Applying Deterministic Feedback Suppression to Reliable Multicasting Protocols

Ronen Chayat and Raphael Rom
Technion—Israel Institute of Technology
Haifa 32000, Israel
{ ronch@tx.technion.ac.il, rom@ee.technion.ac.il }

Abstract -- **IP multicast is becoming the emerging infrastructure for mass delivery of information, from streaming audio and video media to multi-player games, software distributions and shared whiteboards. There have been several proposals to expand the basic unreliable service so that it can provide reliable delivery of data, in terms of ordering and loss recovery. Some of these proposals do not scale well, mainly due to feedback implosion. This is caused by excessive rate of messages arriving from receivers seeking to recover network losses. While many existing reliable-multicasting protocols seek to desynchronize receivers' feedbacks by probabilistic methods, thus circumventing the implosion problem, we show that there are certain deterministic methods, namely the Reactive Window and the Proactive Window, that guarantee implosion avoidance and provide exposure control, without incurring the overhead of excessive state and timer-based maintenance associated with probabilistic schemes. In order to demonstrate their associated performance advantages, we use both methods for building a simple reliable multicast protocol termed SDMP (Scalable Dissemination Multicast Protocol) and compare its performance with PGM (Pragmatic General Multicast), a protocol that uses probabilistic methods of de-synchronization. Our protocol SDMP: (a) takes advantage of spatial and temporal correlation of network events to deterministically control feedback implosion; (b) uses unicast feedbacks and hybrid unicast/subcast retransmissions to control delivery accuracy and exposure, thus conserving network bandwidth; (c) provides shorter arrival and recovery latencies; (d) makes use of network-based processing to detect losses and react on behalf of affected receivers; (e) accommodates local-recovery extensions; (f) has formal proof of correctness.**

I. Introduction

The importance of reliable multicast protocols has been gaining wide recognition in recent years. The unprecedented proliferation of the Internet community has created strong demand for a new class of services, particularly those providing means for groups of users to collaborate and share information over the network in an efficient, real-time manner. Even though the TCP/IP suite has long offered means for efficient multicast routing and delivery [1][2][3], its service quality is derived from the lossy best-effort service model of unicast IP, lacking crucial mechanisms for providing reliability in its corresponding unicast interpretation. In the absence of proven and viable solutions, the strong thrust for seeking answers is well reflected in the growing body of literature on reliable multicasting methods.

There are several difficult scalability challenges associated with reliable multicasting, of which the most important are *implosion control*, *exposure control* and *state management.*

The *sender-initiated* approach that fits reliable unicasting does not scale since the source is required to maintain both *group membership* and the enormous reception state associated with it. Shifting to the *receiver-initiated* approach solves the state problem but introduces *feedback implosion*, which becomes the barrier to scalability. In many scenarios this implosion is due to receiver *feedback synchronization* in the presence of *spatially correlated losses*. Both approaches are faced with the challenge of providing retransmissions only to receivers affected by the loss, avoiding excessive exposure.

Many existing protocols concentrate on solving the implosion problem by suggesting methods to de-synchronize feedbacks from receivers. Some use router modifications to alleviate feedback traffic, e.g., by fusing feedback messages on their way upstream.

However, existing schemes provide only partial solutions. Timer-based, *probabilistic implosion control* mechanisms, such as those presented in SRM [10] and PGM [13], require delicate tuning of timeouts, which cannot always be accomplished. Besides the extra state added, this approach also increases *recovery latency*. Hierarchical approaches such as LBRM [26], RMTP [11] and TMTP [12] are able to provide only approximate scoped recovery, resulting in excessive exposure and repair traffic.

Our Contribution. In this paper we concentrate on the issue of *deterministic receiver de-correlation* and its affiliated advantages in terms of message delivery accuracy, participants exposure and state management. We introduce two novel deterministic suppression methods, the *Reactive Window* and the *Proactive Window*, and use both for building SDMP, a novel NACK-based, router-assisted protocol. It distributes the responsibility for loss detection and repair among the session participants in a deterministic, mutually exclusive manner, so that only a single node is responsible for the repair of each loss. A hybrid unicast/subcast repair mechanism successfully meets the exposure control requirements.

Organization. The remainder of this paper is structured as follows. Section II presents a fundamental background and some of the related work done on reliable multicasting. Section III details the models used for constructing the protocol. Section IV provides comparative simulation results. Conclusions are contained in Section V.

II. Background

Using a rough taxonomy, reliable multicast schemes may be divided into two distinct methods [4]. The first, a sender-initiated approach, strives to extend the unicast corresponding notion by assigning the sender responsible for providing reliability. This is often an *ACK-based* approach that relies on positive feedbacks from receivers and requires maintanance of state and timers at the sender per each receiver in the group. The second is a receiver-initiated approach, often using a *NACK-based* scheme that replaces the group membership maintanance at the sender. It seeks a scalable solution in terms of both processing power and communication bandwidth, hence shifting most of the responsibility to the receivers.

The latter approach, preferred for its scalability properties [5] suffers from a feedback implosion [6] phenomena, in which the sender is flooded with control traffic, hindering a practical deployment. Such implosion is primarily due to correlation between receivers that share a common path from the source on the routing tree, and therefore experience packet loss correlation for every packet lost on the common path. For densely populated groups of receivers, spatial loss correlations account for most of the implosion feedback traffic, whereas for sparsely populated groups, losses are generally spatially independent, except for losses next to the source [7]. Receiver correlation plays a significant role in correct modeling of any particular network, and therefore has a strong influence on issues such as protocol performance, recovery methods and scalability.

Previous work on reliable multicasting includes a few sender-initiated protocols, such as XTP [8], that attempt to implement reliable multicast as an abstract form of unicast, or that use carefully planned polling of receivers to avoid the implosion problem [9]. However, the receiver-initiated approach is more dominant in the design of contemporary protocols because of its scalability advantages over the *sender-initiated* approach. To name a few examples, SRM[10], RMTP, TMTP and PGM are all *receiver-initiated* protocols that build relaibility on top of a best-effort service.

