Network Working Group   T. Speakman
Request for Comments: 3208   Cisco Systems
Category: Experimental   J. Crowcroft
                                                                 UCL
                                                         J. Gemmell
                                                          Microsoft
                                                        D. Farinacci
                                                    Procket Networks
                                                             S. Lin
                                                    Juniper Networks
                                                      D. Leshchiner
                                                      TIBCO Software
                                                            M. Luby
                                                    Digital Fountain
                                                       T. Montgomery
                                                 Talarian Corporation
                                                           L. Rizzo
                                                   University of Pisa
                                                         A. Tweedly
                                                          N. Bhaskar
                                                       R. Edmonstone
                                                    R. Sumanasekera
                                                        L. Vicisano
                                                       Cisco Systems
                                                       December 2001

              PGM Reliable Transport Protocol Specification

Status of this Memo

Copyright Notice

Abstract

   Pragmatic General Multicast (PGM) is a reliable multicast transport
   protocol for applications that require ordered or unordered,
   duplicate-free, multicast data delivery from multiple sources to
   multiple receivers. PGM guarantees that a receiver in the group
   either receives all data packets from transmissions and repairs, or
   is able to detect unrecoverable data packet loss. PGM is

specifically intended as a workable solution for multicast
applications with basic reliability requirements. Its central design
goal is simplicity of operation with due regard for scalability and
network efficiency.

Table of Contents

Nota Bene:

   The publication of this specification is intended to freeze the
   definition of PGM in the interest of fostering both ongoing and
   prospective experimentation with the protocol. The intent of that
   experimentation is to provide experience with the implementation and
   deployment of a reliable multicast protocol of this class so as to be
   able to feed that experience back into the longer-term
   standardization process underway in the Reliable Multicast Transport
   Working Group of the IETF. Appendix G provides more specific detail
   on the scope and status of some of this experimentation. Reports of
   experiments include [16-23]. Additional results and new
   experimentation are encouraged.

1. Introduction and Overview

   A variety of reliable protocols have been proposed for multicast data
   delivery, each with an emphasis on particular types of applications,
   network characteristics, or definitions of reliability ([1], [2],
   [3], [4]). In this tradition, Pragmatic General Multicast (PGM) is a
   reliable transport protocol for applications that require ordered or
   unordered, duplicate-free, multicast data delivery from multiple
   sources to multiple receivers.

   PGM is specifically intended as a workable solution for multicast
   applications with basic reliability requirements rather than as a
   comprehensive solution for multicast applications with sophisticated
   ordering, agreement, and robustness requirements. Its central design
   goal is simplicity of operation with due regard for scalability and
   network efficiency.

   PGM has no notion of group membership. It simply provides reliable
   multicast data delivery within a transmit window advanced by a source
   according to a purely local strategy. Reliable delivery is provided
   within a source's transmit window from the time a receiver joins the
   group until it departs. PGM guarantees that a receiver in the group
   either receives all data packets from transmissions and repairs, or
   is able to detect unrecoverable data packet loss. PGM supports any
   number of sources within a multicast group, each fully identified by
   a globally unique Transport Session Identifier (TSI), but since these
   sources/sessions operate entirely independently of each other, this
   specification is phrased in terms of a single source and extends
   without modification to multiple sources.

   More specifically, 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.

   Rather, PGM is best suited to those applications in which members may
   join and leave at any time, and that are either insensitive to
   unrecoverable data packet loss or are prepared to resort to
   application recovery in the event. Through its optional extensions,
   PGM provides specific mechanisms to support applications as disparate
   as stock and news updates, data conferencing, low-delay real-time
   video transfer, and bulk data transfer.

   In the following text, transport-layer originators of PGM data
   packets are referred to as sources, transport-layer consumers of PGM
   data packets are referred to as receivers, and network-layer entities
   in the intervening network are referred to as network elements.

Unless otherwise specified, the term "repair" will be used to
indicate both the actual retransmission of a copy of a missing packet
or the transmission of an FEC repair packet.

Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [14] and
indicate requirement levels for compliant PGM implementations.

1.1. Summary of Operation

PGM runs over a datagram multicast protocol such as IP multicast [5].
In the normal course of data transfer, a source multicasts sequenced
data packets (ODATA), and receivers unicast selective negative
acknowledgments (NAKs) for data packets detected to be missing from
the expected sequence. Network elements forward NAKs PGM-hop-by-
PGM-hop to the source, and confirm each hop by multicasting a NAK
confirmation (NCF) in response on the interface on which the NAK was
received. Repairs (RDATA) may be provided either by the source
itself or by a Designated Local Repairer (DLR) in response to a NAK.

Since NAKs provide the sole mechanism for reliability, PGM is
particularly sensitive to their loss. To minimize NAK loss, PGM
defines a network-layer hop-by-hop procedure for reliable NAK
forwarding.

Upon detection of a missing data packet, a receiver repeatedly
unicasts a NAK to the last-hop PGM network element on the
distribution tree from the source. A receiver repeats this NAK until
it receives a NAK confirmation (NCF) multicast to the group from that
PGM network element. That network element responds with an NCF to
the first occurrence of the NAK and any further retransmissions of
that same NAK from any receiver. In turn, the network element
repeatedly forwards the NAK to the upstream PGM network element on
the reverse of the distribution path from the source of the original
data packet until it also receives an NCF from that network element.
Finally, the source itself receives and confirms the NAK by
multicasting an NCF to the group.

While NCFs are multicast to the group, they are not propagated by PGM
network elements since they act as hop-by-hop confirmations.

To avoid NAK implosion, PGM specifies procedures for subnet-based NAK suppression amongst receivers and NAK elimination within network elements. The usual result is the propagation of just one copy of a given NAK along the reverse of the distribution path from any network with directly connected receivers to a source.

The net effect is that unicast NAKs return from a receiver to a source on the reverse of the path on which ODATA was forwarded, that is, on the reverse of the distribution tree from the source. More specifically, they return through exactly the same sequence of PGM network elements through which ODATA was forwarded, but in reverse. The reasons for handling NAKs this way will become clear in the discussion of constraining repairs, but first it's necessary to describe the mechanisms for establishing the requisite source path state in PGM network elements.

To establish source path state in PGM network elements, the basic data transfer operation is augmented by Source Path Messages (SPMs) from a source, periodically interleaved with ODATA. SPMs function primarily to establish source path state for a given TSI in all PGM network elements on the distribution tree from the source. PGM network elements use this information to address returning unicast NAKs directly to the upstream PGM network element toward the source, and thereby insure that NAKs return from a receiver to a source on the reverse of the distribution path for the TSI.

SPMs are sent by a source at a rate that serves to maintain up-to-date PGM neighbor information. In addition, SPMs complement the role of DATA packets in provoking further NAKs from receivers, and maintaining receive window state in the receivers.

As a further efficiency, PGM specifies procedures for the constraint of repairs by network elements so that they reach only those network segments containing group members that did not receive the original transmission. As NAKs traverse the reverse of the ODATA path (upward), they establish repair state in the network elements which is used in turn to constrain the (downward) forwarding of the corresponding RDATA.

Besides procedures for the source to provide repairs, PGM also specifies options and procedures that permit designated local repairers (DLRs) to announce their availability and to redirect repair requests (NAKs) to themselves rather than to the original source. In addition to these conventional procedures for loss recovery through selective ARQ, Appendix A specifies Forward Error Correction (FEC) procedures for sources to provide and receivers to request general error correcting parity packets rather than selective retransmissions.

Finally, since PGM operates without regular return traffic from
receivers, conventional feedback mechanisms for transport flow and
congestion control cannot be applied. Appendix B specifies a TCP-
friendly, NE-based solution for PGM congestion control, and cites a
reference to a TCP-friendly, end-to-end solution for PGM congestion
control.

In its basic operation, PGM relies on a purely rate-limited
transmission strategy in the source to bound the bandwidth consumed
by PGM transport sessions and to define the transmit window
maintained by the source.

PGM defines four basic packet types: three that flow downstream
(SPMs, DATA, NCFs), and one that flows upstream (NAKs).

1.2. Design Goals and Constraints

PGM has been designed to serve that broad range of multicast
applications that have relatively simple reliability requirements,
and to do so in a way that realizes the much advertised but often
unrealized network efficiencies of multicast data transfer. The
usual impediments to realizing these efficiencies are the implosion
of negative and positive acknowledgments from receivers to sources,
repair latency from the source, and the propagation of repairs to
disinterested receivers.

1.2.1. Reliability.

Reliable data delivery across an unreliable network is conventionally
achieved through an end-to-end protocol in which a source (implicitly
or explicitly) solicits receipt confirmation from a receiver, and the
receiver responds positively or negatively. While the frequency of
negative acknowledgments is a function of the reliability of the
network and the receiver's resources (and so, potentially quite low),
the frequency of positive acknowledgments is fixed at at least the
rate at which the transmit window is advanced, and usually more
often.

Negative acknowledgments primarily determine repairs and reliability.
Positive acknowledgments primarily determine transmit buffer
management.

When these principles are extended without modification to multicast
protocols, the result, at least for positive acknowledgments, is a
burden of positive acknowledgments transmitted to the source that
quickly threatens to overwhelm it as the number of receivers grows.
More succinctly, ACK implosion keeps ACK-based reliable multicast
protocols from scaling well.

One of the goals of PGM is to get as strong a definition of
reliability as possible from as simple a protocol as possible. ACK
implosion can be addressed in a variety of effective but complicated
ways, most of which require re-transmit capability from other than
the original source.

An alternative is to dispense with positive acknowledgments
altogether, and to resort to other strategies for buffer management
while retaining negative acknowledgments for repairs and reliability.
The approach taken in PGM is to retain negative acknowledgments, but
to dispense with positive acknowledgments and resort instead to
timeouts at the source to manage transmit resources.

The definition of reliability with PGM is a direct consequence of
this design decision. PGM guarantees that a receiver either receives
all data packets from transmissions and repairs, or is able to detect
unrecoverable data packet loss.

PGM includes strategies for repeatedly provoking NAKs from receivers,
and for adding reliability to the NAKs themselves. By reinforcing
the NAK mechanism, PGM minimizes the probability that a receiver will
detect a missing data packet so late that the packet is unavailable
for repair either from the source or from a designated local repairer
(DLR). Without ACKs and knowledge of group membership, however, PGM
cannot eliminate this possibility.

1.2.2. Group Membership

A second consequence of eliminating ACKs is that knowledge of group
membership is neither required nor provided by the protocol.
Although a source may receive some PGM packets (NAKs for instance)
from some receivers, the identity of the receivers does not figure in
the processing of those packets. Group membership MAY change during
the course of a PGM transport session without the knowledge of or
consequence to the source or the remaining receivers.

1.2.3. Efficiency

While PGM avoids the implosion of positive acknowledgments simply by
dispensing with ACKs, the implosion of negative acknowledgments is
addressed directly.

Receivers observe a random back-off prior to generating a NAK during
which interval the NAK is suppressed (i.e. it is not sent, but the
receiver acts as if it had sent it) by the receiver upon receipt of a
matching NCF. In addition, PGM network elements eliminate duplicate
NAKs received on different interfaces on the same network element.

The combination of these two strategies usually results in the source
receiving just a single NAK for any given lost data packet.

Whether a repair is provided from a DLR or the original source, it is
important to constrain that repair to only those network segments
containing members that negatively acknowledged the original
transmission rather than propagating it throughout the group. PGM
specifies procedures for network elements to use the pattern of NAKs
to define a sub-tree within the group upon which to forward the
corresponding repair so that it reaches only those receivers that
missed it in the first place.

1.2.4. Simplicity

PGM is designed to achieve the greatest improvement in reliability
(as compared to the usual UDP) with the least complexity. As a
result, PGM does NOT address conference control, global ordering
amongst multiple sources in the group, nor recovery from network
partitions.

1.2.5. Operability

PGM is designed to function, albeit with less efficiency, even when
some or all of the network elements in the multicast tree have no
knowledge of PGM. To that end, all PGM data packets can be
conventionally multicast routed by non-PGM network elements with no
loss of functionality, but with some inefficiency in the propagation
of RDATA and NCFs.

In addition, since NAKs are unicast to the last-hop PGM network
element and NCFs are multicast to the group, NAK/NCF operation is
also consistent across non-PGM network elements. Note that for NAK
suppression to be most effective, receivers should always have a PGM
network element as a first hop network element between themselves and
every path to every PGM source. If receivers are several hops
removed from the first PGM network element, the efficacy of NAK
suppression may degrade.

1.3. Options

In addition to the basic data transfer operation described above, PGM
specifies several end-to-end options to address specific application
requirements. PGM specifies options to support fragmentation, late
joining, redirection, Forward Error Correction (FEC), reachability,
and session synchronization/termination/reset. Options MAY be
appended to PGM data packet headers only by their original
transmitters. While they MAY be interpreted by network elements,
options are neither added nor removed by network elements.

All options are receiver-significant (i.e., they must be interpreted
by receivers). Some options are also network-significant (i.e., they
must be interpreted by network elements).

Fragmentation MAY be used in conjunction with data packets to allow a
transport-layer entity at the source to break up application-layer
data packets into multiple PGM data packets to conform with the
maximum transmission unit (MTU) supported by the network layer.

Late joining allows a source to indicate whether or not receivers may
request all available repairs when they initially join a particular
transport session.

Redirection MAY be used in conjunction with Poll Responses to allow a
DLR to respond to normal NCFs or POLLs with a redirecting POLR
advertising its own address as an alternative re-transmitter to the
original source.

FEC techniques MAY be applied by receivers to use source-provided
parity packets rather than selective retransmissions to effect loss
recovery.

