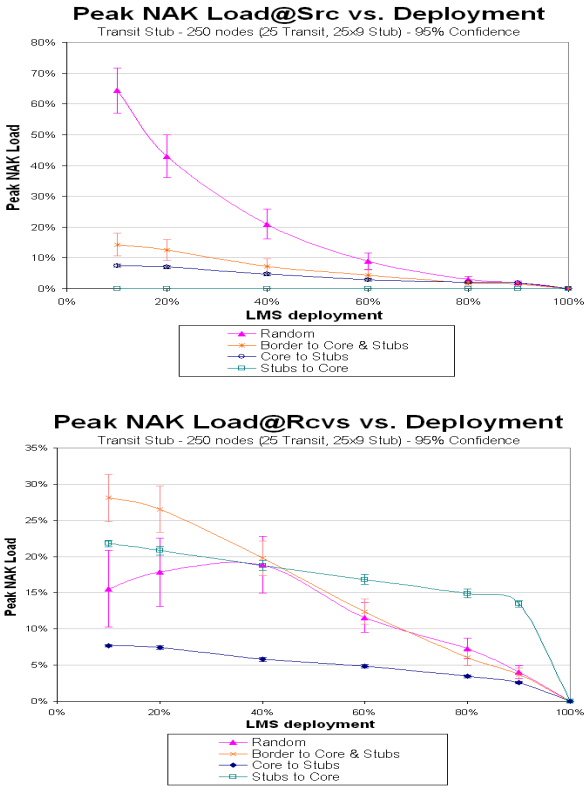


Figure 17: Random topologies, random path: peak NACK load at source and receivers

Transit-stub topologies are different than random topologies because of their hierarchical structure. It has been speculated that these topologies model the Internet better than random topologies, and for this reason we chose to study them. Our results show that for transit-stub topologies, deploying LMS on the core of the network yields good performance, which suggests that an enhanced core allows LMS to target retransmissions better at individual stubs, thus limiting exposure. Unfortunately, core deployment in the wide area Internet may be the hardest to achieve. The next best performing strategy was deployment at the border routers, which while worse than core deployment, was significantly better than random deployment. This suggests that placing an aggregation point at the edge of the network lessens NAK overhead and improves targeting of retransmissions. We consider this result promising, because it implies that a significant performance gain can be realized by initiating deployment at the border routers.

While we have not studied other router-assisted protocols yet, based on our observations the following predictions are plausible. We expect the performance of OTERS to exhibit similar characteristics to LMS because both schemes use turning points in a similar manner. We expect PGM to perform well in terms of exposure at the receivers because its NAK state at the routers allows it to block retransmissions headed for receivers that did not request them. However, unwanted messages may still traverse non-PGM clouds until they encounter a PGM router. For example, with PGM deployment only at the edges, retransmissions and NAK confirmations may traverse the entire network before they are blocked by PGM routers. We plan to study other schemes in more detail as part of our future work.

In conclusion, in this paper we have shown that incremental deployment has a tremendous impact on the performance of router-assisted services. Our results, however, investigated generic deployment strategies. More realistic strategies need to be defined that take into consideration factors like cost, disruption to the network, and performance gains. We plan to work closely with router vendors and network operators and managers to develop such strategies.

7. REFERENCES

[1] Deering, S., “Host Extensions for IP Multicasting,” RFC 1112, January 1989.

[2] Papadopoulos, C., Parulkar, G., Varghese, G., “An Error Control Scheme for Large-Scale Multicast Applications”, Proc. of IEEE INFOCOM’98, San Francisco, CA pp.1188-1196, March 1998.

[3] Levine, B., Garcia-Luna-Aceves, J.J., “Improving Internet Multicast with Routing Labels”, Proc. of IEEE ICNP, Atlanta, GA, Oct. 1997, http://www.cse.ucsc.edu/research/ccrg/publications.html.

[4] Saltzer, J.H., Reed, D.P., Clark, D.D., “End-to-End Arguments in System Design,” ACM Transactions on Computer Systems, Vol. 2, No. 4, November 1984. pp 277, 288.

[5] Speakman, T., Farinacci, D., Lin, S., Tweedly, A., “Pragmatic General Multicast (PGM) Transport Protocol Specification”, draft-speakman-pgm-spec-03.txt, work in progress, June 1999.

[6] UCB/LBNL/VINT Network Simulator - ns (version 2), Software on line, http://www-mash.cs.berkeley.edu/ns/.

[7] Zegura, E., Calvert, K., and Bhattacharjee, S.,. “How to Model an Internetwork.” Proceedings of IEEE Infocom’96, San Francisco, CA.

[8] D. Li and D. R. Cheriton. OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol, Proceedings of 6th IEEE International Conference on Network Protocols (ICNP’98). October 1998, Austin, Texas, pp 237-245.

[9] Papadopoulos, C., “Error control for continuous media and large scale multicast applications,” PhD thesis, Washington University, St. Louis MO, August 1999.

[10] Floyd, S., Jacobson, V., McCanne, S., Liu, C., Zhang, L., “A Reliable Multicast Framework for Light-Weight Sessions and Application Framing,” Proc. of ACM Sigcomm ‘95, pp. 342-356, Cambridge MA 1995.

[11] Lin, J., Paul, S., “RMTP: A Reliable Multicast Transport Protocol”, Infocom ‘96, pp.1414-1424, March 1996.

[12] Differentiated Services, IETF Working group, http://www.ietf.org/html.charters/diffserv-charter.html.

[13] Resource reservation setup protocol, IETF Working group, http://www.ietf.org/html.charters/rsvp-charter.html.

[14] 6Bone, http://www.6bone.net/

[15] Internet Protocol, Version 6 (IPv6) Specification <draft-ietf-ipngwg-ipv6-spec-v2-02.txt> (replaces RFC 1883), December 1998, work in progress.

[16] Katz, D., Atkinson, R., IPv6 Router Alert Option, Internet Draft draft-cisco-ipv6-router-alert-01.txt, work in progress.