The PGM Reliable Multicast Protocol

Jim Gemmell
Microsoft Research

Todd Montgomery
Talarian

Tony Speakman
Cisco Systems

Nidhi Bhaskar
Cisco Systems

Jon Crowcroft
University of Cambridge.

Abstract

Pragmatic General Multicast (PGM) is a reliable multicast transport protocol that runs over a best effort datagram service, such as IP multicast. PGM obtains scalability via hierarchy, forward error correction, NAK (negative acknowledgement) elimination, and NAK suppression. It employs a novel polling scheme for NAK delay tuning to facilitate scaling up and down.

This paper describes the architecture of PGM and discusses performance and security issues. We show that PGM has supports asymmetric networks, achieves high network utilization, and is capable of high-speed (>100 Mbps) operation.

PGM is currently an IETF experimental RFC that has been implemented in both commercial and academic settings.

1 Introduction

Pragmatic General Multicast (PGM) is a reliable multicast transport protocol for applications that require multicast data delivery from a single source to multiple receivers. PGM runs over a best effort datagram service, such as IP multicast. PGM guarantees that a receiver in the group either receives all data packets from transmissions and repairs, or is able to detect (rare) unrecoverable data packet loss. It obtains excellent scalability via hierarchy, forward error correction, NAK (negative acknowledgement) elimination, and NAK suppression. PGM is now an IETF experimental RFC. This paper describes the architecture of PGM and provides some analysis of its performance.

Reliable transport is an end-to-end proposition, as failure in intermediaries must always be accommodated. The most well-know transport protocol is TCP. A naïve implementation of reliable multicast might send messages using IP multicast, and then mimic the behavior of TCP by having receivers acknowledge (ACK) reception of messages, with sending proceeding a la TCP according to the results of the slowest receiver. However, such an approach demonstrates the chief difficulty in making reliable multicast scalable: message "implosion". As the number of receivers increase, their message traffic back to the source will eventually overwhelm either the

source or the network links leading to it. Another problem is that memory proportional to the number of receivers is needed to keep track of each receiver's state, but this is not as severe as the implosion problem.

A number of approaches have been taken to improve the scalability of reliable multicast. One approach is NAK suppression [1, 2]. In this scheme, receivers do not ACK messages, but only negatively acknowledge (NAK) messages they have missed. Before sending a NAK, they wait for a time randomly chosen from an interval. If, while they are waiting, some other receiver sends a NAK for the same packet, then they "suppress" their NAK: it is not sent, but the receiver acts as if it has sent it (since the source got it from another receiver, which is just as good). Receivers find out about others sending NAKs either by receivers multicasting NAKs, or by the source multicasting a NAK confirmation (NCF). The random delay along with suppression is intended to prevent implosion. However, to ensure that implosion does not occur, delays prior to NAKing must increase with the number of receivers. Excessively long delays lead to many problems such as enormous transmit windows and the possibility that a receiver may not be able to send a NAK until the session has ended.

Another way to improve scalability is via hierarchy [3, 4, 5, 6, 7]. A tree is built for the reliable multicast session made up of either receivers or special intermediary nodes. NAKs or ACKs are only sent from a node to its parent. The parent aggregates or eliminates the information before forwarding up the tree. For example, if several children have lost a packet, only one NAK needs to be forwarded up the tree ("NAK elimination"). Another possibility with hierarchy is constrained forwarding of repairs such that they are only forwarded to sub-trees containing receivers that need the repair.

A difficulty with scaling is that as the number of receivers grows, the chances increase that any given packet will be lost by some receiver increases. For example, with 1,000,000 receivers, and a random, independent, loss of 0.01% the chances of all nodes receiving a given packet is less than 10^{-43}. So it is a virtual certainty that each packet will be lost by some node. Therefore, for all receivers to receive all packets, each packet must be sent at least twice, cutting the effective bandwidth in half. In order to avoid this, forward error correction (FEC) may be able to correct different losses across different receivers [8, 9, 10, 11, 12, 13]. For example, if one receiver loses packet 1, and another loses packet 2, a single repair packet containing the parity of packets 1 through 7 would allow both receivers to repair their loss (given they had no other losses).

PGM uses a hybrid scheme including suppression, NAK elimination, constrained forwarding, and FEC to achieve scalability. Hierarchy is constructed using PGM-capable Network Elements (NEs – typically routers enhanced to support PGM in addition to IP multicast). PGM is designed to operate, albeit with less efficiency, even if some or all of the NEs are not PGM-capable. When PGM-capable NEs are few, the fan-out of the PGM tree is increased, giving suppression and FEC a more critical role in providing some scalability.

PGM does not require the receiver to multicast, making it applicable to networks that are only multicast capable from sender to receivers. PGM also makes efficient use of back-channel bandwidth (see section 3) making it well-suited to asymmetric networks that have a high capacity channel from the sender to receivers, but have a constrained back-channel from the receivers to the sender.