2. Architectural Description

As an end-to-end transport protocol, PGM specifies packet formats and
procedures for sources to transmit and for receivers to receive data.
To enhance the efficiency of this data transfer, PGM also specifies
packet formats and procedures for network elements to improve the
reliability of NAKs and to constrain the propagation of repairs. The
division of these functions is described in this section and expanded
in detail in the next section.

2.1. Source Functions

Data Transmission

Sources multicast ODATA packets to the group within the
transmit window at a given transmit rate.

Source Path State

Sources multicast SPMs to the group, interleaved with ODATA if
present, to establish source path state in PGM network
elements.

NAK Reliability

   Sources multicast NCFs to the group in response to any NAKs
   they receive.

Repairs

   Sources multicast RDATA packets to the group in response to
   NAKs received for data packets within the transmit window.

Transmit Window Advance

   Sources MAY advance the trailing edge of the window according
   to one of a number of strategies. Implementations MAY support
   automatic adjustments such as keeping the window at a fixed
   size in bytes, a fixed number of packets or a fixed real time
   duration. In addition, they MAY optionally delay window
   advancement based on NAK-silence for a certain period. Some
   possible strategies are outlined later in this document.

2.2. Receiver Functions

Source Path State

   Receivers use SPMs to determine the last-hop PGM network
   element for a given TSI to which to direct their NAKs.

Data Reception

   Receivers receive ODATA within the transmit window and
   eliminate any duplicates.

Repair Requests

   Receivers unicast NAKs to the last-hop PGM network element (and
   MAY optionally multicast a NAK with TTL of 1 to the local
   group) for data packets within the receive window detected to
   be missing from the expected sequence. A receiver MUST
   repeatedly transmit a given NAK until it receives a matching
   NCF.

NAK Suppression

   Receivers suppress NAKs for which a matching NCF or NAK is
   received during the NAK transmit back-off interval.

Receive Window Advance

Receivers immediately advance their receive windows upon
receipt of any PGM data packet or SPM within the transmit
window that advances the receive window.

2.3. Network Element Functions

Network elements forward ODATA without intervention.

Source Path State

Network elements intercept SPMs and use them to establish
source path state for the corresponding TSI before multicast
forwarding them in the usual way.

NAK Reliability

Network elements multicast NCFs to the group in response to any
NAK they receive. For each NAK received, network elements
create repair state recording the transport session identifier,
the sequence number of the NAK, and the input interface on
which the NAK was received.

Constrained NAK Forwarding

Network elements repeatedly unicast forward only the first copy
of any NAK they receive to the upstream PGM network element on
the distribution path for the TSI until they receive an NCF in
response. In addition, they MAY optionally multicast this NAK
upstream with TTL of 1.

Nota Bene: Once confirmed by an NCF, network elements discard NAK
packets; NAKs are NOT retained in network elements beyond this
forwarding operation, but state about the reception of them is
stored.

NAK Elimination

Network elements discard exact duplicates of any NAK for which
they already have repair state (i.e., that has been forwarded
either by themselves or a neighboring PGM network element), and
respond with a matching NCF.

Constrained RDATA Forwarding

Network elements use NAKs to maintain repair state consisting
of a list of interfaces upon which a given NAK was received,
and they forward the corresponding RDATA only on these
interfaces.

NAK Anticipation

If a network element hears an upstream NCF (i.e., on the
upstream interface for the distribution tree for the TSI), it
establishes repair state without outgoing interfaces in
anticipation of responding to and eliminating duplicates of the
NAK that may arrive from downstream.

3. Terms and Concepts

Before proceeding from the preceding overview to the detail in the
subsequent Procedures, this section presents some concepts and
definitions that make that detail more intelligible.

3.1. Transport Session Identifiers

Every PGM packet is identified by a:

TSI   transport session identifier

TSIs MUST be globally unique, and only one source at a time may act
as the source for a transport session. (Note that repairers do not
change the TSI in any RDATA they transmit). TSIs are composed of the
concatenation of a globally unique source identifier (GSI) and a
source-assigned data-source port.

Since all PGM packets originated by receivers are in response to PGM
packets originated by a source, receivers simply echo the TSI heard
from the source in any corresponding packets they originate.

Since all PGM packets originated by network elements are in response
to PGM packets originated by a receiver, network elements simply echo
the TSI heard from the receiver in any corresponding packets they
originate.

3.2. Sequence Numbers

PGM uses a circular sequence number space from 0 through ((2**32) -
1) to identify and order ODATA packets. Sources MUST number ODATA
packets in unit increments in the order in which the corresponding
application data is submitted for transmission. Within a transmit or

receive window (defined below), a sequence number x is "less" or
"older" than sequence number y if it numbers an ODATA packet
preceding ODATA packet y, and a sequence number y is "greater" or
"more recent" than sequence number x if it numbers an ODATA packet
subsequent to ODATA packet x.

3.3. Transmit Window

The description of the operation of PGM rests fundamentally on the
definition of the source-maintained transmit window. This definition
in turn is derived directly from the amount of transmitted data (in
seconds) a source retains for repair (TXW_SECS), and the maximum
transmit rate (in bytes/second) maintained by a source to regulate
its bandwidth utilization (TXW_MAX_RTE).

In terms of sequence numbers, the transmit window is the range of
sequence numbers consumed by the source for sequentially numbering
and transmitting the most recent TXW_SECS of ODATA packets. The
trailing (or left) edge of the transmit window (TXW_TRAIL) is defined
as the sequence number of the oldest data packet available for repair
from a source. The leading (or right) edge of the transmit window
(TXW_LEAD) is defined as the sequence number of the most recent data
packet a source has transmitted.

The size of the transmit window in sequence numbers (TXW_SQNS) (i.e.,
the difference between the leading and trailing edges plus one) MUST
be no greater than half the PGM sequence number space less one.

When TXW_TRAIL is equal to TXW_LEAD, the transmit window size is one.
When TXW_TRAIL is equal to TXW_LEAD plus one, the transmit window
size is empty.

3.4. Receive Window

The receive window at the receivers is determined entirely by PGM
packets from the source. That is, a receiver simply obeys what the
source tells it in terms of window state and advancement.

For a given transport session identified by a TSI, a receiver
maintains:

RXW_TRAIL   the sequence number defining the trailing edge of the
                 receive window, the sequence number (known from data
                 packets and SPMs) of the oldest data packet available
                 for repair from the source

RXW_LEAD   the sequence number defining the leading edge of the
                  receive window, the greatest sequence number of any
                  received data packet within the transmit window

The receive window is the range of sequence numbers a receiver is
expected to use to identify receivable ODATA.

A data packet is described as being "in" the receive window if its
sequence number is in the receive window.

The receive window is advanced by the receiver when it receives an
SPM or ODATA packet within the transmit window that increments
RXW_TRAIL. Receivers also advance their receive windows upon receipt
of any PGM data packet within the receive window that advances the
receive window.

3.5. Source Path State

To establish the repair state required to constrain RDATA, it's
essential that NAKs return from a receiver to a source on the reverse
of the distribution tree from the source. That is, they must return
through the same sequence of PGM network elements through which the
ODATA was forwarded, but in reverse. There are two reasons for this,
the less obvious one being by far the more important.

The first and obvious reason is that RDATA is forwarded on the same
path as ODATA and so repair state must be established on this path if
it is to constrain the propagation of RDATA.

The second and less obvious reason is that in the absence of repair
state, PGM network elements do NOT forward RDATA, so the default
behavior is to discard repairs. If repair state is not properly
established for interfaces on which ODATA went missing, then
receivers on those interfaces will continue to NAK for lost data and
ultimately experience unrecoverable data loss.

The principle function of SPMs is to provide the source path state
required for PGM network elements to forward NAKs from one PGM
network element to the next on the reverse of the distribution tree
for the TSI, establishing repair state each step of the way. This
source path state is simply the address of the upstream PGM network
element on the reverse of the distribution tree for the TSI. That
upstream PGM network element may be more than one subnet hop away.
SPMs establish the identity of the upstream PGM network element on
the distribution tree for each TSI in each group in each PGM network
element, a sort of virtual PGM topology. So although NAKs are
unicast addressed, they are NOT unicast routed by PGM network
elements in the conventional sense. Instead PGM network elements use

the source path state established by SPMs to direct NAKs PGM-hop-by-
PGM-hop toward the source. The idea is to constrain NAKs to the pure
PGM topology spanning the more heterogeneous underlying topology of
both PGM and non-PGM network elements.

The result is repair state in every PGM network element between the
receiver and the source so that the corresponding RDATA is never
discarded by a PGM network element for lack of repair state.

SPMs also maintain transmit window state in receivers by advertising
the trailing and leading edges of the transmit window (SPM_TRAIL and
SPM_LEAD). In the absence of data, SPMs MAY be used to close the
transmit window in time by advancing the transmit window until
SPM_TRAIL is equal to SPM_LEAD plus one.

3.6. Packet Contents

This section just provides enough short-hand to make the Procedures
intelligible. For the full details of packet contents, please refer
to Packet Formats below.

3.6.1. Source Path Messages

3.6.1.1. SPMs

SPMs are transmitted by sources to establish source-path state in PGM
network elements, and to provide transmit-window state in receivers.

SPMs are multicast to the group and contain:

SPM_TSI   the source-assigned TSI for the session to which the
                 SPM corresponds

SPM_SQN   a sequence number assigned sequentially by the source
                 in unit increments and scoped by SPM_TSI

   Nota Bene: this is an entirely separate sequence than is used to
   number ODATA and RDATA.

SPM_TRAIL   the sequence number defining the trailing edge of the
                 source's transmit window (TXW_TRAIL)

SPM_LEAD   the sequence number defining the leading edge of the
                 source's transmit window (TXW_LEAD)

SPM_PATH   the network-layer address (NLA) of the interface on
                 the PGM network element on which the SPM is forwarded

3.6.2. Data Packets

3.6.2.1. ODATA - Original Data

   ODATA packets are transmitted by sources to send application data to
   receivers.

   ODATA packets are multicast to the group and contain:

   OD_TSI   the globally unique source-assigned TSI

   OD_TRAIL   the sequence number defining the trailing edge of the
                   source's transmit window (TXW_TRAIL)

                   OD_TRAIL makes the protocol more robust in the face of
                   lost SPMs. By including the trailing edge of the
                   transmit window on every data packet, receivers that
                   have missed any SPMs that advanced the transmit window
                   can still detect the case, recover the application,
                   and potentially re-synchronize to the transport
                   session.

   OD_SQN   a sequence number assigned sequentially by the source
                   in unit increments and scoped by OD_TSI

3.6.2.2. RDATA - Repair Data

   RDATA packets are repair packets transmitted by sources or DLRs in
   response to NAKs.

   RDATA packets are multicast to the group and contain:

   RD_TSI   OD_TSI of the ODATA packet for which this is a repair

   RD_TRAIL   the sequence number defining the trailing edge of the
                   source's transmit window (TXW_TRAIL). This is updated
                   to the most current value when the repair is sent, so
                   it is not necessarily the same as OD_TRAIL of the
                   ODATA packet for which this is a repair

   RD_SQN   OD_SQN of the ODATA packet for which this is a repair

3.6.3. Negative Acknowledgments

3.6.3.1. NAKs - Negative Acknowledgments

   NAKs are transmitted by receivers to request repairs for missing data
   packets.

NAKs are unicast (PGM-hop-by-PGM-hop) to the source and contain:

NAK_TSI   OD_TSI of the ODATA packet for which a repair is
                 requested

NAK_SQN   OD_SQN of the ODATA packet for which a repair is
                 requested

NAK_SRC   the unicast NLA of the original source of the missing
                 ODATA.

NAK_GRP   the multicast group NLA

3.6.3.2. NNAKs - Null Negative Acknowledgments

NNAKs are transmitted by a DLR that receives NAKs redirected to it by
either receivers or network elements to provide flow-control feed-
back to a source.

NNAKs are unicast (PGM-hop-by-PGM-hop) to the source and contain:

NNAK_TSI   NAK_TSI of the corresponding re-directed NAK.

NNAK_SQN   NAK_SQN of the corresponding re-directed NAK.

NNAK_SRC   NAK_SRC of the corresponding re-directed NAK.

NNAK_GRP   NAK_GRP of the corresponding re-directed NAK.

3.6.4. Negative Acknowledgment Confirmations

3.6.4.1. NCFs - NAK confirmations

NCFs are transmitted by network elements and sources in response to
NAKs.

NCFs are multicast to the group and contain:

NCF_TSI   NAK_TSI of the NAK being confirmed

NCF_SQN   NAK_SQN of the NAK being confirmed

NCF_SRC   NAK_SRC of the NAK being confirmed

NCF_GRP   NAK_GRP of the NAK being confirmed

3.6.5. Option Encodings

    OPT_LENGTH   0x00 - Option's Length

    OPT_FRAGMENT   0x01 - Fragmentation

    OPT_NAK_LIST   0x02 - List of NAK entries

    OPT_JOIN   0x03 - Late Joining

    OPT_REDIRECT   0x07 - Redirect

    OPT_SYN   0x0D - Synchronization

    OPT_FIN   0x0E - Session Fin   receivers, conventional
                            feedbackish

    OPT_RST   0x0F - Session Reset

    OPT_PARITY_PRM   0x08 - Forward Error Correction Parameters

    OPT_PARITY_GRP   0x09 - Forward Error Correction Group Number

    OPT_CURR_TGSIZE 0x0A - Forward Error Correction Group Size

    OPT_CR   0x10 - Congestion Report

    OPT_CRQST   0x11 - Congestion Report Request

    OPT_NAK_BO_IVL   0x04 - NAK Back-Off Interval

    OPT_NAK_BO_RNG   0x05 - NAK Back-Off Range

    OPT_NBR_UNREACH 0x0B - Neighbor Unreachable

    OPT_PATH_NLA   0x0C - Path NLA

    OPT_INVALID   0x7F - Option invalidated

4. Procedures - General

    Since SPMs, NCFs, and RDATA must be treated conditionally by PGM
    network elements, they must be distinguished from other packets in
    the chosen multicast network protocol if PGM network elements are to
    extract them from the usual switching path.