Further classification of reliable multicasting schemes is based upon the method used to recover from packet losses [14]. Some protocols, for instance RAMP [25], use *centralized error-recovery*, also known as *source-based recovery*, in which missing data is recovered exclusively from the original source. Others, like RMTP and LBRM, use *distributed error-recovery* and may provide retransmissions from nodes other than the original source. Such schemes may provide retransmissions to the whole group by multicasting the missing data, but then incur the *repair-locality problem* in which receivers get *unsolicited repairs* for data they already own. Moreover, with most packet loss due to congestion, multicast retransmission can lead to the *self-defeating effect* of increasing the packet loss [16].

More advanced techniques provide local repairs obtained from ordinary local receivers, dedicated *designated receivers*, or *repair servers*. Local repairs use a limited distribution scope, either by restricting retransmissions to a certain subgroup diameter (e.g., setting the TTL value in the IP header, a method used by TRAM [15], SRM and TMTP), or by using directional forms of multicasting (e.g., *subcasting* into a downstream subtree, like in OTERS [16], or into a specified set of downstream links, like in ARM [22]). TTL manipulation is at best a crude method for controlling exposure since estimating the appropriate TTL value is relatively difficult [17].

NACKs, also referred to as *repair requests*, may as well be either multicast or unicast, a decision which impacts the *exposure* and *accuracy* properties of a protocol. In case NACKs are multicast in a distributed recovery environment, certain *retransmission suppression* methods must be deployed in order to avoid multiple retransmissions from several nodes [13].

Certain reliable multicast protocols use tree-based approaches for dealing with feedback implosion [11] (e.g., using methods for *feedback fusion* or *feedback aggregation* in hierarchies leading back to the source, for instance, ACK trees [18]). However, these methods increase *feedback latency* and *recovery latency,* since they require several hop-by-hop processing and aggregation of messages.

In order to avoid feedback implosion resulting from spatially correlated losses, methods of *probabilistic feedback suppression* may be used to de-synchronize receivers (e.g., *NACK-avoidance, NACK-suppression, slotting and damping* schemes). Such methods usually schedule a random timeout when a loss is detected, and suppress feedback transmission if a corresponding feedback generated by another node is heard during the timeout period. Feedback suppression aims at generating a single NACK per loss, but due to its probabilistic nature, redundant NACKs may be generated and must be ignored by the retransmitter. In order for these methods to work well, feedbacks should be multicast so they can be heard by other receivers. An accurate tuning of the timeout period is crucial, and is typically derived from RTT estimates between participating nodes [10][13][19]. Use of probabilistic feedback suppression methods impacts feedback latency and consequently recovery latency, as there is a tradeoff between the amount of feedback generated and the timeliness of the protocol.

Few *deterministic feedback suppression* methods, such as RMP [20], have been proposed in order to bound the number of feedbacks generated by session participants, by allowing only certain receivers to respond at any phase of the data exchange. These methods use tokens and polling that greatly enlarge feedback latencies.

Recently proposed router-assisted schemes, such as PGM, LMS [17], OTERS, ARM and Search-Party [23] require certain modifications to routers in order to control and process the forwarding of feedbacks and retransmissions. These schemes are becoming more popular as more powerful Network Processors are introduced, making it feasible to implement even more sophisticated packet processing and forwarding decisions. They are also inspired by developments in active networking [24].

III. The Model

In this paper we introduce two novel methods, namely the Reactive Window and the Proactive Window, that provide deterministic implosion avoidance and exposure control in the context of a reliable multicast session. Both methods take advantage of packet-loss correlations in order to provide fast repair without compromising delivery accuracy and exposure. In detail, the Reactive Window is a deterministic scheme that uses knowledge of spatial loss correlations to guarantee generation of a single NACK request per packet loss, thus avoiding the well-known *NACK implosion* problem. The Proactive Window is a complementary method that uses knowledge of temporal loss correlations to provide fast recovery, when consecutive losses occur on adjacent links and cause a loss burst. These deterministic methods are used for building SDMP, a simple and efficient protocol scheme for one-to-many reliable dissemination of data. Briefly, SDMP is a receiver-initiated, NACK-based, reliable multicasting protocol that uses deterministic feedback suppression methods to desynchronize feedback from participating nodes. SDMP is the first reliable-multicasting protocol to take this deterministic approach. It provides means for sequenced delivery of information, in a manner guaranteeing eventual delivery of data as long as no network partitioning occurs.

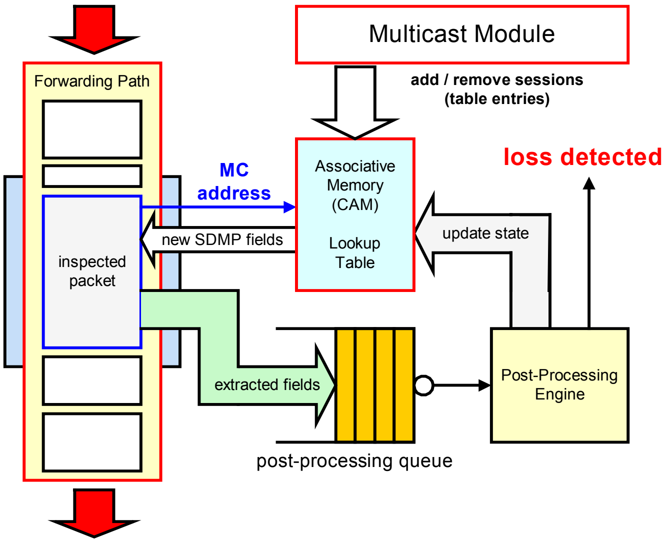

Figure 1: A simplified router modification for SDMP

SDMP provides the following properties: (1) *Implosion avoidance*, generation of exactly single NACK per loss (2) *Exposure control*, no unsolicited messages arrive at participating nodes (3) *Local Repair* (protocol extension, described at [36]), provides retransmissions from local repair servers (4) *Termination property* (protocol extension, described at [36]), guarantees that all participating nodes fully received the entire data sent during the session lifetime, before session tear-down occurs.