PGM is not intended for use with applications that depend either upon acknowledged delivery to a known group of recipients, or upon total ordering amongst multiple sources, and does not support these features. PGM permits receivers to join and leave at any time, providing reliability only within the current transmit window. Hence, it is best suited to applications that can tolerate some lost packets or support some application-level recovery. This is not to say that recurrent unrecoverable packet loss is to be expected with PGM; reliable transmission can be expected as

long as the sender does not advance the transmit window too aggressively. Rather, it is a feature of PGM multicast that an application may choose to advance with sending old data, determining that old data is now "stale" or that it is more important to move on than to satisfy late joiners.

Examples of applications suited to PGM are stock quote dissemination and disk imaging. In the case of the stock quotes, reliability is very important, but if transmission has still not been successful by the time the quote has been superceded it should be abandoned. For disk imaging, it can be more efficient to carry on with new data, rather than slowing down for a few slow receivers, who can fetch the missing data later via standard client-server methods.

We provide more details on the architecture of PGM in section 2. Section 3 considers PGM performance and security issues. Section 4 covers related work, and Section 5 offers our conclusions.

2   Architecture

We now describe the architecture of PGM. This section is not intended to be exhaustive, but rather to cover the most essential and interesting features of PGM. For a full description, the reader is referred to [14]. PGM can be deployed over any multicast mechanism, but for clarity we will focus our discussion to IP multicast in this paper.

The PGM tree is constructed using the PGM-enabled NEs in the existing multicast tree. A PGM source multicasts sequenced data packets (ODATA) to the receivers. When receivers detect packets missing from the sequence, they unicast a NAK to their parent in the PGM tree. Parents confirm the reception of the NAK to all their children with a multicast NAK confirmation (NCF). Repairs (RDATA) are generated either by a source or a Designated Local Repairer (DLR) in response to a NAK (PGM NEs never store ODATA or provide repairs). A repair is either the lost packet re-sent or an FEC packet, depending on session parameters. Before sending a NAK, receivers perform a random back-off, and suppress the NAK if they receive a corresponding NCF, data or repair data.

2.1   Hierarchy

To establish the PGM hierarchy, PGM senders periodically emit Source Path Messages (SPMs). SPMs also serve other purposes, some of which we will discuss below (e.g., they also update the highest sequence number, allowing receivers to detect losses without any new ODATA).[1] Each SPM contains the address of the PGM parent node that it came from. NEs replace this address with their own when they forward an SPM, so that their children will know their parent. This technique allows non-PGM NEs to operate transparently between PGM nodes. If the multicast routing changes, or if a PGM NE reverts to non-PGM operation, SPMs cause the PGM tree to be updated.

Thus, the PGM tree precisely overlays the raw multicast tree. If all NEs are PGM-enabled, the trees will be identical. All multicast packets are sent with the packet source address set to that of the PGM source, even if they originate from an NE. This ensures they travel the same path as any packets coming from the PGM source. Using the same tree avoids the need to create additional NE state for the PGM tree, and also avoids problems such as encountering different loss characteristics between two trees.

---

[1] Additional information SPMs can carry include: the current transmit window, the NAK back off interval, a "finish" option to indicate transmission is complete, a "reset" option to indicate sender error, and information on any FEC being used.

To minimize NAK loss, PGM defines a network-layer hop-by-hop procedure for NAK forwarding. After detecting a lost packet, a receiver repeatedly unicasts a NAK to its PGM parent until it receives an NCF for the packet. Upon receipt of a NAK, NEs that do not eliminate it immediately unicast the NAK to their parent until an NCF is received. The NAK will be repeated just a few times over a short interval until either an NCF is received or these attempts fail.[2] Whether or not it is successfully forwarded, the NAK is then discarded by the NE. So NAK forwarding by the NEs improves but does not guarantee the reliability of NAKs. The ultimate responsibility for regenerating unconfirmed NAKs falls back upon the receiver in keeping with the end-to-end principle.[3]

PGM NEs also perform "NAK Anticipation". That is, they will sometimes receive an NCF that does not correspond to any outstanding NAK. If the NCF comes from the NE's parent, the NE will record the NCF in anticipation that its children may NAK for the same packet. If this happens, the NAK can be confirmed without any further action, as it has already been confirmed up-stream. NAK anticipation, along with NAK elimination, serves to improve the odds that only one NAK is forwarded upstream, fulfilling the goal of NAK reduction.

NCFs are multicast. However, they are not propagated by PGM NEs, as they act as hop-by-hop confirmations. PGM parent nodes (NEs or the source) generate an NCF for every NAK, even if the NAK has been previously confirmed. However, reception of a duplicate NAK does not cause an NE to NAK upstream if it has already been confirmed by its parent.