The most obvious way for network elements to achieve this is to
examine every packet in the network for the PGM transport protocol
and packet types. However, the overhead of this approach is costly
for high-performance, multi-protocol network elements. An
alternative, and a requirement for PGM over IP multicast, is that
SPMs, NCFs, and RDATA MUST be transmitted with the IP Router Alert
Option [6]. This option gives network elements a network-layer
indication that a packet should be extracted from IP switching for
more detailed processing.

5. Procedures - Sources

5.1. Data Transmission

   Since PGM relies on a purely rate-limited transmission strategy in
   the source to bound the bandwidth consumed by PGM transport sessions,
   an assortment of techniques is assembled here to make that strategy
   as conservative and robust as possible. These techniques are the
   minimum REQUIRED of a PGM source.

5.1.1. Maximum Cumulative Transmit Rate

   A source MUST number ODATA packets in the order in which they are
   submitted for transmission by the application. A source MUST
   transmit ODATA packets in sequence and only within the transmit
   window beginning with TXW_TRAIL at no greater a rate than
   TXW_MAX_RTE.

   TXW_MAX_RTE is typically the maximum cumulative transmit rate of SPM,
   ODATA, and RDATA. Different transmission strategies MAY define
   TXW_MAX_RTE as appropriate for the implementation.

5.1.2. Transmit Rate Regulation

   To regulate its transmit rate, a source MUST use a token bucket
   scheme or any other traffic management scheme that yields equivalent
   behavior. A token bucket [7] is characterized by a continually
   sustainable data rate (the token rate) and the extent to which the
   data rate may exceed the token rate for short periods of time (the
   token bucket size). Over any arbitrarily chosen interval, the number
   of bytes the source may transmit MUST NOT exceed the token bucket
   size plus the product of the token rate and the chosen interval.

   In addition, a source MUST bound the maximum rate at which successive
   packets may be transmitted using a leaky bucket scheme drained at a
   maximum transmit rate, or equivalent mechanism.

5.1.3. Outgoing Packet Ordering

   To preserve the logic of PGM's transmit window, a source MUST
   strictly prioritize sending of pending NCFs first, pending SPMs
   second, and only send ODATA or RDATA when no NCFs or SPMs are
   pending. The priority of RDATA versus ODATA is application
   dependent. The sender MAY implement weighted bandwidth sharing
   between RDATA and ODATA. Note that strict prioritization of RDATA
   over ODATA may stall progress of ODATA if there are receivers that
   keep generating NAKs so as to always have RDATA pending (e.g. a
   steady stream of late joiners with OPT_JOIN). Strictly prioritizing
   ODATA over RDATA may lead to a larger portion of receivers getting
   unrecoverable losses.

5.1.4. Ambient SPMs

   Interleaved with ODATA and RDATA, a source MUST transmit SPMs at a
   rate at least sufficient to maintain current source path state in PGM
   network elements. Note that source path state in network elements
   does not track underlying changes in the distribution tree from a
   source until an SPM traverses the altered distribution tree. The
   consequence is that NAKs may go unconfirmed both at receivers and
   amongst network elements while changes in the underlying distribution
   tree take place.

5.1.5. Heartbeat SPMs

   In the absence of data to transmit, a source SHOULD transmit SPMs at
   a decaying rate in order to assist early detection of lost data, to
   maintain current source path state in PGM network elements, and to
   maintain current receive window state in the receivers.

   In this scheme [8], a source maintains an inter-heartbeat timer
   IHB_TMR which times the interval between the most recent packet
   (ODATA, RDATA, or SPM) transmission and the next heartbeat
   transmission. IHB_TMR is initialized to a minimum interval IHB_MIN
   after the transmission of any data packet. If IHB_TMR expires, the
   source transmits a heartbeat SPM and initializes IHB_TMR to double
   its previous value. The transmission of consecutive heartbeat SPMs
   doubles IHB each time up to a maximum interval IHB_MAX. The
   transmission of any data packet initializes IHB_TMR to IHB_MIN once
   again. The effect is to provoke prompt detection of missing packets
   in the absence of data to transmit, and to do so with minimal
   bandwidth overhead.

5.1.6. Ambient and Heartbeat SPMs

   Ambient and heartbeat SPMs are described as driven by separate timers
   in this specification to highlight their contrasting functions.
   Ambient SPMs are driven by a count-down timer that expires regularly
   while heartbeat SPMs are driven by a count-down timer that keeps
   being reset by data, and the interval of which changes once it begins
   to expire. The ambient SPM timer is just counting down in real-time
   while the heartbeat timer is measuring the inter-data-packet
   interval.

   In the presence of data, no heartbeat SPMs will be transmitted since
   the transmission of data keeps setting the IHB_TMR back to its
   initial value. At the same time however, ambient SPMs MUST be
   interleaved into the data as a matter of course, not necessarily as a
   heartbeat mechanism. This ambient transmission of SPMs is REQUIRED
   to keep the distribution tree information in the network current and
   to allow new receivers to synchronize with the session.

   An implementation SHOULD de-couple ambient and heartbeat SPM timers
   sufficiently to permit them to be configured independently of each
   other.

5.2. Negative Acknowledgment Confirmation

   A source MUST immediately multicast an NCF in response to any NAK it
   receives. The NCF is REQUIRED since the alternative of responding
   immediately with RDATA would not allow other PGM network elements on
   the same subnet to do NAK anticipation, nor would it allow DLRs on
   the same subnet to provide repairs. A source SHOULD be able to
   detect a NAK storm and adopt countermeasure to protect the network
   against a denial of service. A possible countermeasure is to send
   the first NCF immediately in response to a NAK and then delay the
   generation of further NCFs (for identical NAKs) by a small interval,
   so that identical NCFs are rate-limited, without affecting the
   ability to suppress NAKs.

5.3. Repairs

   After multicasting an NCF in response to a NAK, a source MUST then
   multicast RDATA (while respecting TXW_MAX_RTE) in response to any NAK
   it receives for data packets within the transmit window.

   In the interest of increasing the efficiency of a particular RDATA
   packet, a source MAY delay RDATA transmission to accommodate the
   arrival of NAKs from the whole loss neighborhood. This delay SHOULD
   not exceed twice the greatest propagation delay in the loss
   neighborhood.

6. Procedures - Receivers

6.1. Data Reception

   Initial data reception

   A receiver SHOULD initiate data reception beginning with the first
   data packet it receives within the advertised transmit window. This
   packet's sequence number (ODATA_SQN) temporarily defines the trailing
   edge of the transmit window from the receiver's perspective. That
   is, it is assigned to RXW_TRAIL_INIT within the receiver, and until
   the trailing edge sequence number advertised in subsequent packets
   (SPMs or ODATA or RDATA) increments past RXW_TRAIL_INIT, the receiver
   MUST only request repairs for sequence numbers subsequent to
   RXW_TRAIL_INIT. Thereafter, it MAY request repairs anywhere in the
   transmit window. This temporary restriction on repair requests
   prevents receivers from requesting a potentially large amount of
   history when they first begin to receive a given PGM transport
   session.

   Note that the JOIN option, discussed later, MAY be used to provide a
   different value for RXW_TRAIL_INIT.

   Receiving and discarding data packets

   Within a given transport session, a receiver MUST accept any ODATA or
   RDATA packets within the receive window. A receiver MUST discard any
   data packet that duplicates one already received in the transmit
   window. A receiver MUST discard any data packet outside of the
   receive window.

   Contiguous data

   Contiguous data is comprised of those data packets within the receive
   window that have been received and are in the range from RXW_TRAIL up
   to (but not including) the first missing sequence number in the
   receive window. The most recently received data packet of contiguous
   data defines the leading edge of contiguous data.

   As its default mode of operation, a receiver MUST deliver only
   contiguous data packets to the application, and it MUST do so in the
   order defined by those data packets' sequence numbers. This provides
   applications with a reliable ordered data flow.

Non contiguous data

PGM receiver implementations MAY optionally provide a mode of
operation in which data is delivered to an application in the order
received. However, the implementation MUST only deliver complete
application protocol data units (APDUs) to the application. That is,
APDUs that have been fragmented into different TPDUs MUST be
reassembled before delivery to the application.

6.2. Source Path Messages

Receivers MUST receive and sequence SPMs for any TSI they are
receiving. An SPM is in sequence if its sequence number is greater
than that of the most recent in-sequence SPM and within half the PGM
number space. Out-of-sequence SPMs MUST be discarded.

For each TSI, receivers MUST use the most recent SPM to determine the
NLA of the upstream PGM network element for use in NAK addressing. A
receiver MUST NOT initiate repair requests until it has received at
least one SPM for the corresponding TSI.

Since SPMs require per-hop processing, it is likely that they will be
forwarded at a slower rate than data, and that they will arrive out
of sync with the data stream. In this case, the window information
that the SPMs carry will be out of date. Receivers SHOULD expect
this to be the case and SHOULD detect it by comparing the packet lead
and trail values with the values the receivers have stored for lead
and trail. If the SPM packet values are less, they SHOULD be
ignored, but the rest of the packet SHOULD be processed as normal.

6.3. Data Recovery by Negative Acknowledgment

Detecting missing data packets

Receivers MUST detect gaps in the expected data sequence in the
following manners:

    by comparing the sequence number on the most recently received
    ODATA or RDATA packet with the leading edge of contiguous data

    by comparing SPM_LEAD of the most recently received SPM with the
    leading edge of contiguous data

In both cases, if the receiver has not received all intervening data
packets, it MAY initiate selective NAK generation for each missing
sequence number.

In addition, a receiver may detect a single missing data packet by
receiving an NCF or multicast NAK for a data packet within the
transmit window which it has not received. In this case it MAY
initiate selective NAK generation for the said sequence number.

In all cases, receivers SHOULD temper the initiation of NAK
generation to account for simple mis-ordering introduced by the
network. A possible mechanism to achieve this is to assume loss only
after the reception of N packets with sequence numbers higher than
those of the (assumed) lost packets. A possible value for N is 2.
This method SHOULD be complemented with a timeout based mechanism
that handles the loss of the last packet before a pause in the
transmission of the data stream. The leading edge field in SPMs
SHOULD also be taken into account in the loss detection algorithm.

Generating NAKs

NAK generation follows the detection of a missing data packet and is
the cycle of:

    waiting for a random period of time (NAK_RB_IVL) while listening
    for matching NCFs or NAKs

    transmitting a NAK if a matching NCF or NAK is not heard

    waiting a period (NAK_RPT_IVL) for a matching NCF and recommencing
    NAK generation if the matching NCF is not received

    waiting a period (NAK_RDATA_IVL) for data and recommencing NAK
    generation if the matching data is not received

The entire generation process can be summarized by the following
state machine:

```
                              |
                              | detect missing tpdu
                              |    - clear data retry count
                              |    - clear NCF retry count
                              V
         matching NCF |--------------------------|
   <---------------|      BACK-OFF_STATE         |  <----------------------
   |               |   start timer(NAK_RB_IVL)   |         ^          ^
   |               |                             |         |          |
   |               |--------------------------|  |         |          |
   |    matching  |           | timer expires    |         |          |
   |       NAK    |           | - send NAK        |         |          |
   |              |           |                   |         |          |
   |              V           V                   |         |          |
   |              |--------------------------|    |         |          |
   |              |      WAIT_NCF_STATE       |    |         |          |
   |  matching NCF |   start timer(NAK_RPT_IVL) |  |         |          |
   |<-------------|                             |  |         |          |
   |              |--------------------------|  |-----------> |          |
   |              |    |           ^            | timer expires |        |
   |  NAK_NCF_RETRIES |    |       |            |  - increment NCF         |
   |     exceeded  |    |       |            |     retry count           |
   |              |    V       ----------            |                   |
   |           Cancelation    matching NAK           |                   |
   |                          - restart timer(NAK_RPT_IVL) |             |
   |                                                                     |
   |                                                                     |
   V              |--------------------------|                            |
   --------------->|      WAIT_DATA_STATE      |--------------------------->
                  |start timer(NAK_RDATA_IVL)|  timer expires
                  |                          |   - increment data
                  |--------------------------|     retry count
                  |    |           ^           |
   NAK_DATA_RETRIES |    |       |
      exceeded   |    |       |
                  |    ----------
                  |    matching NCF or NAK
                  V      - restart timer(NAK_RDATA_IVL)
              Cancellation
```

   In any state, receipt of matching RDATA or ODATA completes data
   recovery and successful exit from the state machine. State
   transition stops any running timers.

   In any state, if the trailing edge of the window moves beyond the
   sequence number, data recovery for that sequence number terminates.

During NAK_RB_IVL a NAK is said to be pending. When awaiting data or
an NCF, a NAK is said to be outstanding.

Backing off NAK transmission

Before transmitting a NAK, a receiver MUST wait some interval
NAK_RB_IVL chosen randomly over some time period NAK_BO_IVL. During
this period, receipt of a matching NAK or a matching NCF will suspend
NAK generation. NAK_RB_IVL is counted down from the time a missing
data packet is detected.

A value for NAK_BO_IVL learned from OPT_NAK_BO_IVL (see 16.4.1 below)
MUST NOT be used by a receiver (i.e., the receiver MUST NOT NAK)
unless either NAK_BO_IVL_SQN is zero, or the receiver has seen
POLL_RND == 0 for POLL_SQN =< NAK_BO_IVL_SQN within half the sequence
number space.