The Network Model. SDMP targets "best-effort" IP networks, in which packets may be duplicated, lost, or delayed. Participating nodes are required to monitor their incoming traffic and respond to losses by sending feedbacks back to the sender (or to repair-servers, when local-repair extensions are available). SDMP assumes a static multicast dissemination topology of a tree rooted at the sender, and will use this assumption to infer spatial loss correlations in the network. Integration of SDMP into the IPv4 model may be done by adding a new sublayer between the network (IP) and the transport (e.g., UDP) layers.

SDMP is a router-assisted protocol in that it can take advantage of certain network-based processing, to improve the delivery service for its recipients. Specifically, it requires routers to stamp certain fields in the transport header of SDMP packets, with a summary of the router reception status. Figure 1 depicts a simplified modification of a router-forwarding-path that accommodates SDMP. For each forwarded SDMP packet, the router copies the SDMP header from the packet into a queue for later post-processing, and stamps (replaces) certain SDMP fields in the packet header with previously computed reception-status values stored in an associative memory (CAM). The packet is then forwarded downstream. In order to support multiple multicast groups, the multicast group address in the packet header serves as the key to the CAM. The SDMP headers in the post-processing queue are later used to compute the next reception-status values that will be stored in the CAM for that multicast session. The forwarding and SDMP header processing mechanisms are fully decoupled (asynchronous) in order to avoid performance penalties.

Protocol Architecture Model. SDMP employs a fixed-size transport header accompanied by an on-the-fly field-swapping scheme. Since it requires only a very limited evaluation of each forwarded packet, it becomes an applicable candidate for integration into the fast-forwarding-path of a router, without incurring performance penalties.

Three types of SDMP messages exist: ODATA (original data), NACK (repair request) and RDATA (retransmission).

Each node that participates in an SDMP multicast session maintains a sliding window that represents its current state in the session. Routers and receivers maintain a *receive window* (Figure 2), whereas the source maintains a *transmit window* (Figure 3). The transmit window at the source is composed of:

- *Transmit window leading edge* - denotes the highest SN (sequence number) sent by the source
- *Transmit window trailing edge* - equal by definition to the transmit window leading edge.

An SDMP source disseminates ODATA packets to the multicast address of the session. Each ODATA packet emanating from the source contains a fixed-size SDMP header composed of the following three fields (Figure 3):

- *SN* Field – an end-to-end immutable field carrying an incrementing sequence number, beginning with a 0 for the first packet in a session.
- *Trailing edge* Field – a hop-by-hop mutable field. When the packet leaves the source, this field advertises the trailing edge of the transmit window at the source (equal by definition to the leading edge of the same transmit window). This value denotes the greatest SN transmitted so far. When the packet later traverses intermediate router nodes, this field is replaced hop-by-hop (Figure 4) by an advertisement of the router's recently calculated receive-window trailing edge (SN).
- *Proactive bit-array* (N bits) Field – a hop-by-hop mutable field of size N bits, initialized to zero at the source, used later by intermediate SDMP-aware routers to advertise the reception state of N (protocol parameter) sequence numbers following the trailing edge of their receive window (Figure 4).

The receive window of SDMP-aware routers and receivers is composed of:

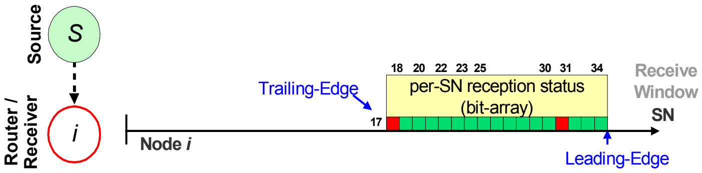

Figure 2: The Receive Window

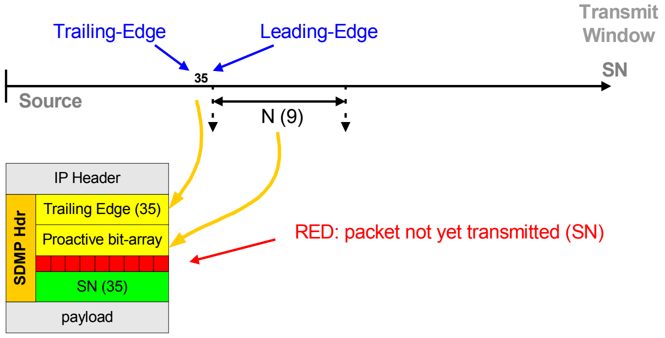

Figure 3: An SDMP Packet Emanating from the Source

- *Receive window trailing edge* - denotes the highest contiguous SN of data packets received by the node. All data packets sent by the source from the beginning of the multicast session up until that SN, are treated as received by the node. This knowledge plays a significant role in the loss-detection mechanism.
- *Receive window leading edge* - denotes the highest SN seen in any of the fields of any SDMP packet type received at the node from the beginning of the session.
- *Receive window per-SN reception status* (bit-array) - describes the reception status of each SN in the receive window portion starting at the window trailing edge, up to and including the leading edge.

In SDMP, each participating node, router or receiver, is responsible for detecting losses on the link that connects it to its parent SDMP-node and initiate a repair process on behalf of the subtree affected by these losses. This subtree is rooted at the node, and may consist of only the node itself in case of an ultimate receiver.

Intermediate routers are expected to participate in the protocol and provide accurate early loss feedbacks, a key component in the performance of SDMP. Since loss of a data packet may occur anywhere on the delivery tree, detection of the loss as close as possible to the loss point guarantees not only shorter recovery latencies, but also accurate *spatial localization* of the loss and its repair. Lower *delivery latencies* result from such faster repair of losses.

Loss Detection and Repair Mechanisms. SDMP nodes detect losses from SN gaps in the packet stream. Tail losses may be detected by introducing low-rate *session messages* or *heartbeat messages* that contain the highest sequence number transmitted so far. Their loss feedback mechanism is built upon two main concepts: the Reactive Window and the Proactive Window. Both windows represent adjacent portions of the receive window of an SDMP node. These two windows enable a node to accurately localize the loss point of its missing packets, by inquiring the reception state of the corresponding sequence numbers covered by these windows. Spatial localization of a loss determines whether the node should proceed with its repair.