Figure 1 shows the sequence in a NAK/NCF scenario, where both PGM and non-PGM routers are present. Recall that all multicasts originating from PGM NEs are generated as if they originated from the sender, so that the non-PGM routers will forward them on the same multicast tree used for multicast data coming from the sender. This scenario begins when the receiver on the far left detects a loss and (1) unicasts a NAK to its PGM parent. Its PGM parent router multicasts a NAK confirmation (2), suppressing any NAKs that the other receivers might have generated for that packet. The router now unicasts a NAK (3) to its parent, who multicasts an NCF(4). Note that this NCF is not forwarded by the downstream PGM nodes. The PGM router on the right will store the NCF for future reference. A unicast NAK (5) is then sent to the sender, who responds with a multicast NCF (6) (again, not forwarded by downstream PGM routers). At a later time, another receiver detects the loss of the same packet and unicasts a NAK (7) to its PGM parent, who multicasts an NCF (8). The parent does not send an upstream NAK, as it has already noted the reception of the corresponding NCF ("NAK anticipation").

---

[2] An NE will continue to NAK and wait for an NCF for up to NAK_RDATA_IVL time units. NAKs are sent every NAK_RPT_IVL time units. The PGM RFC does not specify these times. One implementation sets NAK_RDATA_IVL to two seconds and NAK_RPT_IVL to 0.5 seconds.

[3] The receiver only needs to know it is talking to a PGM parent according to the protocol. Whether this is a NE or the sender, or indeed whether there are any PGM NEs, makes no difference to the receiver.
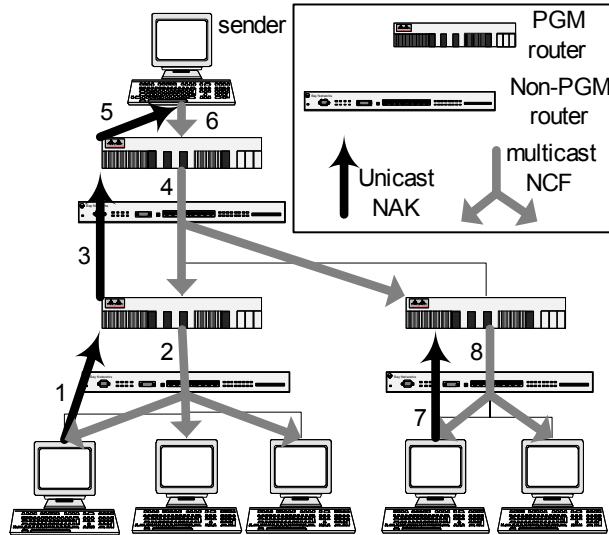
Figure 1: NAK/NCF scenario. (1) Receiver NAKs for a lost packet. (2) Its PGM parent router multicasts an NCF (3). The router unicasts a NAK to its parent, who (4) multicasts an NCF. (5) A unicast NAK is sent to the sender, who responds with a multicast NCF. (6) At a later time, another receiver detects the loss of the same packet and unicasts a NAK (7) to its PGM parent, who multicasts an NCF (8). The parent does not send an upstream NAK, as it has already noted the reception of the corresponding NCF.

Network elements do not attempt to ensure that RDATA is received in response to a NAK. It is up to receivers to re-issue the NAK (following the end-to-end principle). NEs simply discard NAK state after a time-out, which has the added benefit of keeping repair state in the network relatively responsive to routing changes.

When repairs are sent, PGM NEs will only forward them on interfaces for which they have received a NAK corresponding to the repair. This is called "constrained repair forwarding". Thus, repairs are only forwarded to sub-trees containing receivers who need the repair. Note that PGM's scheme of propagating NAKs upstream follows the same sequence of PGM networks on which the ODATA was forwarded, but in reverse. This allows the appropriate NAK state to be established in the NEs to support constrained forwarding.

SPMs, NCFs, and RDATA require special treatment by PGM network elements. An obvious way to discover these packets is to examine every packet in the network for the PGM transport protocol and packet types. However, the burden on the NE of examining every packet is prohibitive. Instead, SPMs, NCFs, and RDATA are transmitted with the IP router alert option.[4] This option gives network elements a network-layer indication that a packet should be extracted from IP switching for more detailed processing. Original data packets are forwarded just as any other IP multicast packets would be, without any special router attention.

2.2 NAK suppression

Receivers delay for a random time within an interval before sending a NAK. If a matching NCF, RDATA or ODATA is received during the delay, the receiver suppresses its NAK. After sending

---

[4] SPMs used to update session window parameters do not carry the router alert option. This is to prevent hop by hop (slow) processing by PGM NEs and ensure that the window information remains in sync with the ODATA. Only SPMs meant to refresh path state carry the router alert option. PGM NEs update the upstream neighbour information with their address when forwarding these SPMs.

or suppressing a NAK, the receiver waits for an NCF up to a given time-out. After receiving an NCF, the receiver waits for RDATA or ODATA to repair the loss up to a given time-out. If the receiver times out waiting for the NCF, or waiting for packet repair after an NCF, it starts over and issues another NAK after a random delay with suppression.