When a parity NAK (Appendix A, FEC) is being generated, the back-off
interval SHOULD be inversely biased with respect to the number of
parity packets requested. This way NAKs requesting larger numbers of
parity packets are likely to be sent first and thus suppress other
NAKs. A NAK for a given transmission group suppresses another NAK
for the same transmission group only if it is requesting an equal or
larger number of parity packets.

When a receiver has to transmit a sequence of NAKs, it SHOULD
transmit the NAKs in order from oldest to most recent.

Suspending NAK generation

Suspending NAK generation just means waiting for either NAK_RB_IVL,
NAK_RPT_IVL or NAK_RDATA_IVL to pass. A receiver MUST suspend NAK
generation if a duplicate of the NAK is already pending from this
receiver or the NAK is already outstanding from this or another
receiver.

NAK suppression

A receiver MUST suppress NAK generation and wait at least
NAK_RDATA_IVL before recommencing NAK generation if it hears a
matching NCF or NAK during NAK_RB_IVL. A matching NCF must match
NCF_TSI with NAK_TSI, and NCF_SQN with NAK_SQN.

Transmitting a NAK

Upon expiry of NAK_RB_IVL, a receiver MUST unicast a NAK to the
upstream PGM network element for the TSI specifying the transport
session identifier and missing sequence number. In addition, it MAY

multicast a NAK with TTL of 1 to the group, if the PGM parent is not
directly connected. It also records both the address of the source
of the corresponding ODATA and the address of the group in the NAK
header.

It MUST repeat the NAK at a rate governed by NAK_RPT_IVL up to
NAK_NCF_RETRIES times while waiting for a matching NCF. It MUST then
wait NAK_RDATA_IVL before recommencing NAK generation. If it hears a
matching NCF or NAK during NAK_RDATA_IVL, it MUST wait anew for
NAK_RDATA_IVL before recommencing NAK generation (i.e. matching NCFs
and NAKs restart NAK_RDATA_IVL).

Completion of NAK generation

NAK generation is complete only upon the receipt of the matching
RDATA (or even ODATA) packet at any time during NAK generation.

Cancellation of NAK generation

NAK generation is cancelled upon the advancing of the receive window
so as to exclude the matching sequence number of a pending or
outstanding NAK, or NAK_DATA_RETRIES / NAK_NCF_RETRIES being
exceeded. Cancellation of NAK generation indicates unrecoverable
data loss.

Receiving NCFs and multicast NAKs

A receiver MUST discard any NCFs or NAKs it hears for data packets
outside the transmit window or for data packets it has received.
Otherwise they are treated as appropriate for the current repair
state.

7. Procedures - Network Elements

7.1. Source Path State

   Upon receipt of an in-sequence SPM, a network element records the
   Source Path Address SPM_PATH with the multicast routing information
   for the TSI. If the receiving network element is on the same subnet
   as the forwarding network element, this address will be the same as
   the address of the immediately upstream network element on the
   distribution tree for the TSI. If, however, non-PGM network elements
   intervene between the forwarding and the receiving network elements,
   this address will be the address of the first PGM network element
   across the intervening network elements.

The network element then forwards the SPM on each outgoing interface
for that TSI. As it does so, it encodes the network address of the
outgoing interface in SPM_PATH in each copy of the SPM it forwards.

7.2. NAK Confirmation

Network elements MUST immediately transmit an NCF in response to any
unicast NAK they receive. The NCF MUST be multicast to the group on
the interface on which the NAK was received.

    Nota Bene: In order to avoid creating multicast routing state for
    PGM network elements across non-PGM-capable clouds, the network-
    header source address of NCFs transmitted by network elements MUST
    be set to the ODATA source's NLA, not the network element's NLA as
    might be expected.

Network elements should be able to detect a NAK storm and adopt
counter-measure to protect the network against a denial of service.
A possible countermeasure is to send the first NCF immediately in
response to a NAK and then delay the generation of further NCFs (for
identical NAKs) by a small interval, so that identical NCFs are
rate-limited, without affecting the ability to suppress NAKs.

Simultaneously, network elements MUST establish repair state for the
NAK if such state does not already exist, and add the interface on
which the NAK was received to the corresponding repair interface list
if the interface is not already listed.

7.3. Constrained NAK Forwarding

The NAK forwarding procedures for network elements are quite similar
to those for receivers, but three important differences should be
noted.

First, network elements do NOT back off before forwarding a NAK
(i.e., there is no NAK_BO_IVL) since the resulting delay of the NAK
would compound with each hop. Note that NAK arrivals will be
randomized by the receivers from which they originate, and this
factor in conjunction with NAK anticipation and elimination will
combine to forestall NAK storms on subnets with a dense network
element population.

Second, network elements do NOT retry confirmed NAKs if RDATA is not
seen; they simply discard the repair state and rely on receivers to
re-request the repair. This approach keeps the repair state in the
network elements relatively ephemeral and responsive to underlying
routing changes.

Third, note that ODATA does NOT cancel NAK forwarding in network
elements since it is switched by network elements without transport-
layer intervention.

    Nota Bene: Once confirmed by an NCF, network elements discard NAK
    packets; they are NOT retained in network elements beyond this
    forwarding operation.

NAK forwarding requires that a network element listen to NCFs for the
same transport session. NAK forwarding also requires that a network
element observe two time out intervals for any given NAK (i.e., per
NAK_TSI and NAK_SQN): NAK_RPT_IVL and NAK_RDATA_IVL.

The NAK repeat interval NAK_RPT_IVL, limits the length of time for
which a network element will repeat a NAK while waiting for a
corresponding NCF. NAK_RPT_IVL is counted down from the transmission
of a NAK. Expiry of NAK_RPT_IVL cancels NAK forwarding (due to
missing NCF).

The NAK RDATA interval NAK_RDATA_IVL, limits the length of time for
which a network element will wait for the corresponding RDATA.
NAK_RDATA_IVL is counted down from the time a matching NCF is
received. Expiry of NAK_RDATA_IVL causes the network element to
discard the corresponding repair state (due to missing RDATA).

During NAK_RPT_IVL, a NAK is said to be pending. During
NAK_RDATA_IVL, a NAK is said to be outstanding.

A Network element MUST forward NAKs only to the upstream PGM network
element for the TSI.

A network element MUST repeat a NAK at a rate of NAK_RPT_RTE for an
interval of NAK_RPT_IVL until it receives a matching NCF. A matching
NCF must match NCF_TSI with NAK_TSI, and NCF_SQN with NAK_SQN.

Upon reception of the corresponding NCF, network elements MUST wait
at least NAK_RDATA_IVL for the corresponding RDATA. Receipt of the
corresponding RDATA at any time during NAK forwarding cancels NAK
forwarding and tears down the corresponding repair state in the
network element.

7.4. NAK elimination

Two NAKs duplicate each other if they bear the same NAK_TSI and
NAK_SQN. Network elements MUST discard all duplicates of a NAK that
is pending.

Once a NAK is outstanding, network elements MUST discard all
duplicates of that NAK for NAK_ELIM_IVL. Upon expiry of
NAK_ELIM_IVL, network elements MUST suspend NAK elimination for that
TSI/SQN until the first duplicate of that NAK is seen after the
expiry of NAK_ELIM_IVL. This duplicate MUST be forwarded in the
usual manner. Once this duplicate NAK is outstanding, network
elements MUST once again discard all duplicates of that NAK for
NAK_ELIM_IVL, and so on. NAK_RDATA_IVL MUST be reset each time a NAK
for the corresponding TSI/SQN is confirmed (i.e., each time
NAK_ELIM_IVL is reset). NAK_ELIM_IVL MUST be some small fraction of
NAK_RDATA_IVL.

NAK_ELIM_IVL acts to balance implosion prevention against repair
state liveness. That is, it results in the elimination of all but at
most one NAK per NAK_ELIM_IVL thereby allowing repeated NAKs to keep
the repair state alive in the PGM network elements.

7.5. NAK Anticipation

An unsolicited NCF is one that is received by a network element when
the network element has no corresponding pending or outstanding NAK.
Network elements MUST process unsolicited NCFs differently depending
on the interface on which they are received.

If the interface on which an NCF is received is the same interface
the network element would use to reach the upstream PGM network
element, the network element simply establishes repair state for
NCF_TSI and NCF_SQN without adding the interface to the repair
interface list, and discards the NCF. If the repair state already
exists, the network element restarts the NAK_RDATA_IVL and
NAK_ELIM_IVL timers and discards the NCF.

If the interface on which an NCF is received is not the same
interface the network element would use to reach the upstream PGM
network element, the network element does not establish repair state
and just discards the NCF.

Anticipated NAKs permit the elimination of any subsequent matching
NAKs from downstream. Upon establishing anticipated repair state,
network elements MUST eliminate subsequent NAKs only for a period of
NAK_ELIM_IVL. Upon expiry of NAK_ELIM_IVL, network elements MUST
suspend NAK elimination for that TSI/SQN until the first duplicate of
that NAK is seen after the expiry of NAK_ELIM_IVL. This duplicate
MUST be forwarded in the usual manner. Once this duplicate NAK is
outstanding, network elements MUST once again discard all duplicates
of that NAK for NAK_ELIM_IVL, and so on. NAK_RDATA_IVL MUST be reset

each time a NAK for the corresponding TSI/SQN is confirmed (i.e.,
each time NAK_ELIM_IVL is reset). NAK_ELIM_IVL must be some small
fraction of NAK_RDATA_IVL.

7.6. NAK Shedding

   Network elements MAY implement local procedures for withholding NAK
   confirmations for receivers detected to be reporting excessive loss.
   The result of these procedures would ultimately be unrecoverable data
   loss in the receiver.

7.7. Addressing NAKs

   A PGM network element uses the source and group addresses (NLAs)
   contained in the transport header to find the state for the
   corresponding TSI, looks up the corresponding upstream PGM network
   element's address, uses it to re-address the (unicast) NAK, and
   unicasts it on the upstream interface for the distribution tree for
   the TSI.

7.8. Constrained RDATA Forwarding

   Network elements MUST maintain repair state for each interface on
   which a given NAK is received at least once. Network elements MUST
   then use this list of interfaces to constrain the forwarding of the
   corresponding RDATA packet only to those interfaces in the list. An
   RDATA packet corresponds to a NAK if it matches NAK_TSI and NAK_SQN.

   Network elements MUST maintain this repair state only until either
   the corresponding RDATA is received and forwarded, or NAK_RDATA_IVL
   passes after forwarding the most recent instance of a given NAK.
   Thereafter, the corresponding repair state MUST be discarded.

   Network elements SHOULD discard and not forward RDATA packets for
   which they have no repair state. Note that the consequence of this
   procedure is that, while it constrains repairs to the interested
   subset of the network, loss of repair state precipitates further NAKs
   from neglected receivers.

8. Packet Formats

   All of the packet formats described in this section are transport-
   layer headers that MUST immediately follow the network-layer header
   in the packet. Only data packet headers (ODATA and RDATA) may be
   followed in the packet by application data. For each packet type,
   the network-header source and destination addresses are specified in

addition to the format and contents of the transport layer header.
Recall from General Procedures that, for PGM over IP multicast, SPMs,
NCFs, and RDATA MUST also bear the IP Router Alert Option.

For PGM over IP, the IP protocol number is 113.

In all packets the descriptions of Data-Source Port, Data-Destination
Port, Type, Options, Checksum, Global Source ID (GSI), and Transport
Service Data Unit (TSDU) Length are:

   Data-Source Port:

      A random port number generated by the source. This port number
      MUST be unique within the source. Source Port together with
      Global Source ID forms the TSI.

   Data-Destination Port:

      A globally well-known port number assigned to the given PGM
      application.

   Type:

      The high-order two bits of the Type field encode a version
      number, 0x0 in this instance. The low-order nibble of the type
      field encodes the specific packet type. The intervening two
      bits (the low-order two bits of the high-order nibble) are
      reserved and MUST be zero.

      Within the low-order nibble of the Type field:

         values in the range 0x0 through 0x3 represent SPM-like
         packets (i.e., session-specific, sourced by a source,
         periodic),

         values in the range 0x4 through 0x7 represent DATA-like
         packets (i.e., data and repairs),

         values in the range 0x8 through 0xB represent NAK-like
         packets (i.e., hop-by-hop reliable NAK forwarding
         procedures),

         and values in the range 0xC through 0xF represent SPMR-like
         packets (i.e., session-specific, sourced by a receiver,
         asynchronous).

   Options:

      This field encodes binary indications of the presence and
      significance of any options. It also directly encodes some
      options.

      bit 0 set => One or more Option Extensions are present

      bit 1 set => One or more Options are network-significant

         Note that this bit is clear when OPT_FRAGMENT and/or
         OPT_JOIN are the only options present.

      bit 6 set => Packet is a parity packet for a transmission group
      of variable sized packets (OPT_VAR_PKTLEN). Only present when
      OPT_PARITY is also present.

      bit 7 set => Packet is a parity packet (OPT_PARITY)

      Bits are numbered here from left (0 = MSB) to right (7 = LSB).

      All the other options (option extensions) are encoded in
      extensions to the PGM header.

   Checksum:

      This field is the usual 1's complement of the 1's complement
      sum of the entire PGM packet including header.

      The checksum does not include a network-layer pseudo header for
      compatibility with network address translation. If the
      computed checksum is zero, it is transmitted as all ones. A
      value of zero in this field means the transmitter generated no
      checksum.

      Note that if any entity between a source and a receiver
      modifies the PGM header for any reason, it MUST either
      recompute the checksum or clear it. The checksum is mandatory
      on data packets (ODATA and RDATA).