The SDMP header portion in each data packet in transit contains a short summary of the reception status of the last SDMP-node it traversed. Each multicast data packet is stamped on-the-fly when passing through an SDMP-router. The router updates two mutable fields in the protocol header that correspond to its reception status for the multicast session (Figure 4):

- *Trailing edge* Field - an advertisement of the router’s recent receive-window trailing edge (SN).
- *Proactive bit-array* Field - an advertisement of the reception state of N sequence numbers following the trailing edge of the receive window. This SN range covers [TrailingEdge+1,TrailingEdge+N] of the router’s receive window bit-array.

The Reactive Window and Proactive Window are both adjacent portions of the receive window of a node. The node is able to detect loss of packets and proceed with repair only for sequence numbers covered by these two windows.

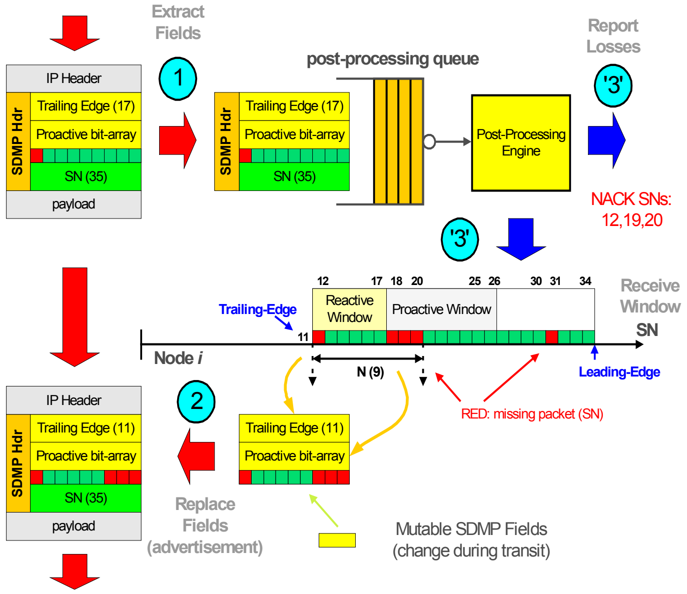

Figure 4: Detailed Step-by-step Router Algorithm

Sequence numbers located left to the Reactive Window were already receiveed by the node, whereas sequence numbers located right to the Proactive Window are either received or missing, but the node is not able to infer whether they were lost on the link connecting it to its parent node, or on earlier portions of the path closer to the source. In order to guarantee a single NACK per loss, each SDMP node is responsible for repairing only losses occuring on the link that connects it to its parent node, therefore it must have an unambiguous localization of the loss point of a packet before it is allowed to proceed with repair of that loss.

The Reactive Window of an SDMP-node stretches from the node's receive window trailing edge, up to and including the recent trailing-edge advertisement made by its SDMP-parent. The portion of the node's receive window reception bit array contained within the Reactive Window lists sequence numbers of packets that are either already received, or lost on the link connecting the node to its SDMP-parent. According to the scheme presented for SDMP, the node is responsible for recovering those losses appearing in its Reactive Window, and therefore must immediately initiate a repair process for them on behalf of the affected subtree.

The Proactive Window of a node is immediately adjacent to its Reactive Window. It covers N (a protocol parameter) consecutive SNs following the highest SN in the Reactive Window. The reception status of each SN in the Proactive Window is compared to the Proactive bit-array advertisement made by the parent node in the last SDMP packet received. Any SN reported by the parent as received should have been also received by the node, otherwise it was lost on the link connecting it to its parent. In that case the node issues a NACK for that missing SN. The Proactive Window allows faster repairs when bursts of losses occur on consecutive links. It aids in circumventing *additive repair latencies* that may emerge when nodes are not able to localize the loss point of their missing packets and therefore withhold generation of NACKs for these losses.

In Figure 4, node *i* contiguously received all packets from the beginning of the session up to SN=11. However, SN=12 is missing. Since its parent, node *i-1* reported a trailing edge of 17, node *i* is able to infer that the packet carrying SN=12 was lost on its incoming link (the link that connects it to its parent *i-1*). It therefore generates a NACK for SN=12. By Comparing the proactive bit-array advertisement made by the parent *i-1* to its own Proactive Window, node *i* can also infer that SNs=19,20 were lost on its incoming link. It then generates a NACK for these losses too. On the other hand, SN=18, which is also reported as missing by the parent *i-1* in its proactive bit-array advertisement, is not handled by node *i* since the loss point is not located on its incoming link.

A node sends a unicast NACK (repair request) towards the source (or a local repair server) as soon as it detects a loss on the link connecting it to its parent SDMP-node. Since each node is responsible only for losses occurring on that particular link, any loss is handled by exactly one SDMP node and *NACK implosion* is deterministically suppressed. In contrast to other router-assisted protocols (e.g., ARM [22]), SDMP does not require the unicast path to the source to be the reverse path for multicast routing. SDMP is able to impose more relaxed constraints on routing because its unicast NACKs are not supposed to generate subscription information in routers along the path to the source, a method used by most router-assisted protocols. The source (or local repair server) unicasts an RDATA retransmission (repair reply) to the requesting node, which in turn subcasts the reply to its own downstream subtree, if such exists.

Delivery accuracy is maintained both for the NACK and for the retransmission: the recipient of the NACK is the source that sent the missing data (or a local repair server that is guaranteed by placement to have that data), while retransmission is delivered only to the subtree affected by the loss. No unsolicited repair requests (NACKs) or repair replies (retransmissions) arrive at other nodes. In order to keep the repair process reliable, a node will repeat the NACK after a certain timeout expires without receiving the corresponding retransmission. Figure 5 depicts typical recovery scenarios for SDMP over a balanced binary tree topology.