The base delay interval is specified in the SPMs, which allows the PGM parent to adjust delays based on its experience in the session, and its estimate of the number of its children (see section 2.6). The actual delay interval used by the receiver is derived from the base interval, but may be increased based on the number of time-outs for NCFs/repairs that have occurred. The interval is also increased if the NAK is for more than one parity packet (see section 2.3).

PGM also supports the optional multicasting of a NAK with TTL=1 in conjunction with sending a unicast NAK to a parent node. Other nodes receiving the multicast NAK will suppress their NAKs. This option is in place to achieve improved/faster LAN-local suppression in the case that the PGM parent is several hops upstream.

2.3 Forward error correction

PGM supports packet-level Reed-Solomon based erasure correction as described in [8, 12, 13]. Linear block codes are used that generate h parity packets from k original data packets such that any k of the total ($h+k$) packets may be decoded to obtain the original k packets. The original k packets are referred to as the transmission group. FEC may be proactive, that is, FEC packets may be sent in anticipation of possible losses, without waiting for any feedback from receivers. FEC may also be on-demand, that is, generated at the request of the receivers. To obtain on-demand FEC, a receiver generates a parity NAK, which indicates the number of parity packets requested from a given transmission group. Both proactive and on-demand FEC are optional, and may be combined at the discretion of the sender. A parity NAK suppresses another NAK for the same transmission group if it requests an equal or greater number of parity packets.

Use of FEC provides a number of advantages:

- efficient retransmission (one parity message can repair several different lost messages);

- more efficient suppression (a NAK resulting from a loss of one packet may suppress the NAK for some other packet, if they are in the same transmission group);

- more efficient usage of NAK bandwidth (e.g., it is more compact to represent "30 packets lost from group 999" than listing those 30 packets).

2.4 Transmit window

In a unicast transfer, it is obvious that the trailing edge of the sender's transmit window should advance to the point at which all previous packets have been acknowledged. For PGM, with no positive acknowledgement, this is not feasible. PGM allows the trailing edge of the transmit window to be arbitrarily advanced by the sender. This allows the application to determine that data has become stale and may be safely dropped out of the window, or to decide that it has enough confidence from a period of NAK silence to believe that it is safe to advance the window.

However, while some application writers may desire this full flexibility to do application-level window advancement, others may not want the burden of window management. Some simple and widely applicable strategies supported in early API development include window advancement by time or size. Advancing by time keeps a packet in the transmit window for some fixed time, e.g., the window contains any packets sent in the last five seconds. Advancing by size fixes the

size of the transmit window and advances the trailing edge as necessary, e.g., the window is a one MB FIFO queue.

2.5 Local repair: DLRs

In addition to having the source provide repairs, PGM supports optional Designated Local Repairers (DLRs) to repair losses. DLRs are host nodes, not NEs. DLRs announce their presence with a multicast message, causing subsequent NAKs to be redirected to the DLR rather than the source. DLRs receive and cache all packets in the transmit window. They then perform NCFs and repairs just as the sender would.

It is obvious how a DLR that is upstream of a NE inserts itself in place of the sender. A less obvious scenario also supported by PGM is "off-tree" DLRs which are one PGM hop downstream from an NE. They are called off-tree because even though they are downstream from the point of loss, they may belong to a sub-tree that was unaffected by the loss. An off-tree DLR responds to a NAK by unicasting an NCF and the repair data to its parent. The parent then forwards the repair downstream.

Off-tree DLRs are discovered via the PGM polling mechanism (see next section). PGM NEs transmit multicast polls (to the session's group address) and hosts capable of being DLRs respond with a message containing redirecting information. These redirecting messages are recorded by the NEs and used for redirecting NAKs.

2.6 Polling

Random delays preceding a NAK should be proportional to the number of PGM siblings. This prevents implosion when scaling up. Often overlooked but just as important is scaling down; that is, reducing delays as the number of siblings decreases so that PGM does not have unnecessarily sluggish performance for small groups. PGM supports polling to allow a PGM parent to estimate the number of its children. Based on the results of the poll, and experience with the session, (e.g., noting when the parent has been overwhelmed by NAKs) the parent uses a field in SPM messages to indicate to its children the interval for random delay.

A poll is performed by sending a multicast POLL packet. POLL packets are not forwarded by PGM NEs as they only apply to a parent's children. Like NCFs, they always have an IP source address of the PGM source so that they will use the same multicast tree. Each POLL packet indicates a probability with which a child should respond to the poll with a unicast poll response message. It also contains an interval for random back-off that children must observe before replying. Furthermore, each poll proceeds in a number of rounds. Only children that were part of the first round of a poll (i.e., received the POLL packet for the first round) may respond to later rounds of a poll.