   Global Source ID:

      A globally unique source identifier. This ID MUST NOT change
      throughout the duration of the transport session. A
      RECOMMENDED identifier is the low-order 48 bits of the MD5 [9]
      signature of the DNS name of the source. Global Source ID
      together with Data-Source Port forms the TSI.

TSDU Length:

>   The length in octets of the transport data unit exclusive of
>   the transport header.

>   Note that those who require the TPDU length must obtain it from
>   sum of the transport header length (TH) and the TSDU length.
>   TH length is the sum of the size of the particular PGM packet
>   header (type_specific_size) plus the length of any options that
>   might be present.

Address Family Indicators (AFIs) are as specified in [10].

8.1. Source Path Messages

SPMs are sent by a source to establish source path state in network
elements and to provide transmit window state to receivers.

The network-header source address of an SPM is the unicast NLA of the
entity that originates the SPM.

The network-header destination address of an SPM is a multicast group
NLA.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Source Port         |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Options    |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Global Source ID            ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...    Global Source ID       |          TSDU Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      SPM's Sequence Number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Trailing Edge Sequence Number               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Leading Edge Sequence Number                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          NLA AFI           |            Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Path NLA                     ...    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+
| Option Extensions when present ...                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ...  -+-+-+-+-+-+-+-+-+-+-+-+-+
```
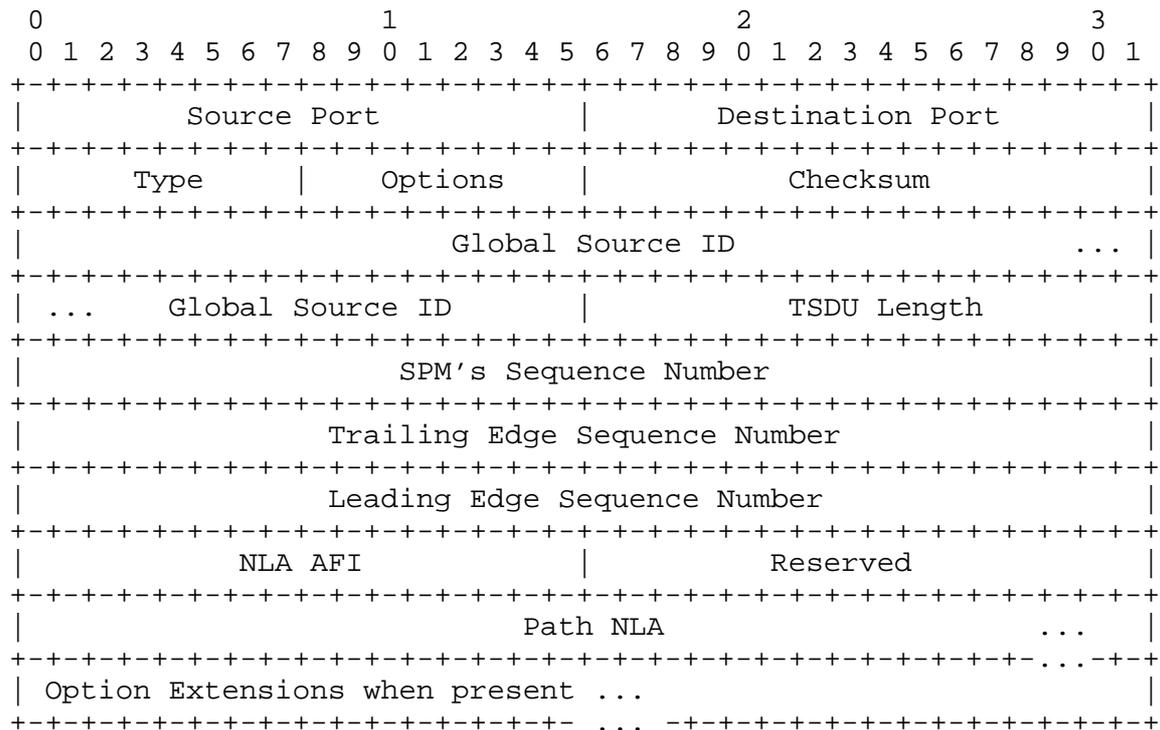
Source Port:

   SPM_SPORT

   Data-Source Port, together with SPM_GSI forms SPM_TSI

Destination Port:

   SPM_DPORT

   Data-Destination Port

Type:

   SPM_TYPE = 0x00

Global Source ID:

   SPM_GSI

   Together with SPM_SPORT forms SPM_TSI

SPM's Sequence Number

   SPM_SQN

   The sequence number assigned to the SPM by the source.

Trailing Edge Sequence Number:

   SPM_TRAIL

   The sequence number defining the current trailing edge of the
   source's transmit window (TXW_TRAIL).

Leading Edge Sequence Number:

   SPM_LEAD

   The sequence number defining the current leading edge of the
   source's transmit window (TXW_LEAD).

   If SPM_TRAIL == 0 and SPM_LEAD == 0x80000000, this indicates that
   no window information is present in the packet.

      Path NLA:

         SPM_PATH

         The NLA of the interface on the network element on which this SPM
         was forwarded. Initialized by a source to the source's NLA,
         rewritten by each PGM network element upon forwarding.

   8.2. Data Packets

      Data packets carry application data from a source or a repairer to
      receivers.

         ODATA:

            Original data packets transmitted by a source.

         RDATA:

            Repairs transmitted by a source or by a designated local
            repairer (DLR) in response to a NAK.

      The network-header source address of a data packet is the unicast NLA
      of the entity that originates the data packet.

      The network-header destination address of a data packet is a
      multicast group NLA.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Source Port         |       Destination Port        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |     Type      |    Options    |            Checksum           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                       Global Source ID            ...         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | ...     Global Source ID      |           TSDU Length         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Data Packet Sequence Number                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                 Trailing Edge Sequence Number                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Option Extensions when present ...                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ... -+-+-+-+-+-+-+-+-+-+-+-+-+
      | Data ...
      +-+-+- ...
```
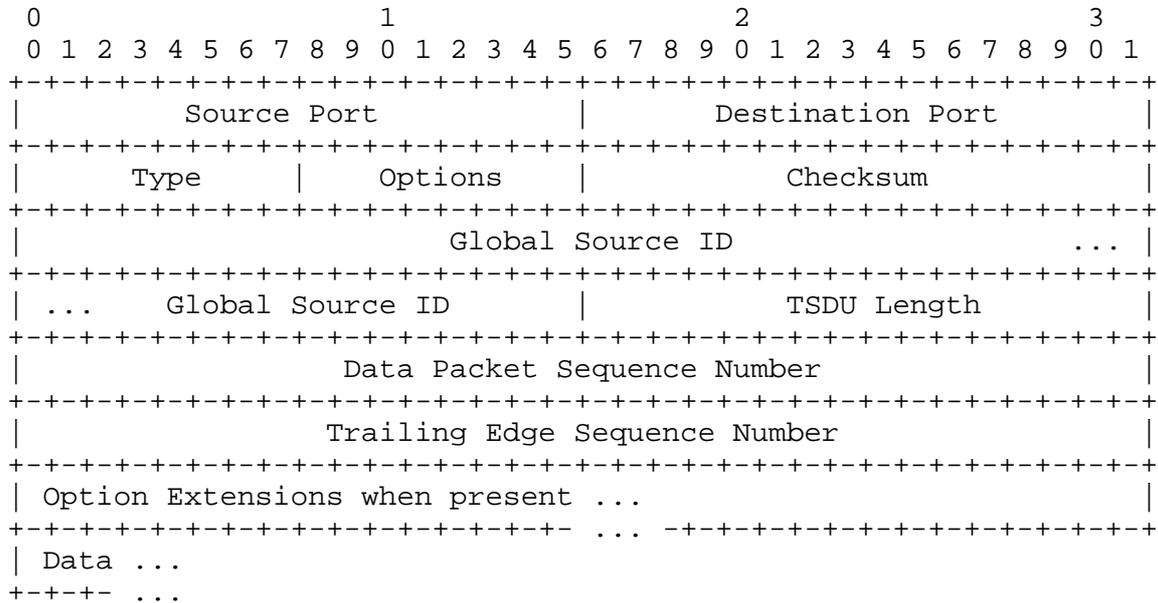
   Source Port:

      OD_SPORT, RD_SPORT

      Data-Source Port, together with Global Source ID forms:

      OD_TSI, RD_TSI

   Destination Port:

      OD_DPORT, RD_DPORT

      Data-Destination Port

   Type:

      OD_TYPE = 0x04 RD_TYPE = 0x05

   Global Source ID:

      OD_GSI, RD_GSI

      Together with Source Port forms:

      OD_TSI, RD_TSI

   Data Packet Sequence Number:

      OD_SQN, RD_SQN

      The sequence number originally assigned to the ODATA packet by the
      source.

   Trailing Edge Sequence Number:

      OD_TRAIL, RD_TRAIL

      The sequence number defining the current trailing edge of the
      source's transmit window (TXW_TRAIL). In RDATA, this MAY not be
      the same as OD_TRAIL of the ODATA packet for which it is a repair.

   Data:

      Application data.

8.3. Negative Acknowledgments and Confirmations

     NAK:

          Negative Acknowledgments are sent by receivers to request the
          repair of an ODATA packet detected to be missing from the
          expected sequence.

     N-NAK:

          Null Negative Acknowledgments are sent by DLRs to provide flow
          control feedback to the source of ODATA for which the DLR has
          provided the corresponding RDATA.

The network-header source address of a NAK is the unicast NLA of the
entity that originates the NAK. The network-header source address of
NAK is rewritten by each PGM network element with its own.

The network-header destination address of a NAK is initialized by the
originator of the NAK (a receiver) to the unicast NLA of the upstream
PGM network element known from SPMs. The network-header destination
address of a NAK is rewritten by each PGM network element with the
unicast NLA of the upstream PGM network element to which this NAK is
forwarded. On the final hop, the network-header destination address
of a NAK is rewritten by the PGM network element with the unicast NLA
of the original source or the unicast NLA of a DLR.

     NCF:

          NAK Confirmations are sent by network elements and sources to
          confirm the receipt of a NAK.

The network-header source address of an NCF is the ODATA source's
NLA, not the network element's NLA as might be expected.

The network-header destination address of an NCF is a multicast group
NLA.