Deployment, Interoperability and Feasibility. Although SDMP takes advantage of router-based processing, it does not require an underlying *homogeneous network* in which all routers are SDMP-aware and participate in the protocol. The correctness of the protocol is preserved even if none of the routers in the network is an SDMP-aware router, albeit at a slightly reduced performance. This property allows a gradual deployment in existing networks and interoperability with legacy entities. In general, reduction in performance is common to many other router-assisted, reliable multicast protocols operating in *heterogeneous networks*, composed of protocol-aware and non protocol-aware routers. However, unlike some other protocols (e.g., PGM), SDMP is able to keep exposure control unaffected in such heterogeneous networks, since it uses end-to-end unicast NACKs and hybrid unicast/subcast retransmissions, instead of using *convergecast* or other feedback-fusion techniques that heavily depend on protocol-aware router processing.

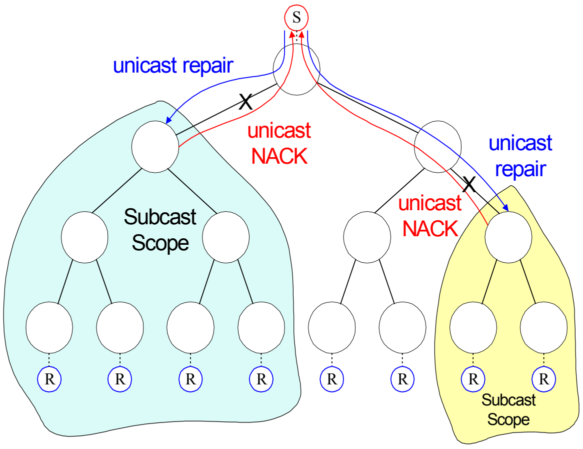

Figure 5: SDMP over a balanced binary tree topology

The amount of state required for each multicast session monitored by an SDMP-router is independent of the number of receivers, making this approach scalable [31].

The stamping process, applicable for deployment in the fast-forwarding path of a router, involves only the replacement of certain fields at known offsets in the packet header. It does not require complicated packet processing nor relies on information present in the fields replaced. Moreover, in order for the protocol to operate correctly, the new values placed in the packet header do not have to reflect immediate changes in the node's receive window due to that packet reception, or reception of packets preceding it. The router places the old contents of the packet fields it replaced into a queue for later processing. This post-processing is asynchronous to the forwarding path (Figures 1,4). The post-processing engine extracts these fields from the queue and uses them to infer losses on the link connecting the node to its SDMP-parent.

Proof of Correctness. Differing from most other reliable multicast protocols, SDMP has a formal proof of correctness [36].

IV. Simulation Results

In order to evaluate the behavior and performance of the Reactive Window and Proactive Window schemes, we integrated them into the SDMP reliable multicasting protocol and built a simulation environment using the NS simulator from U.C. Berkeley/LNBL. NS allows the user to define arbitrary network topologies, composed of routers, links and shared media. A rich set of built-in and contributed protocol agents is available for selection. The user may instantiate these agents and attach certain protocols to nodes on the topology.

We chose to concentrate on comparing SDMP against PGM, since the latter is a relatively mature scheme that has been tested and deployed in several router implementations. Though both PGM and SDMP are router-assisted schemes, they differ in almost every aspect of their operation. For instance, PGM uses probabilistic feedback suppression methods, while SDMP uses deterministic. PGM employs *state-based scoped retransmissions*, while SDMP uses hybrid unicast/multicast repairs. PGM routers do not process ODATA and therefore do not initiate repairs, SDMP does both.

The results show that SDMP performs well with respect to the following parameters of interest:

- average packet arrival latency
- maximum packet arrival latency
- implosion control
- repair traffic

Latency results are presented in both absolute and normalized time, according to the measured RTT to the source.

Topology. Simulations were run on a synthetic, 128-node, balanced binary tree topology, similar to the topology structure in Figure 5. The tree consists of a single source, 63 SDMP-routers and 64 receivers. Each receiver is exactly 7 hops away from the source. All links are identical and symmetric, 1.5Mbps bandwidth, 10ms delay each. Experiments with other topologies, e.g., shoestrings of different lengths and quad trees of different heights, yielded results similar to those presented for the binary tree.

Traffic. Source rate was set to 128Kbps CBR. Original data packets (ODATA) are sent at a rate of 16 packets per second, i.e., every 64ms. Each ODATA packet is 1024 bytes. No restrictions were imposed on the rate of outgoing repair traffic. NACK packets are 256 bytes; RDATA packets are 1024 bytes, same size as ODATA.

Loss model. Implemented independent losses on links, with predetermined probability set at the beginning of a test. Repair requests are fully reliable (types NAK, NCF for PGM, and NACK for SDMP). Retransmissions (RDATA) are reliable only from the source node down to the subcast point (for PGM, down to the node pass the original loss point). From that point, down to the ultimate receiver, RDATA packets are affected by the same loss mechanisms as ODATA. This particular selection is more realistic than models that assume no loss of retransmissions [10][35][17][22]. At the same time it simplifies the interpretation of simulation results.

General settings. SDMP proactive window size (N) was set to 16 sequence numbers. PGM parameter tuning followed the available code contributed by the authors of [35].

Data collection. Parameters of interest presented in the following graphs are measured on a wide range of link loss probabilities, ranging from 0% to 8% in equal steps of 0.1% each. Each such step is represented by a point on the graph, which corresponds to an average of 100 measurements of approximately 1MB traffic each, using a different random seed for each run. Total data exchanged during the simulation exceeded 8GB. Our experiments show that the use of many runs with different seeds helps to create smoother plots and remove much of the noise appearing in graphs that rely on fewer samples.

Average arrival latency. Figures 6 and 7 show measurements of arrival latency, including for packets that were lost and recovered, and therefore their arrival latency values also reflect the time elapsed during the repair process. The latency is measured from the dispatch of a particular packet until it reaches the set of ultimate receivers. Arrival latency measurements are identical for PGM and SDMP when all links are error-free. However, when loss probability on links increases, the advantage of SDMP in providing earlier loss feedbacks by routers is substantial.