The probability of response combined with round numbering can be used to avoid implosion in the case of a sudden massive increase in child population. The first round of a poll can be set with a very low probability of response, so that even with millions of children, implosion will not occur. Subsequent rounds can increase the probability, until enough responses are obtained to get a reasonable estimate of the number of children. If many children join the session after the probability of response has increased, they will not respond to the poll (having not been present in the first round) and implosion will be avoided.

Sessions that desire increased protection from implosion can use a packet option (NAK_BO_IVL_SQN) to ensure that only receivers who have been part of a poll are allowed to NAK. This gives the parent the security of knowing that the NAK back-off interval reflects the number of children. However, because polling in rounds with response back-off can take

significant time, there is a price to be paid in repair time as new receiver's join. Senders that can safely assume the number of receivers will not suddenly and dramatically grow can allow new receivers to NAK immediately without waiting to take part in a POLL (By always setting NAK_BO_IVL_SQN to zero).

2.7 Congestion control

PGM's basic operating mode simply rate-limits the sender. To perform congestion control the source must receive information about the state of the transmission. To this end, PGM supports three types of feedback to the source: (1) worst link load as measured by the NEs (2) worst end-to-end path load as measured by the NEs and (3) worst end-to-end path load as reported by receivers. The source uses this feedback to adjust its sending rate. However, PGM does not specify how the rate is to be adjusted; the specific congestion control scheme is left up to the implementer.

3   PGM Performance and Security Issues

In this section, we discuss PGM performance and security issues. We cover network element memory requirements, back-channel bandwidth reduction, network utilization efficiency, and high-speed transmission issues. We also consider security vulnerabilities in PGM.

3.1 Network element memory requirements

PGM requires NEs to store PGM state information. The source path state is small and simple: it suffices to record the source path address from the SPM and indicate what multicast session it applies to. Additionally, and more significantly, a PGM NE must store sequence number information for every outstanding NAK. If many PGM sessions are using a NE, it may have insufficient memory for all the outstanding NAKs from all sessions. In this case, it can simply default to operating transparently as a non-PGM NE for some of the sessions to reduce the memory requirements to a level it can accommodate.[5] This scenario is mostly like to occur in routers located in Internet backbones, where the number of sessions crossing may become extreme. Hence, Internet backbone support of PGM is unlikely. However, as discussed above, PGM works even across non-PGM routers, and the scaling benefits of hierarchy can be realized by the PGM-routers outside of the backbone.

3.2 Back-channel usage

PGM includes a number of options to make efficient use of the back-channel for NAK transmission. First, it supports the OPT_NAK_LIST option to include several individual NAKs within the same NAK packet. A NAK packet for a single loss is 56 bytes, while a packet with the OPT_NAK_LIST option is 64 bytes, plus 4 bytes for each sequence number indicated. Thus, the back-channel traffic can be significantly reduced, as shown in Figure 2

Use of FEC can produce further improvement. If the G packets are in a transmission group, then a loss rate of $1/G$, in the worst case, can lead to a packet being lost from each group, requiring a NAK for one packet from the group. Any higher loss rate would not increase the number of NAKs, as every group is already being reported as having a loss; the loss count is simply higher. For example, with G=8, the maximum number of NAKs will peak at a loss of 12.5%. (Second

---

[5] PGM sessions can be shed and/or rejected when memory falls below certain thresholds. However, the memory requirements will vary according to the number of NAKs outstanding at any given moment, so it is impossible to predict exactly how many sessions can be supported by a given NE. The precise mechanism is left up to PGM NE implementers.

order effects of lost NAK packets and lost parity may increase the NAK bandwidth, but we are concerned with first order effects for the purposes of this discussion.)

Figure 2 illustrates how the use of FEC caps the NAK traffic. In this example, using OPT_NAK_LIST with no FEC is always better than using FEC with no OPT_NAK_LIST. However, with a larger group size, e.g., *G*=64 as in Figure 3, the lines cross and FEC outperforms OPT_NAK_LIST for loss rates above about 20%. Using both FEC and the OPT_NAK_LIST option is always minimizes the NAK back traffic.
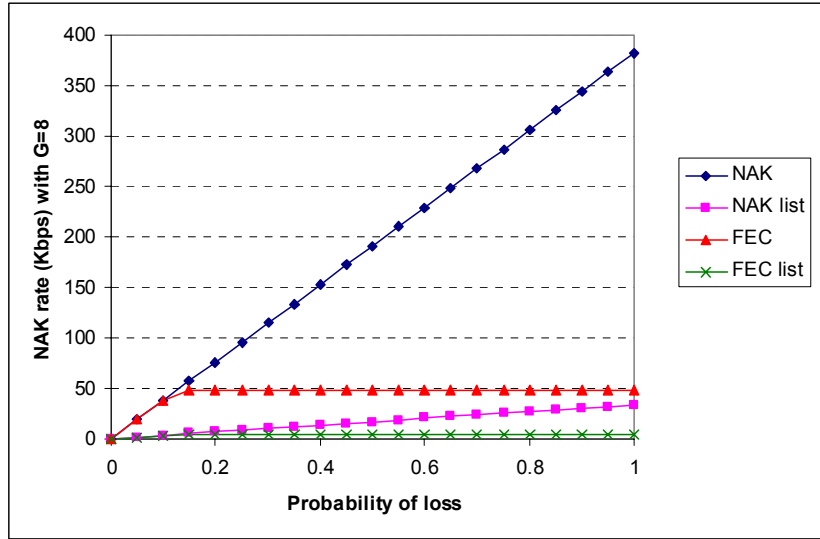