Note that in NAKs and N-NAKs, unlike the other packets, the field
SPORT contains the Data-Destination port and the field DPORT contains
the Data-Source port. As a general rule, the content of SPORT/DPORT
is determined by the direction of the flow: in packets which travel
down-stream SPORT is the port number chosen in the data source
(Data-Source Port) and DPORT is the data destination port number
(Data-Destination Port). The opposite holds for packets which travel
upstream. This makes DPORT the protocol endpoint in the recipient
host, regardless of the direction of the packet.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Options    |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Global Source ID                     ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...    Global Source ID       |          TSDU Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Requested Sequence Number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         NLA AFI                |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Source NLA                           ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+
|         NLA AFI                |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Multicast Group NLA                      ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+
| Option Extensions when present ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ...
```
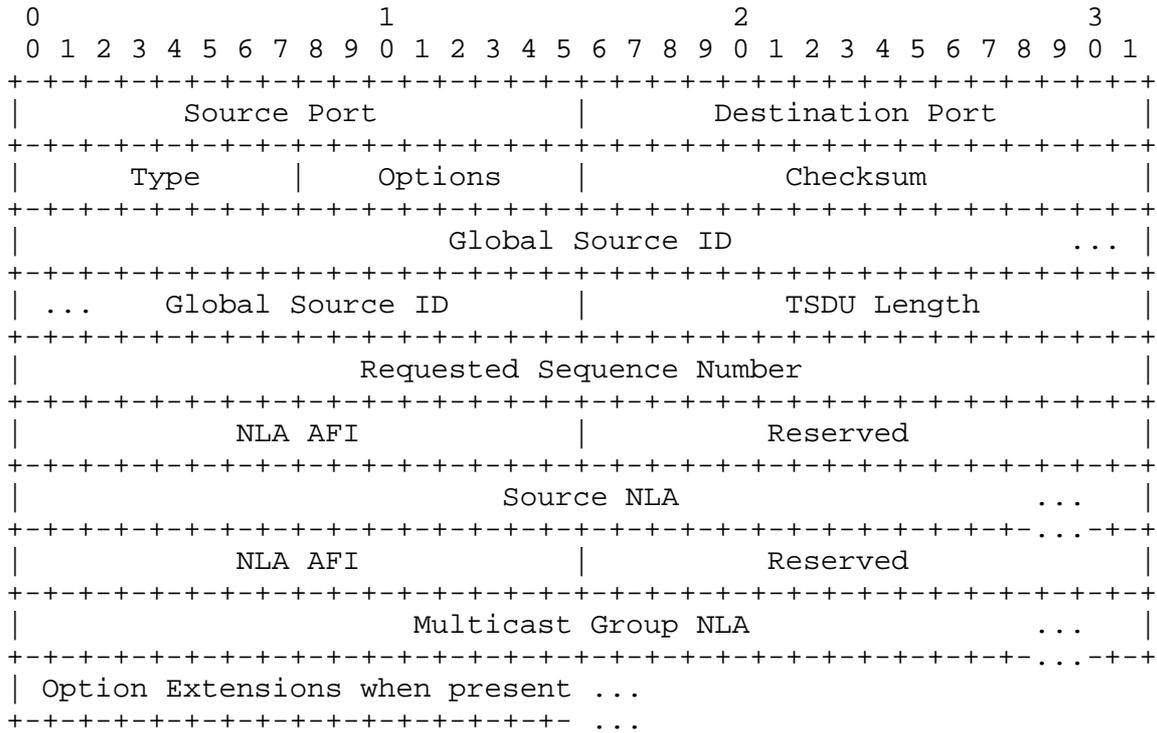
Source Port:

   NAK_SPORT, NNAK_SPORT

      Data-Destination Port

   NCF_SPORT

   Data-Source Port, together with Global Source ID forms NCF_TSI

Destination Port:

   NAK_DPORT, NNAK_DPORT

      Data-Source Port, together with Global Source ID forms:

         NAK_TSI, NNAK_TSI

   NCF_DPORT

   Data-Destination Port

   Type:

      NAK_TYPE = 0x08 NNAK_TYPE = 0x09

      NCF_TYPE = 0x0A

   Global Source ID:

      NAK_GSI, NNAK_GSI, NCF_GSI

      Together with Data-Source Port forms

         NAK_TSI, NNAK_TSI, NCF_TSI

   Requested Sequence Number:

      NAK_SQN, NNAK_SQN

      NAK_SQN is the sequence number of the ODATA packet for which a
      repair is requested.

      NNAK_SQN is the sequence number of the RDATA packet for which a
      repair has been provided by a DLR.

      NCF_SQN

      NCF_SQN is NAK_SQN from the NAK being confirmed.

   Source NLA:

      NAK_SRC, NNAK_SRC, NCF_SRC

      The unicast NLA of the original source of the missing ODATA.

   Multicast Group NLA:

      NAK_GRP, NNAK_GRP, NCF_GRP

      The multicast group NLA. NCFs MAY bear OPT_REDIRECT and/or
      OPT_NAK_LIST

9. Options

   PGM specifies several end-to-end options to address specific
   application requirements. PGM specifies options to support
   fragmentation, late joining, and redirection.

Options MAY be appended to PGM data packet headers only by their
original transmitters. While they MAY be interpreted by network
elements, options are neither added nor removed by network elements.

Options are all in the TLV style, or Type, Length, Value. The Type
field is contained in the first byte, where bit 0 is the OPT_END bit,
followed by 7 bits of type. The OPT_END bit MUST be set in the last
option in the option list, whichever that might be. The Length field
is the total length of the option in bytes, and directly follows the
Type field. Following the Length field are 5 reserved bits, the
OP_ENCODED flag, the 2 Option Extensibility bits OPX and the
OP_ENCODED_NULL flag. Last are 7 bits designated for option specific
information which may be defined on a per-option basis. If not
defined for a particular option, they MUST be set to 0.

The Option Extensibility bits dictate the desired treatment of an
option if it is unknown to the network element processing it.

   Nota Bene: Only network elements pay any attention to these bits.

   The OPX bits are defined as follows:

   00 -   Ignore the option

   01 -   Invalidate the option by changing the type to OPT_INVALID
             = 0x7F

   10 -   Discard the packet

   11 -   Unsupported, and reserved for future use

Some options present in data packet (ODATA and RDATA) are strictly
associated with the packet content (PGM payload), OPT_FRAGMENT being
an example. These options must be preserved even when the data
packet that would normally contain them is not received, but its the
payload is recovered though the use of FEC. PGM specifies a
mechanism to accomplish this that uses the F (OP_ENCODED) and U
(OP_ENCODED_NULL) bits in the option common header. OP_ENCODED and
OP_ENCODED_NULL MUST be normally set to zero except when the option
is used in FEC packets to preserve original options. See Appendix A
for details.

There is a limit of 16 options per packet.

General Option Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|Opt. Specific|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Option Value                   ...    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...+-+-+
```
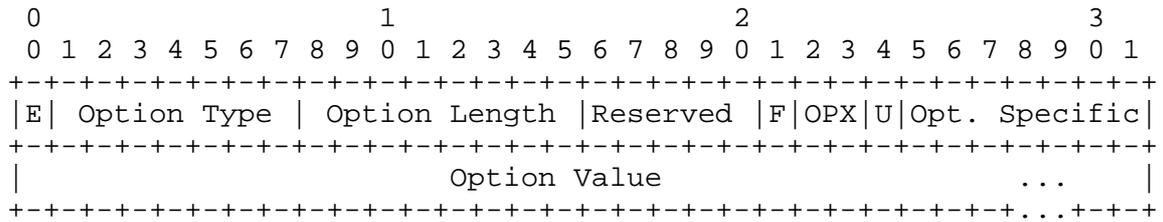
9.1. Option extension length - OPT_LENGTH

When option extensions are appended to the standard PGM header, the
extensions MUST be preceded by an option extension length field
specifying the total length of all option extensions.

In addition, the presence of the options MUST be encoded in the
Options field of the standard PGM header before the Checksum is
computed.

All network-significant options MUST be appended before any
exclusively receiver-significant options.

To provide an indication of the end of option extensions, OPT_END
(0x80) MUST be set in the Option Type field of the trailing option
extension.

9.1.1. OPT_LENGTH - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Option Type   | Option Length |  Total length of all options  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
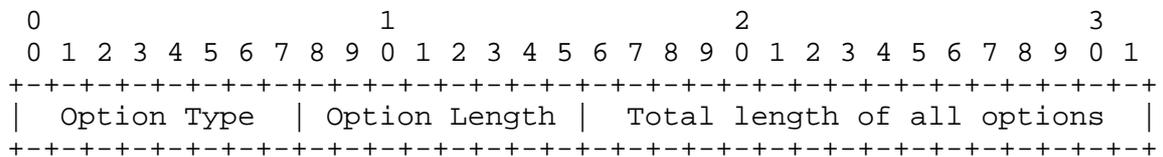
Option Type = 0x00

Option Length = 4 octets

Total length of all options

   The total length in octets of all option extensions including
   OPT_LENGTH.

OPT_LENGTH is NOT network-significant.

9.2. Fragmentation Option - OPT_FRAGMENT

   Fragmentation allows transport-layer entities at a source to break up
   application protocol data units (APDUs) into multiple PGM data
   packets (TPDUs) to conform with the MTU supported by the network
   layer. The fragmentation option MAY be applied to ODATA and RDATA
   packets only.

   Architecturally, the accumulation of TSDUs into APDUs is applied to
   TPDUs that have already been received, duplicate eliminated, and
   contiguously sequenced by the receiver. Thus APDUs MAY be
   reassembled across increments of the transmit window.

9.2.1. OPT_FRAGMENT - Packet Extension Contents

   OPT_FRAG_OFF   the offset of the fragment from the beginning of the
                   APDU

   OPT_FRAG_LEN   the total length of the original APDU

9.2.2. OPT_FRAGMENT - Procedures - Sources

   A source fragments APDUs into a contiguous series of fragments no
   larger than the MTU supported by the network layer. A source
   sequentially and uniquely assigns OD_SQNs to these fragments in the
   order in which they occur in the APDU. A source then sets
   OPT_FRAG_OFF to the value of the offset of the fragment in the
   original APDU (where the first byte of the APDU is at offset 0, and
   OPT_FRAG_OFF numbers the first byte in the fragment), and set
   OPT_FRAG_LEN to the value of the total length of the original APDU.

9.2.3. OPT_FRAGMENT - Procedures - Receivers

   Receivers detect and accumulate fragmented packets until they have
   received an entire contiguous sequence of packets comprising an APDU.
   This sequence begins with the fragment bearing OPT_FRAG_OFF of 0, and
   terminates with the fragment whose length added to its OPT_FRAG_OFF
   is OPT_FRAG_LEN.

9.2.4. OPT_FRAGMENT - Packet Extension Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |E| Option Type | Option Length |Reserved |F|OPX|U|            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                     First Sequence Number                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            Offset                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            Length                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
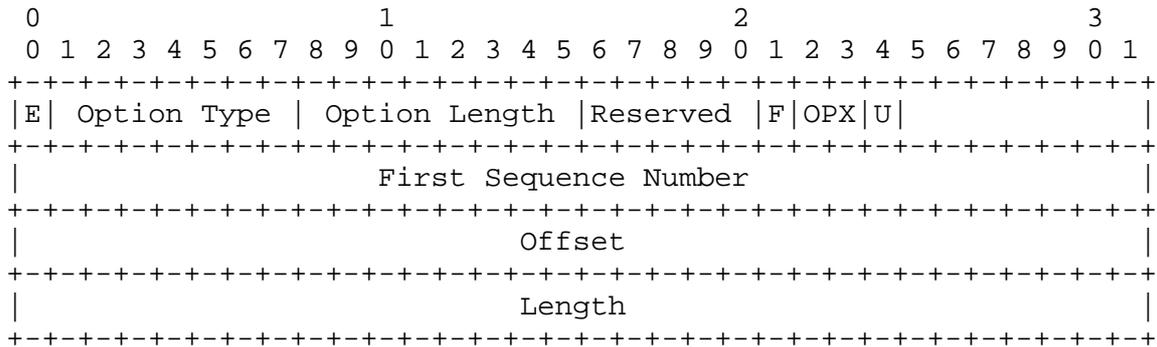
Option Type = 0x01

Option Length = 12 octets

First Sequence Number

   Sequence Number of the PGM DATA/RDATA packet containing the first
   fragment of the APDU.

Offset

   The byte offset of the fragment from the beginning of the APDU
   (OPT_FRAG_OFF).

Length

   The total length of the original APDU (OPT_FRAG_LEN).

   OPT_FRAGMENT is NOT network-significant.

9.3. NAK List Option - OPT_NAK_LIST

   The NAK List option MAY be used in conjunction with NAKs to allow
   receivers to request transmission for more than one sequence number
   with a single NAK packet. The option is limited to 62 listed NAK
   entries. The NAK list MUST be unique and duplicate free. It MUST be
   ordered, and MUST consist of either a list of selective or a list of
   parity NAKs. In general, network elements, sources and receivers
   must process a NAK list as if they had received individual NAKs for
   each sequence number in the list. The procedures for each are
   outlined in detail earlier in this document. Clarifications and
   differences are detailed here.

9.3.1. OPT_NAK_LIST - Packet Extensions Contents

   A list of sequence numbers for which retransmission is requested.

9.3.2. OPT_NAK_LIST - Procedures - Receivers

   Receivers MAY append the NAK List option to a NAK to indicate that
   they wish retransmission of a number of RDATA.

   Receivers SHOULD proceed to back off NAK transmission in a manner
   consistent with the procedures outlined for single sequence number
   NAKs. Note that the repair of each separate sequence number will be
   completed upon receipt of a separate RDATA packet.

   Reception of an NCF or multicast NAK containing the NAK List option
   suspends generation of NAKs for all sequence numbers within the NAK
   list, as well as the sequence number within the NAK header.

9.3.3. OPT_NAK_LIST - Procedures - Network Elements

   Network elements MUST immediately respond to a NAK with an identical
   NCF containing the same NAK list as the NAK itself.

   Network elements MUST forward a NAK containing a NAK List option if
   any one sequence number specified by the NAK (including that in the
   main NAK header) is not currently outstanding. That is, it MUST
   forward the NAK, if any one sequence number does not have an
   elimination timer running for it. The NAK must be forwarded intact.

   Network elements MUST eliminate a NAK containing the NAK list option
   only if all sequence numbers specified by the NAK (including that in
   the main NAK header) are outstanding. That is, they are all running
   an elimination timer.

   Upon receipt of an unsolicited NCF containing the NAK list option, a
   network element MUST anticipate data for every sequence number
   specified by the NAK as if it had received an NCF for every sequence
   number specified by the NAK.

9.3.4. OPT_NAK_LIST - Procedures - Sources

   A source MUST immediately respond to a NAK with an identical NCF
   containing the same NAK list as the NAK itself.

   It MUST then multicast RDATA (while respecting TXW_MAX_RTE) for every
   requested sequence number.

9.3.5. OPT_NAK_LIST - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Requested Sequence Number 1               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         .....                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Requested Sequence Number N               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
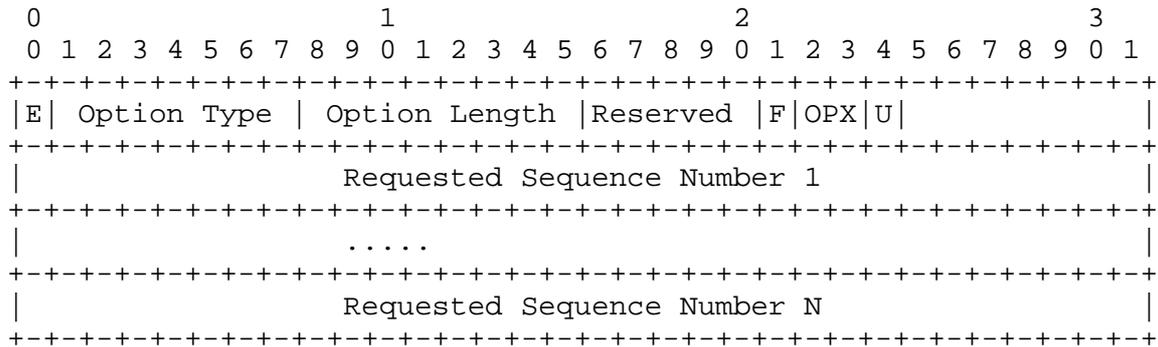
Option Type = 0x02

Option Length = 4 + (4 * number of SQNs) octets

Requested Sequence Number

   A list of up to 62 additional sequence numbers to which the NAK
   applies.

OPT_NAK_LIST is network-significant.

9.4. Late Joining Option - OPT_JOIN

Late joining allows a source to bound the amount of repair history
receivers may request when they initially join a particular transport
session.

This option indicates that receivers that join a transport session in
progress MAY request repair of all data as far back as the given
minimum sequence number from the time they join the transport
session. The default is for receivers to receive data only from the
first packet they receive and onward.

9.4.1. OPT_JOIN - Packet Extensions Contents

OPT_JOIN_MIN   the minimum sequence number for repair

9.4.2. OPT_JOIN - Procedures - Receivers

If a PGM packet (ODATA, RDATA, or SPM) bears OPT_JOIN, a receiver MAY
initialize the trailing edge of the receive window (RXW_TRAIL_INIT)
to the given Minimum Sequence Number and proceeds with normal data
reception.