Another issue is a design weakness of PGM, the *PGM dangling NAK state*, pointed out in [35]. When PGM operates in a lossy environment, RDATA packets become subject to network losses, and recovery latencies become affected by stale states in routers. Since PGM routers use *NAK-elimination* techniques to avoid feedback implosion, when the RDATA corresponding to the state is lost, further NAKs are blocked until the stale state is cleared (the inherited parameter in our tests is 10 seconds, meaning the router waits for the repair up to 10 seconds, after that it removes the state). When loss probability increases, RDATA losses become more dominant and contribute to the steep slope in the arrival latency graphs for PGM. A workaround for this weakness is proposed in a new PGM draft [34] and is yet to be tested.

Maximum arrival latency. Figures 8 and 9 show the effects of the dangling NAK state on the maximum arrival latency experienced by receivers. For each test run, the list of maximum arrival latencies experienced by all 64 receivers is averaged to a single value. This run is repeated 100 times for different seeds, resulting in 100 such values that are averaged into a point on the graph. Since SDMP does not employ NAK-elimination techniques nor uses *state-based forwarding* decisions, the increase in its maximum arrival latency is moderate. Due to stale states, PGM may delay the delivery of data packet for tens of seconds, even when link loss probability is relatively low.

Proactive Window Size. Figure 11 depicts the impact of N, the size of the Proactive Window, on the performance of SDMP. The topology used to create this graph is a 100-hop shoestring, 0.1% identical link loss-probabilities for all links, N is taken from the list {0, 4, 8, 16}, source is 64Kbps CBR, loss of ODATA packets only, feedbacks and retransmissions are not lost. All other parameters identical to those selected for the binary-tree topology. In this topology, the single receiver resides 100 hops away from the source. The actual arrival time at the ultimate receiver is shown for each packet, identified by its sequence number. The optimal no-loss curve closely follows the CBR source. The source starts transmitting packets at T = 2 Secs.

For N=0, the Proactive Window is effectively disabled, leading to *additive recovery latencies* when temporally-correlated losses occur. When N=0, SDMP cannot repair loss bursts in an efficient manner. Selection of a higher N provides more protection against temporally-correlated losses. Experiments we conducted on different topologies using different loss patterns, show an effect of *diminishing returns*, i.e., selection of N=16 produces results fairly identical to those achieved using an optimal 'infinite N' (meaning, an N value large enough to cover the whole receive window of any node. For that selection, each node sees the whole receive window of its parent SDMP-node and is able to infer its losses directly without waiting for repairs requested by its parent).

V. Conclusions

We study the scalability properties of reliable multicast protocols. In this paper, we devise an efficient protocol that combines most of the fundamental known approaches for providing scalability, with newly introduced techniques:

- ***Deterministic feedback suppression,*** based on spatial loss correlations. Augmented with efficient repair support for temporal loss correlations.
- ***Scoped retransmissions,*** by deterministic grouping of receivers affected by a certain loss.
- ***Delivery accuracy, exposure control,*** using a hybrid unicast/subcast repair process.
- ***State minimization*** avoidance of timer-based schemes as much as possible. Use of efficient per-session state representation in routers.

All these techniques mentioned assist in conserving network bandwidth and processing power. SDMP is the first router-assisted protocol to suggest interaction of routers with original data packets (ODATA), in a way that may be applied on-the-fly without expensive header processing. We argue that our approach significantly improves the performance of reliable multicast protocols, albeit the added deployment efforts.

As seen in simulations, SDMP is more robust than protocols that use hop-by-hop feedback forwarding, like PGM. The authors of OTERS [16] already argued that schemes of the latter kind are subject to failures in network elements. SDMP nodes interact directly with the sender (or a local repair server) during the repair process.

Many other aspects of our work, including a formal proof of correctness, more extensive simulations and detailed descriptions of other protocol extensions and aspects (e.g., local recovery, support for special network configurations) could not have been conveyed under the limited scope of this paper, however they are fully described in the thesis work available at [36].

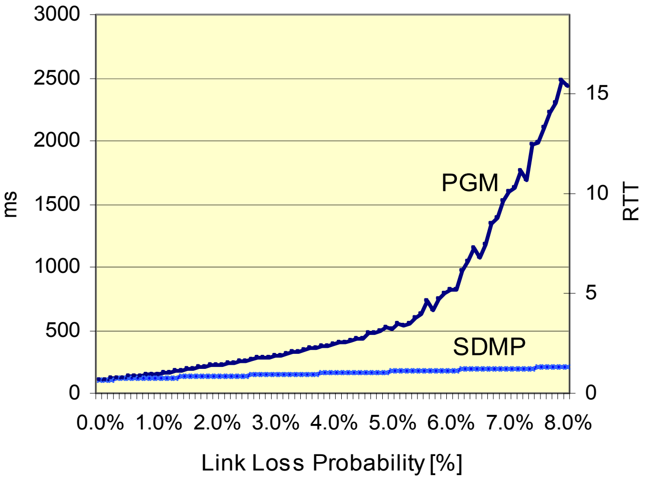

Figure 6: Average packet arrival latency vs. link loss probability

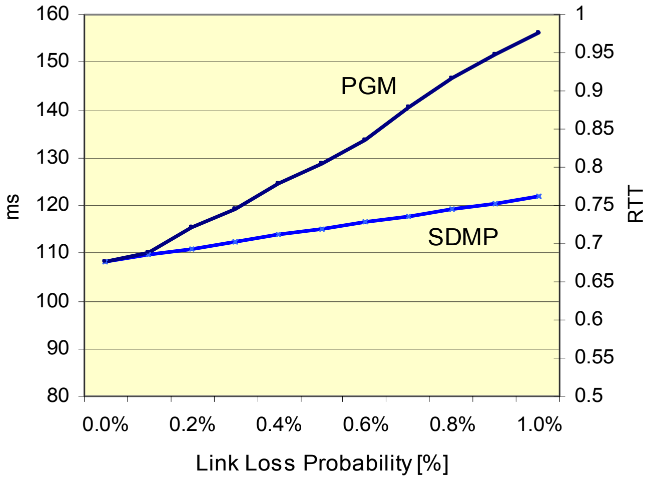

Figure 7: Average packet arrival latency vs. link loss probability (Enlarged portion of Figure 6)