Figure 2 – NAK rate for a single receiver vs. probability of loss. 1500 byte packets are sent 874 times per second for a data rate of approximately 10 Mbps. A group size of 8 is used for FEC.
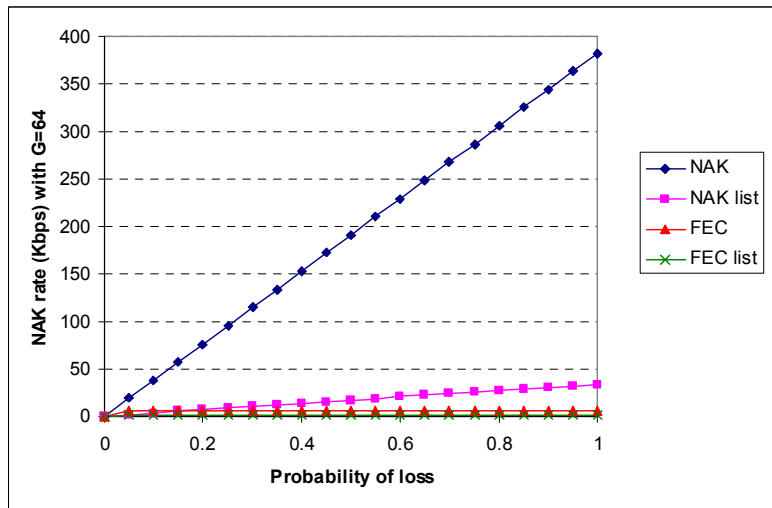

Figure 3 – NAK rate for a single receiver vs. probability of loss. 1500 byte packets are sent 874 times per second for a data rate of approximately 10 Mbps. A group size of 64 is used for FEC.

The use of FEC also reduces back-channel traffic by improving suppression. This can be observed when considering the NAK traffic flowing back to the sender. Given a large enough number of receivers, even with a very low probability of loss, the odds are that every packet is lost by some receiver and needs to be NAKed. For example, a random, independent loss of 0.01%

over 1,000,000 receivers implies that the odds of all nodes receiving a given packet are less than 10^{-43}. As the loss at any given receiver in this example is small, NAKs would be individual (there would not be enough from a single receiver at one point in time to use OPT_NAK_LIST). Therefore, with a data rate of 10 Mbps, 1500 byte packets and no FEC, one would expect 382 Kbps of NAK traffic at the sender. In contrast, using FEC with a group size of 8 could potentially reduce the NAK traffic arriving at the sender to as low as 47.8 Kbps (this stream of parity NAKs is enough to suppress all other single-loss NAKs).[6]

3.3 Network utilization

PGM and IP header overhead limit the network utilization of PGM to 97.1%. A further factor in network utilization is SPM messages, which are typically sent about once per second. When the data rate is high, 10 Mbps or more, the impact on network utilization is negligible. Even at dialup data rates, and increasing SPMs to 5 per second, the network utilization is still above 90%. Table 1 shows the network utilization for different rates of data and SPMs.

Data rate	SPM/sec	Network utilization
Any	0	97.1%
10 Mb/s	1	97.1%
10 Mb/s	5	97.0%
47 Kb/s	1	95.9%
47 Kb/s	5	91.7%

Table 1: Impact of SPMs on network utilization

Independent losses and FEC can also impact network utilization. Consider again the example of 1,000,000 receivers with independent loss of 0.01% so that every packet is likely to be lost by some receiver. Without FEC, every packet must be sent twice (for simplicity, assume that repair packets are not lost). If the data rate is 10 Mb/s with 1500 byte packets, the network utilization is only 48.5%. With FEC and group size of 8, repair traffic would only be 1/8 of original traffic, increasing utilization to 86.3%.

3.4 High-speed transmission issues

We now turn our attention to high-speed transmission using PGM. To begin with, a PGM implementation cannot send at high rates if the host system is incapable of sending raw IP packets at the same rate. This requires suitable NICs, network buffer settings and kernel settings.