9.4.3. OPT_JOIN - Packet Extension Format

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |E| Option Type | Option Length |Reserved |F|OPX|U|            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Minimum Sequence Number                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
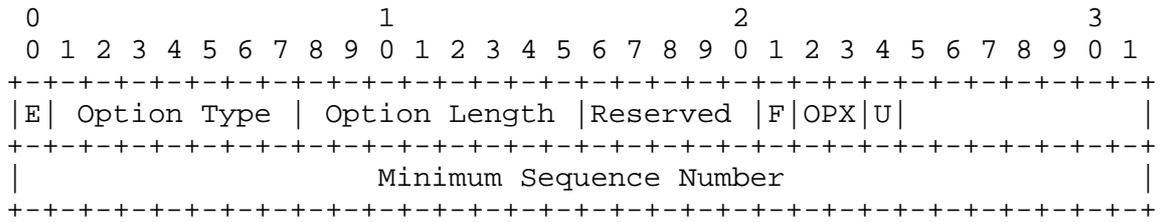
Option Type = 0x03

Option Length = 8 octets

Minimum Sequence Number

    The minimum sequence number defining the initial trailing edge of
    the receive window for a late joining receiver.

OPT_JOIN is NOT network-significant.

9.5. Redirect Option - OPT_REDIRECT

Redirection MAY be used by a designated local repairer (DLR) to
advertise its own address as an alternative to the original source,
for requesting repairs.

These procedures allow a PGM Network Element to use a DLR that is one
PGM hop from it either upstream or downstream in the multicast
distribution tree. The former are referred to as upstream DLRs. The
latter are referred to as off-tree DLRs. Off-Tree because even
though they are downstream of the point of loss, they might not lie
on the subtree affected by the loss.

A DLR MUST receive any PGM sessions for which it wishes to provide
retransmissions. A DLR SHOULD respond to NCFs or POLLs sourced by
its PGM parent with a redirecting POLR response packet containing an
OPT_REDIRECT which provides its own network layer address.
Recipients of redirecting POLRs MAY then direct NAKs for subsequent
ODATA sequence numbers to the DLR rather than to the original source.
In addition, DLRs that receive redirected NAKs for which they have
RDATA MUST send a NULL NAK to provide flow control to the original
source without also provoking a repair from that source.

9.5.1. OPT_REDIRECT - Packet Extensions Contents

   OPT_REDIR_NLA   the DLR's own unicast network-layer address to which
                   recipients of the redirecting POLR MAY direct
                   subsequent NAKs for the corresponding TSI.

9.5.2. OPT_REDIRECT - Procedures - DLRs

   A DLR MUST receive any PGM sessions for which it wishes to provide a
   source of repairs. In addition to acting as an ordinary PGM
   receiver, a DLR MAY then respond to NCFs or relevant POLLs sourced by
   parent network elements (or even by the source itself) by sending a
   POLR containing an OPT_REDIRECT providing its own network-layer
   address.

   If a DLR can provide FEC repairs it MUST denote this by setting
   OPT_PARITY in the PGM header of its POLR response.

9.5.2.1. Upstream DLRs

   If the NCF completes NAK transmission initiated by the DLR itself,
   the DLR MUST NOT send a redirecting POLR.

   When a DLR receives an NCF from its upstream PGM parent, it SHOULD
   send a redirecting POLR, multicast to the group. The DLR SHOULD
   record that it is acting as an upstream DLR for the said session.
   Note that this POLR MUST have both the data source's source address
   and the router alert option in its network header.

   An upstream DLR MUST act as an ordinary PGM source in responding to
   any NAK it receives (i.e., directed to it). That is, it SHOULD
   respond first with a normal NCF and then RDATA as usual. In
   addition, an upstream DLR that receives redirected NAKs for which it
   has RDATA MUST send a NULL NAK to provide flow control to the
   original source. If it cannot provide the RDATA it forwards the NAK
   to the upstream PGM neighbor as usual.

      Nota Bene: In order to propagate on exactly the same distribution
      tree as ODATA, RDATA and POLR packets transmitted by DLRs MUST
      bear the ODATA source's NLA as the network-header source address,
      not the DLR's NLA as might be expected.

9.5.2.2. Off-Tree DLRs

   A DLR that receives a POLL with sub-type PGM_POLL_DLR MUST respond
   with a unicast redirecting POLR if it provides the appropriate
   service. The DLR SHOULD respond using the rules outlined for polling
   in Appendix D of this text. If the DLR responds, it SHOULD record
   that it is acting as an off-tree DLR for the said session.

   An off-tree DLR acts in a special way in responding to any NAK it
   receives (i.e., directed to it). It MUST respond to a NAK directed
   to it from its parent by unicasting an NCF and RDATA to its parent.
   The parent will then forward the RDATA down the distribution tree.
   The DLR uses its own and the parent's NLA addresses in the network
   header for the source and destination respectively. The unicast NCF
   and RDATA packets SHOULD not have the router alert option. In all
   other ways the RDATA header should be "as if" the packet had come
   from the source.

   Again, an off-tree DLR that receives redirected NAKs for which it has
   RDATA MUST originate a NULL NAK to provide flow control to the
   original source. It MUST originate the NULL NAK before originating
   the RDATA. This must be done to reduce the state held in the network
   element.

   If it cannot provide the RDATA for a given NAK, an off-tree DLR
   SHOULD confirm the NAK with a unicast NCF as normal, then immediately
   send a NAK for the said data packet back to its parent.

9.5.2.3. Simultaneous Upstream and Off-Tree DLR operation

   Note that it is possible for a DLR to provide service to its parent
   and to downstream network elements simultaneously. A downstream loss
   coupled with a loss for the same data on some other part of the
   distribution tree served by its parent could cause this. In this
   case it may provide both upstream and off-tree functionality
   simultaneously.

   Note that a DLR differentiates between NAKs from an NE downstream or
   from its parent by comparing the network-header source address of the
   NAK with it's upstream PGM parent's NLA. The DLR knows the parent's
   NLA from the session's SPM messages.

9.5.3. OPT_REDIRECT - Procedures - Network Elements

9.5.3.1. Discovering DLRs

   When a PGM router receives notification of a loss via a NAK, it
   SHOULD first try to use a known DLR to recover the loss. If such a
   DLR is not known it SHOULD initiate DLR discovery. DLR discovery may
   occur in two ways. If there are upstream DLRs, the NAK transmitted
   by this router to its PGM parent will trigger their discovery, via a
   redirecting POLR. Also, a network element SHOULD initiate a search
   for off-tree DLRs using the PGM polling mechanism, and the sub-type
   PGM_POLL_DLR.

   If a DLR can provide FEC repairs it will denote this by setting
   OPT_PARITY in the PGM header of its POLR response. A network element
   SHOULD only direct parity NAKs to a DLR that can provide FEC repairs.

9.5.3.2. Redirected Repair

   When it can, a network element SHOULD use upstream DLRs.

   Upon receiving a redirecting POLR, network elements SHOULD record the
   redirecting information for the TSI, and SHOULD redirect subsequent
   NAKs for the same TSI to the network address provided in the
   redirecting POLR rather than to the PGM neighbor known via the SPMs.
   Note, however, that a redirecting POLR is NOT regarded as matching
   the NAK that provoked it, so it does not complete the transmission of
   that NAK. Only a normal matching NCF can complete the transmission
   of a NAK.

   For subsequent NAKs, if the network element has recorded redirection
   information for the corresponding TSI, it MAY change the destination
   network address of those NAKs and attempt to transmit them to the
   DLR. No NAK for a specific SQN SHOULD be sent to an off-tree DLR if
   a NAK for the SQN has been seen on the interface associated with the
   DLR. Instead the NAK SHOULD be forwarded upstream. Subsequent NAKs
   for different SQNs MAY be forwarded to the said DLR (again assuming
   no NAK for them has been seen on the interface to the DLR).

   If a corresponding NCF is not received from the DLR within
   NAK_RPT_IVL, the network element MUST discard the redirecting
   information for the TSI and re-attempt to forward the NAK towards the
   PGM upstream neighbor.

If a NAK is received from the DLR for a requested SQN, the network
element MUST discard the redirecting information for the SQN and re-
attempt to forward the NAK towards the PGM upstream neighbor. The
network element MAY still direct NAKs for different SQNs to the DLR.

RDATA and NCFs from upstream DLRs will flow down the distribution
tree. However, RDATA and NCFs from off-tree DLRs will be unicast to
the network element. The network element will terminate the NCF, but
MUST put the source's NLA and the group address into the network
header and MUST add router alert before forwarding the RDATA packet
to the distribution subtree.

NULL NAKs from an off-tree DLR for an RDATA packet requested from
that off-tree DLR MUST always be forwarded upstream. The network
element can assume that these will arrive before the matching RDATA.
Other NULL NAKs are forwarded only if matching repair state has not
already been created. Network elements MUST NOT confirm or retry
NULL NAKs and they MUST NOT add the receiving interface to the repair
state. If a NULL NAK is used to initially create repair state, this
fact must be recorded so that any subsequent non-NULL NAK will not be
eliminated, but rather will be forwarded to provoke an actual repair.
State created by a NULL NAK exists only for NAK_ELIM_IVL.

9.5.4. OPT_REDIRECT - Procedures - Receivers

These procedures are intended to be applied in instances where a
receiver's first hop router on the reverse path to the source is not
a PGM Network Element. So, receivers MUST ignore a redirecting POLR
from a DLR on the same IP subnet that the receiver resides on, since
this is likely to suffer identical loss to the receiver and so be
useless. Therefore, these procedures are entirely OPTIONAL. A
receiver MAY choose to ignore all redirecting POLRs since in cases
where its first hop router on the reverse path is PGM capable, it
would ignore them anyway. Also, note that receivers will never learn
of off-tree DLRs.

Upon receiving a redirecting POLR, receivers SHOULD record the
redirecting information for the TSI, and MAY redirect subsequent NAKs
for the same TSI to the network address provided in the redirecting
POLR rather than to the PGM neighbor for the corresponding ODATA for
which the receiver is requesting repair. Note, however, that a
redirecting POLR is NOT regarded as matching the NAK that provoked
it, so it does not complete the transmission of that NAK. Only a
normal matching NCF can complete the transmission of a NAK.

For subsequent NAKs, if the receiver has recorded redirection
information for the corresponding TSI, it MAY change the destination
network address of those NAKs and attempt to transmit them to the

DLR. If a corresponding NCF is not received within NAK_RPT_IVL, the
receiver MUST discard the redirecting information for the TSI and
re-attempt to forward the NAK to the PGM neighbor for the original
source of the missing ODATA.