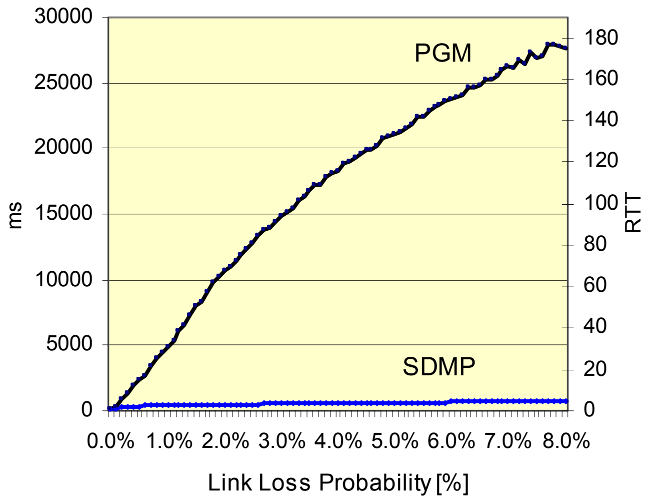

Figure 8: Receiver experienced maximum arrival latency vs. link loss probability, averaged on receivers per test run

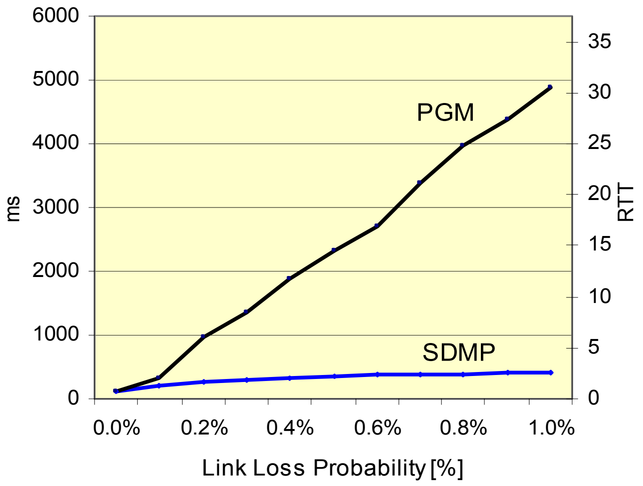

Figure 9: Receiver experienced maximum arrival latency vs. link loss probability, averaged on receivers per test run (Enlarged portion of Figure 8)

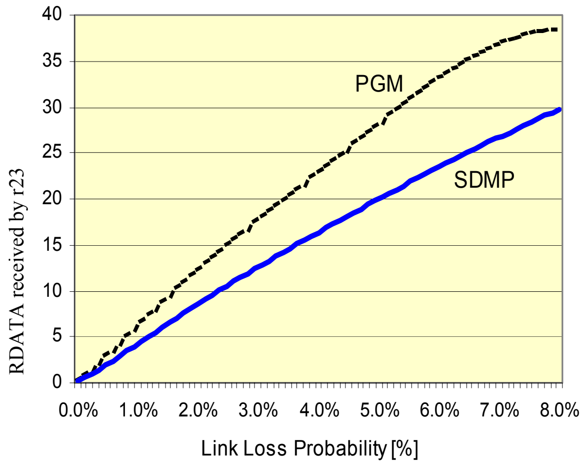

Figure 10 – Receiver Repair Traffic (Total RDATA received by an ultimate receiver r23)

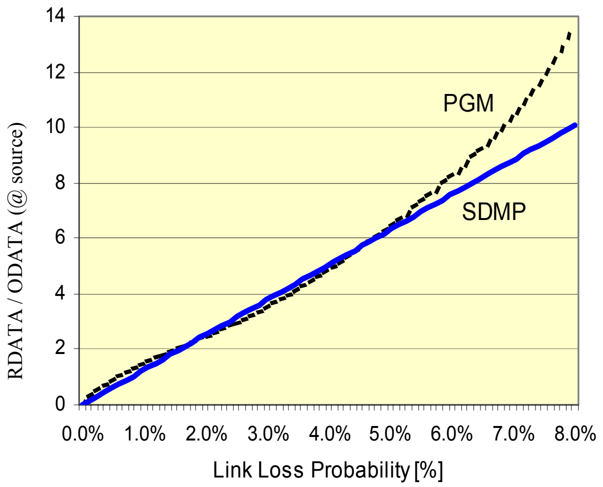

Figure 11 – Source Repair Traffic Ratio (Ratio of RDATA packets to original ODATA packets sent by the source)

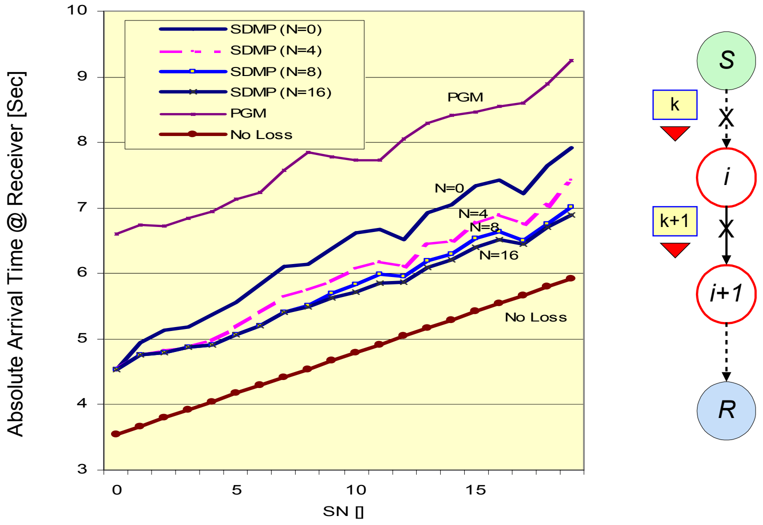

Figure 12: Impact of Proactive Window size on the performance of SDMP

VI. REFERENCES

[1] D. R. Cheriton and S. E. Deering. *Host Groups: A Multicast Extension for Datagram Internetworks*. In Proceedings of the Ninth Data Communications Symposium. ACM/IEEE, September, 1985.