Additionally, the sender's buffer for the PGM transmission window is no small matter at high rates. A transmission window of 30 seconds at 100 Mbps transmission rate requires over 375 MB of buffer space. If the transmission buffer is not held entirely in physical memory, page faults can severely degrade performance. To ensure high speed operation, it may be necessary to require enough RAM to hold the entire transmission window.

When FEC is employed, the receiver may also have similar memory requirements. Consider the case where a packet is lost from each group in the transmit window. All *k* packets in each group will be needed for decoding, so they must be kept until group reception is complete. Thus, the receiver may need to keep up to *(k-1)/k* of the transmit window. Once again, if these packets are not in memory, then performance may be severely degraded.

---

[6] In practice, both these figures could be slightly higher because suppression is not perfect in practice.

A PGM receiver must prevent loss due to buffer overrun whenever possible, because PGM uses raw IP and does not have any built-in flow control. If a sender is sending too fast, the receivers buffers are overrun, loss occurs, and NAKs are seen. However, a sender cannot tell if the loss is due to receiver buffer overrun or from network congestion. The reaction to network congestion, such as with PGMCC [15], should be handled differently than loss due to buffer overrun in terms of adjusting the sender's rate. To reduce the amount of buffer overrun at the receiver, the receiver application needs to be aware of the processing load required to read PGM packets from the network and to allow PGM to do so with a high priority compared to other application level tasks. In addition, the operating system socket buffers normally require resizing to handle the capacities that PGM traffic can reach from a well tuned sender.

As of this writing, implementations of PGM on Linux, Solaris, Windows 2000, and Windows XP have sent and received at speeds saturating a 100 Mbps Ethernet switch (about 90 Mbps) with over 250 receivers. We expect at least 400 Mbps to be attainable, based on observed processor load.

3.5 Security Considerations

In addition to the usual problems of end-to-end authentication, PGM is vulnerable to a number of security risks. Without full authentication of all neighboring sources, receivers, DLRs, and NEs, the protocol may be abused by use of SPMs, NAKs, NCFs, and RDATA messages. For example:

- False SPMs may cause PGM network elements to mis-direct NAKs, preventing repairs from being generated.

- False NAKs may cause PGM network elements to establish spurious repair state that will expire only upon time-out and could lead to memory exhaustion in the meantime.

- False NCFs may cause PGM network elements to suspend NAK forwarding prematurely resulting eventually in loss of repairs.

- False repair packets may cause PGM network elements to tear down legitimate repair state resulting eventually in loss of legitimate repairs.

The development of precautions against such attacks remains for future work. Some protections are readily apparent, such as: (1) Damping changes in the sender address and PGM parent in SPMs (the sender address should only change very infrequently and the PGM parent should only change occasionally, as the underlying multicast routing changes). (2) NEs could protect themselves from sessions generating an excessive number of NAKs by dropping the session. (3) A three-way handshake between NEs and DLRs would permit an NE to ascertain with greater confidence that an alleged DLR is identified by the alleged network-header source address, and is PGM conversant.

4 Related Work

Many other reliable multicast researchers have investigated the use of hierarchy, for example [3, 4, 5, 6, 7]. Suppression was pioneered by Ramakrishnan and Jain [2] and popularized for reliable multicast by the SRM protocol [1]. Integration of FEC and reliability was first proposed by Metzner [10]. PGM's use of NAK lists and FEC with suppression is based on ECSRM [8], which is similar to the work of Nonnenmacher et. al. [12]. The default erasure codes in PGM are due to Rizzo [13]. Kermode's simulations demonstrate the benefits of integrating FEC, suppression and hierarchy [9]. Bolot et. al. [16] proposed polling for feedback in a multicast session.

A TCP-friendly congestion control scheme suitable for PGM has been proposed by Rizzo [15]. It does not require any modification to PGM NEs.

Papadopoulos and Laliotis studied the incremental deployment of LMS (a protocol with key similarities to PGM) and demonstrate the scalability gains of even partial deployment among routers [17]. They conclude that PGM would see similar gains.[7]

Generic Router Assist (GRA) is a proposal that generalizes some of the concepts in PGM to make them non-protocol-specific [18]. A tree is built of GRA-capable NEs as a subset of the IP multicast tree, just as in PGM. "Filters" are defined with corresponding GRA header fields. When a packet's header matches a filter definition, the some predefined actions are taken. For example, the handling of parity NAKs and parity retransmissions in PGM could be accomplished via GRA filters.

5 Conclusion

PGM is extremely scalable. It has an excellent deployment plan: with no router support, suppression and FEC handle a limited scaling load. As scalability requirements increase, PGM routers can be deployed to improve scalability via hierarchy. The best scaling and performance is achieved when all routers support PGM. PGM employs a novel NAK delay tuning scheme that uses polling. This facilitates scaling down as well as scaling up.