[2] S. Deering, *RFC-1112: Host Extension for IP Multicasting*, Request For Comment, August 1989.

[3] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei, *An Architecture for Wide-Area Multicast Routing (PIM)*, in ACM SIGCOMM'94 Conference, February 1994.

[4] Brian Neil Levine, A Comparison of Known Classes of *Reliable Multicast Protocols*, MSc thesis, University of California, June 1996.

[5] Don Towsley, Jim Kurose and Sridhar Pingali, *A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols*, ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Proceedings, Performance Evaluation Review, volume 22, May 1994.

[6] P.B. Danzig, *Optimally Selecting the Parameters of Adaptive Backoff Algorithms for Computer Networks and Multiprocessors*. PhD thesis, University of California, Berkeley, December 1989.

[7] Maya Yajnik, Jim Kurose and Don Towsley, *Packet Loss Correlation in the MBone Multicast Network*, In Proceedings of IEEE Global Internet, November 1996.

[8] *XTP Protocol Definition Revision 3.6*, Protocol Engines Incorporated, PEI 92-10, Mountain View, CA, January 1992.

[9] Marinho P. Barcellos and Paul D. Ezhilchelvan, *An End-to-End Reliable Multicast Protocol Using Polling for Scalability*, in IEEE INFOCOM'98.

[10] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne and Lixia Zhang, *A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing (SRM)*, in ACM SIGCOMM'95, October 1995, pp. 342-356.

[11] S. Paul, K. K. Sabnani, J.C. Lin, and S. Bhattacharyya, *Reliable Multicast Transport Protocol (RMTP)*, in proceedings of the IEEE INFOCOM, San Francisco, CA, March 1996.

[12] R. Yavatkar, J. Griffioen, and M. Sudan, *A Reliable Dissemination Protocol for Interactive Collaborative Applications (TMTP)*. In Proceedings of ACM Multi-media, 1996.

[13] Speakman, T., Farinacci, D., Lin, S., Tweedly, A., *Pragmatic General Multicast (PGM) Transport Protocol Specification*, in Internet Draft, August 1998.

[14] Jorg Nonnenmacher, Martin Lacher, Matthias Jung, Ernst W. Biersack, George Carle, *How Bad is Reliable Multicast without Local Recovery ?* In IEEE INFOCOM'98.

[15] M. Kadansky, D. Chiu, J. Wesley, J.Provino, *Tree-based Reliable Multicast (TRAM)*, Internet draft, draft-kadansky-tram-02.txt

[16] D. Li and D. R. Cheriton, *OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol*, Proc. 6th IEEE International Conference on Network Protocols (ICNP'98), October 1998.

[17] Papadopoulos, C., Parulkar, G., Varghese, G., *An Error Control Scheme for Large-Scale Multicast Applications (LMS)*, Proceedings of INFOCOM'98, San Francisco, CA, March 1998.

[18] B.N. Levine, David Lavo, and J.J. Garcia-Luna-Aceves, *The Case for Concurrent Reliable Multicasting Using Shared Ack Trees*, Proc. ACM Multimedia 1996, Boston, MA, November 1996.

[19] Dante DeLucia, Katia Obraczka, *Multicast Feedback Suppression Using Representatives*, in IEEE INFOCOM'97.

[20] B. Whetten, S. Kaplan, and T. Montgomery, *A High Performance Totally Ordered Multicast Protocol (RMP)*, August 1994.

[21] Injong Rhee, Srinath R. Joshi, Minsuk Lee , S. Muthukrishnan, and Volkan Ozdemir, *Layered Multicast Recovery (LMR)*, in IEEE INFOCOM'2000.

[22] Li-wei H. Lehman, Stephen J. Garland, and David L. Tennenhouse, *Active Reliable Multicast (ARM)*, in IEEE INFOCOM'98.

[23] Adam M. Costello, Steven McCanne, *Search Party: Using Randomcast for Reliable Multicast with Local Recovery*, in IEEE INFOCOM'99.

[24] David Tennenhouse and David Wetheratl, *Towards an Active Network Archhecture*, in SPIE Proceedings of Conference on Multimedia Computing and Networking 1996, San Jose, CA, January 1996.

[25] R. Braudes and S. Zabele, *Requirements for Multicast Protocols (RAMP)*, RFC-1458, IETF, May 1993.

[26] H. W. Holbrook, S. K. Singhal, and D. R. Cheriton, *Log-Based Receiver Reliable Multicast for Distributed Interactive Simulation (LBRM)*, in proceedings of ACM SIGCOMM'95, October 1995.

[27] B.N. Levine, S. Paul, and J.J. Garcia-Luna-Aceves, *Organizing Multicast Receivers Deterministically by Packet-Loss Correlation*, Proc. Sixth ACM International Multimedia Conference (ACM Multimedia 98), September 1998.

[28] D. Katz, *IP Router Alert Option*, RFC-2113, IETF, February 1997.

[29] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC-1883, IETF, December 1995.

[30] C. Partridge and A. Jackson, *IPv6 Router Alert Option*, RFC-2711, IETF, October 1999.

[31] Jorg Nonnenmacher and Ernst W. Biersack, *Optimal Multicast Feedback*, in IEEE INFOCOM'98.

[32] Yatin Chawathe, Steven McCanne, Eric A. Brewer, *RMX: Reliable Multicast for Heterogeneous Networks*, in IEEE INFOCOM'2000 conference, March 2000.

[33] Kang-Won Lee, Sungwon Ha, Vaduvur Bharghavan, *IRMA: A Reliable Multicast Architecture for the Internet*, in IEEE INFOCOM'99.

[34] Speakman, T., Farinacci, D., Lin, S., Tweedly, A., et al., *Pragmatic General Multicast (PGM) Transport Protocol Specification*, draft-speakman-pgm-spec-04.txt, April 2000.

[35] Christos Papadopoulos, Sherlia Shi, Guru Parulkar and George Varghese, *Performance Comparison of LMS and PGM using Simulation*.

[36] Ronen Chayat, *Improving the Functionality of Reliable Multicast Protocols*, available at: http://www-comnet.technion.ac.il/~ronch/SDMP