We have shown that PGM has excellent asymmetric support via the use of FEC, NAK lists, and unicast NAKs. We have also explained how PGM achieves high network utilization, even over low-speed (dialup) connections. Experience with PGM shows that it is capable of high-speed (>100 Mbps) operation. We have outlined some implementation issues that must be addressed to achieve such high speeds.

PGM is currently an experimental RFC [14], and has been implemented in both commercial and academic settings. Client/server implementations include those by Talarian (SmartPGM) and Microsoft (Windows XP). Cisco has added PGM support to its routers. Luigi Rizzo has posted a public source implementation of PGM.[8]

Acknowledgements

References

[1]   Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", Proceedings of ACM SIGCOMM '95, Cambridge, MA, August 1995.

---

[7] In particular, NAK aggregation would be identical. They suggest that LMS would outperform PGM in terms of constrained forwarding. However, ideal constrained forwarding is not nearly as critical as implosion prevention. Implosion can destroy a session. Constrained forwarding simply removes some repair traffic from a sub-net where original data traffic was expected, anyhow.

[8] http://info.iet.unipi.it/~luigi/pgm.html

[2]   Ramakrishnan, S. and Jain, B. N., "A Negative Acknowledgement With Periodic Polling Protocol for Multicast over LANs", Proceedings of IEEE Infocom '87, pp. 502-511, March/April 1987.

[3]   Holbrook, H.W., Singhal, S.K., and Cheriton, D.R., "Log-based Receiver-Reliable Multicast for Distributed Interactive Simulation", Proceedings of SIGCOMM '95, Cambridge, MA, August 1995.

[4]   Markopoulou, Athina P., Tobagi, Fouad A., "Hierarchical Reliable Multicast: performance analysis and placement of proxies," Networked group communication 2000, November 2000, pp. 27-36.

[5]   S. Paul, K.K. Sabnani, J.C.-H. Lin, S. Bhattacharyya Reliable Multicast Transport Protocol (RMTP). IEEE Journal on Selected Areas in Communications, 15:3 (Apr 1997) 407-421.

[6]   Radoslavov, Pavlin, Papadopoulos, Christos, Govindan, Ramesh, and Estrin, Deborah,"A Comparison of Application-Level and Router-Assisted Hierarchical Schemes for Reliable Multicast", Proceedings of the IEEE Infocom 2001, Anchorage, Alaska, USA, April 22-26, 2001.

[7]   Yavatkar, R, Griffioen, J, Sudan, M, "A Reliable Dissemination Protocol for Interactive Collaborative Applications", Proceedings of ACM Multimedia 95, Pages 333-343, November 1995.

[8]   J. Gemmell, Scalable Reliable Multicast Using Erasure-Correcting Re-sends. Technical Report, MSR-TR-97-20, Microsoft Research, Redmond, WA (June 1997).

[9]   R. Kermode, Smart Network Caches: Localized Content and Application Negotiated Recovery Mechanisms for Multicast Media Distribution, PhD Dissertation, MIT, 1998.

[10]   Metzner, J., An Improved Broadcast Retransmission Protocol, IEEE Transactions on Communications, Vol. 32, No. 6, June 1984, pp. 679-683.

[11]   Nonnenmacher, J. and Biersack, E. W. "Reliable multicast: Where to use FEC", Proceedings of IFIP 5th International Workshop on Protocols for High Speed Networks (PfHSN'96), (INRIA, Sophia Antipolis, FRANCE), IFIP, Chapman & Hall, October 1996, pp. 134-148.

[12]   J. Nonnenmacher, E. Biersack, D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission", ACM SIGCOMM, September 1997.

[13]   L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", Computer Communication Review, April 1997.

[14]   Speakman, T., Crowcroft, J., Gemmell, J., Farinacci, D. , Lin, S., Leshchiner, D., Luby, M., Montgomery, T. , Rizzo, L., Tweedly, A., Bhaskar, N., Edmonstone, R., Sumanasekera, R., Vicisano, L., PGM Reliable Transport Protocol Specification, RFC 3208, December 2001. ftp://ftp.rfc-editor.org/in-notes/rfc3208.txt

[15]   L. Rizzo, "pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme", Proc. of ACM SIGCOMM August 2000.

[16]   Bolot, J.C., Turletti, T., and Wakeman, I., Scalable feedback control for multicast video distribution in the Internet', Proceedings of ACM SIGCOMM '94, Oct. 1994, pp. 58-67.

[17]   Papadopoulos, Christos and Laliotis, Emmanouil, "Incremental deployment of a router-assisted reliable multicast scheme", Proceedings of NGC 2000 on Networked group communication, November 8 - 10, 2000, Stanford, CA USA

[18]   Brad Cain, Tony Speakman, and Don Towsley, "Generic Router Assist (GRA) Building Block Motivation and Architecture", IETF draft draft-ietf-rmt-gra-arch-02.txt, July 2001 (expired Jan 2002).