9.5.5. OPT_REDIRECT - Packet Extension Format

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |E| Option Type | Option Length |Reserved |F|OPX|U|             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              NLA AFI          |             Reserved          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          DLR's NLA                    ...     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+
```
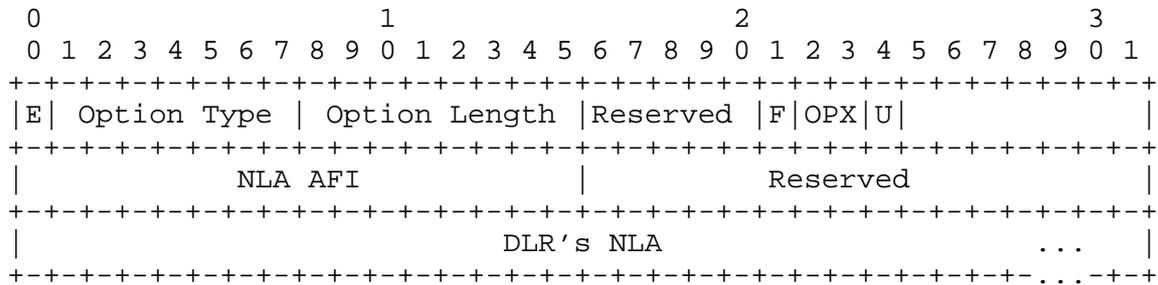
Option Type = 0x07

Option Length = 4 + NLA length

DLR's NLA

   The DLR's own unicast network address to which recipients of the
   redirecting POLR may direct subsequent NAKs.

OPT_REDIRECT is network-significant.

9.6. OPT_SYN - Synchronization Option

   The SYN option indicates the starting data packet for a session. It
   must only appear in ODATA or RDATA packets.

   The SYN option MAY be used to provide a useful abstraction to
   applications that can simplify application design by providing stream
   start notification. It MAY also be used to let a late joiner to a
   session know that it is indeed late (i.e. it would not see the SYN
   option).

9.6.1. OPT_SYN - Procedures - Receivers

   Procedures for receivers are implementation dependent. A receiver
   MAY use the SYN to provide its applications with abstractions of the
   data stream.

9.6.2. OPT_SYN - Procedures - Sources

   Sources MAY include OPT_SYN in the first data for a session. That
   is, they MAY include the option in:

      the first ODATA sent on a session by a PGM source

      any RDATA sent as a result of loss of this ODATA packet

      all FEC packets for the first transmission group; in this case it
      is interpreted as the first packet having the SYN

9.6.3. OPT_SYN - Procedures - DLRs

      In an identical manner to sources, DLRs MUST provide OPT_SYN in
      any retransmitted data that is at the start of a session.

9.6.4. OPT_SYN - Packet Extension Format

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |E| Option Type | Option Length |Reserved |F|OPX|U|            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
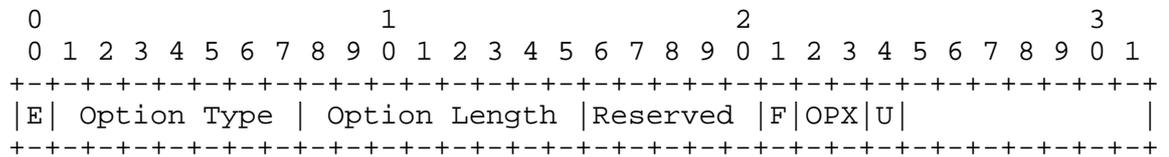
      Option Type = 0x0D

      Option Length = 4

      OPT_SYN is NOT network-significant.

9.7. OPT_FIN - Session Finish Option

      This FIN option indicates the last data packet for a session and
      an orderly close down.

      The FIN option MAY be used to provide an abstraction to
      applications that can simplify application design by providing
      stream end notification.

      This option MAY be present in the last data packet or transmission
      group for a session. The FIN PGM option MUST appear in every SPM
      sent after the last ODATA for a session. The SPM_LEAD sequence
      number in an SPM with the FIN option indicates the last known data
      successfully transmitted for the session.

9.7.1. OPT_FIN - Procedures - Receivers

     A receiver SHOULD use receipt of a FIN to let it know that it can
     tear down its data structures for the said session once a suitable
     time period has expired (TXW_SECS). It MAY still try to solicit
     retransmissions within the existing transmit window.

     Other than this, procedures for receivers are implementation
     dependent. A receiver MAY use the FIN to provide its applications
     with abstractions of the data stream and to inform its
     applications that the session is ending.

9.7.2. OPT_FIN - Procedures - Sources

     Sources MUST include OPT_FIN in every SPM sent after it has been
     determined that the application has closed gracefully. If a
     source is aware at the time of transmission that it is ending a
     session the source MAY include OPT_FIN in,

     the last ODATA

     any associated RDATAs for the last data

     FEC packets for the last transmission group; in this case it is
     interpreted as the last packet having the FIN

   When a source detects that it needs to send an OPT_FIN it SHOULD
   immediately send it. This is done either by appending it to the last
   data packet or transmission group or by immediately sending an SPM
   and resetting the SPM heartbeat timer (i.e. it does not wait for a
   timer to expire before sending the SPM). After sending an OPT_FIN,
   the session SHOULD not close and stop sending SPMs until after a time
   period equal to TXW_SECS.

9.7.3. OPT_FIN - Procedures - DLRs

   In an identical manner to sources, DLRs MUST provide OPT_FIN in any
   retransmitted data that is at the end of a session.

9.7.4. OPT_FIN - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
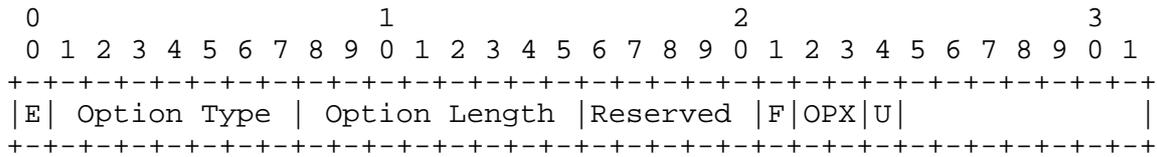
   Option Type = 0x0E

   Option Length = 4

   OPT_FIN is NOT network-significant.

9.8. OPT_RST - Session Reset Option

   The RST option MAY appear in every SPM sent after an unrecoverable
   error is identified by the source. This acts to notify the receivers
   that the session is being aborted. This option MAY appear only in
   SPMs. The SPM_LEAD sequence number in an SPM with the RST option
   indicates the last known data successfully transmitted for the
   session.

9.8.1. OPT_RST - Procedures - Receivers

   Receivers SHOULD treat the reception of OPT_RST in an SPM as an abort
   of the session.

   A receiver that receives an SPM with an OPT_RST with the N bit set
   SHOULD not send any more NAKs for the said session towards the
   source. If the N bit (see 9.8.5) is not set, the receiver MAY
   continue to try to solicit retransmit data within the current
   transmit window.

9.8.2. OPT_RST - Procedures - Sources

   Sources SHOULD include OPT_RST in every SPM sent after it has been
   determined that an unrecoverable error condition has occurred. The N
   bit of the OPT_RST SHOULD only be sent if the source has determined
   that it cannot process NAKs for the session. The cause of the
   OPT_RST is set to an implementation specific value. If the error
   code is unknown, then the value of 0x00 is used. When a source
   detects that it needs to send an OPT_RST it SHOULD immediately send
   it. This is done by immediately sending an SPM and resetting the SPM
   heartbeat timer (i.e. it does not wait for a timer to expire before
   sending the SPM). After sending an OPT_RST, the session SHOULD not
   close and stop sending SPMs until after a time period equal to
   TXW_SECS.

9.8.3. OPT_RST - Procedures - DLRs

   None.

9.8.4. OPT_RST - Packet Extension Format

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |E| Option Type | Option Length |Reserved |F|OPX|U|N|Error Code |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
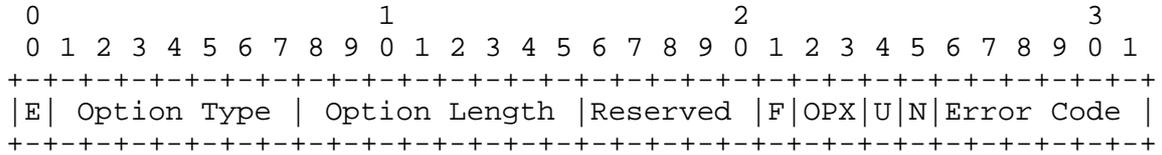
   Option Type = 0x0F

   Option Length = 4

   N bit

      The N bit is set to 1 to indicate that NAKs for previous ODATA
      will go unanswered from the source. The application will tell the
      source to turn this bit on or off.

   Error Code

      The 6 bit error code field is used to forward an error code down
      to the receivers from the source.

      The value of 0x00 indicates an unknown reset reason. Any other
      value indicates the application purposely aborted and gave a
      reason (the error code value) that may have meaning to the end
      receiver application. These values are entirely application
      dependent.

   OPT_RST is NOT network-significant.

10. Security Considerations

   In addition to the usual problems of end-to-end authentication, PGM
   is vulnerable to a number of security risks that are specific to the
   mechanisms it uses to establish source path state, to establish
   repair state, to forward NAKs, to identify DLRs, and to distribute
   repairs. These mechanisms expose PGM network elements themselves to
   security risks since network elements not only switch but also
   interpret SPMs, NAKs, NCFs, and RDATA, all of which may legitimately
   be transmitted by PGM sources, receivers, and DLRs. Short of full
   authentication of all neighboring sources, receivers, DLRs, and
   network elements, the protocol is not impervious to abuse.

So putting aside the problems of rogue PGM network elements for the
moment, there are enough potential security risks to network elements
associated with sources, receivers, and DLRs alone. These risks
include denial of service through the exhausting of both CPU
bandwidth and memory, as well as loss of (repair) data connectivity
through the muddling of repair state.

False SPMs may cause PGM network elements to mis-direct NAKs intended
for the legitimate source with the result that the requested RDATA
would not be forthcoming.

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 (or to mis-direct NAKs in the case of redirecting POLRs)
resulting eventually in loss of RDATA.

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

The development of precautions for network elements to protect
themselves against incidental or unsophisticated versions of these
attacks is work outside of this spec and includes:

    Damping of jitter in the value of either the network-header source
    address of SPMs or the path NLA in SPMs. While the network-header
    source address is expected to change seldom, the path NLA is
    expected to change occasionally as a consequence of changes in
    underlying multicast routing information.

The extension of NAK shedding procedures to control the volume, not
just the rate, of confirmed NAKs. In either case, these procedures
assist network elements in surviving NAK attacks at the expense of
maintaining service. More efficiently, network elements may use the
knowledge of TSIs and their associated transmit windows gleaned from
SPMs to control the proliferation of repair state.

A three-way handshake between network elements and DLRs that would
permit a network element to ascertain with greater confidence that an
alleged DLR is identified by the alleged network-header source
address, and is PGM conversant.

11. Appendix A - Forward Error Correction

11.1. Introduction

   The following procedures incorporate packet-level Reed Solomon
   Erasure correcting techniques as described in [11] and [12] into PGM.
   This approach to Forward Error Correction (FEC) is based upon the
   computation of h parity packets from k data packets for a total of n
   packets such that a receiver can reconstruct the k data packets out
   of any k of the n packets. The original k data packets are referred
   to as the Transmission Group, and the total n packets as the FEC
   Block.

   These procedures permit any combination of pro-active FEC or on-
   demand FEC with conventional ARQ (selective retransmission) within a
   given TSI to provide any flavor of layered or integrated FEC. The
   two approaches can be used by the same or different receivers in a
   single transport session without conflict. Once provided by a
   source, the actual use of FEC or selective retransmission for loss
   recovery in the session is entirely at the discretion of the
   receivers. Note however that receivers SHOULD NOT ask for selective
   retransmissions when FEC is available, nevertheless sources MUST
   provide selective retransmissions in response to selective NAKs from
   the leading partial transmission group (i.e. the most recent
   transmission group, which is not yet full). For any group that is
   full, the source SHOULD provide FEC on demand in response to a
   selective NAK.

   Pro-active FEC refers to the technique of computing parity packets at
   transmission time and transmitting them as a matter of course
   following the data packets. Pro-active FEC is RECOMMENDED for
   providing loss recovery over simplex or asymmetric multicast channels
   over which returning repair requests is either impossible or costly.
   It provides increased reliability at the expense of bandwidth.

   On-demand FEC refers to the technique of computing parity packets at
   repair time and transmitting them only upon demand (i.e., receiver-
   based loss detection and repair request). On-demand FEC is
   RECOMMENDED for providing loss recovery of uncorrelated loss in very
   large receiver populations in which the probability of any single
   packet being lost is substantial. It provides equivalent reliability
   to selective NAKs (ARQ) at no more and typically less expense of
   bandwidth.

   Selective NAKs are NAKs that request the retransmission of specific
   packets by sequence number corresponding to the sequence number of
   any data packets detected to be missing from the expected sequence
   (conventional ARQ). Selective NAKs can be used for recovering losses

occurring in leading partial transmission groups, i.e. in the most
recent transmission group, which is not yet full. The RECOMMENDED
way of handling partial transmission groups, however, is for the data
source to use variable-size transmission groups (see below).

Parity NAKs are NAKs that request the transmission of a specific
number of parity packets by count corresponding to the count of the
number of data packets detected to be missing from a group of k data
packets (on-demand FEC).

The objective of these procedures is to incorporate these FEC
techniques into PGM so that:

    sources MAY provide parity packets either pro-actively or on-
    demand, interchangeably within the same TSI,

    receivers MAY use either selective or parity NAKs interchangeably
    within the same TSI (however, in a session where on-demand parity
    is available receivers SHOULD only use parity NAKs).

    network elements maintain repair state based on either selective
    or parity NAKs in the same data structure, altering only search,
    RDATA constraint, and deletion algorithms in either case,

    and only OPTION additions to the basic packet formats are
    REQUIRED.

11.2. Overview

    Advertising FEC parameters in the transport session

    Sources add OPT_PARITY_PRM to SPMs to provide session-specific
    parameters such as the number of packets (TGSIZE == k) in a
    transmission group. This option lets receivers know how many packets
    there are in a transmission group, and it lets network elements sort
    repair state by transmission group number. This option includes an
    indication of whether pro-active and/or on-demand parity is available
    from the source.

    Distinguishing parity packets from data packets

    Sources send pro-active parity packets as ODATA (NEs do not forward
    RDATA unless a repair state is present) and on-demand parity packets
    as RDATA. A source MUST add OPT_PARITY to the ODATA/RDATA packet
    header of parity packets to permit network elements and receivers to
    distinguish them from data packets.

Data and parity packet numbering

Parity packets MUST be calculated over a fixed number k of data
packets known as the Transmission Group. Grouping of packets into
transmission groups effectively partitions a packet sequence number
into a high-order portion (TG_SQN) specifying the transmission group
(TG), and a low-order portion (PKT_SQN) specifying the packet number
(PKT-NUM in the range 0 through k-1) within that group. From an
implementation point of view, it's handy if k, the TG size, is a
power of 2. If so, then TG_SQN and PKT_SQN can be mapped side-by-
side into the 32 bit SQN. log2(TGSIZE) is then the size in bits of
PKT_SQN.

This mapping does not reduce the effective sequence number space
since parity packets marked with OPT_PARITY allow the sequence space
(PKT_SQN) to be completely reused in order to number the h parity
packets, as long as h is not greater than k.

In the case where h is greater than k, a source MUST add
OPT_PARITY_GRP to any parity packet numbered j greater than k-1,
specifying the number m of the group of k parity packets to which the
packet belongs, where m is just the quotient from the integer
division of j by k. Correspondingly, PKT-NUM for such parity packets
is just j modulo k. In other words, when a source needs to generate
more parity packets than there were original data packets (perhaps
because of a particularly lossy line such that a receiver lost not
only the original data but some of the parity RDATA as well), use the
OPT_PARITY_GRP option in order to number and identify the
transmission group of the extra packets that would exceed the normal
sequential number space.

Note that parity NAKs (and consequently their corresponding parity
NCFs) MUST also contain the OPT_PARITY flag in the options field of
the fixed header, and that in these packets, PKT_SQN MUST contain
PKT_CNT, the number of missing packets, rather than PKT_NUM, the SQN
of a specific missing packet. More on all this later.

Variable Transmission Group Size

The transmission group size advertised in the OPT_PARITY_PRM option
on SPMs MUST be a power of 2 and constant for the duration of the
session. However, the actual transmission group size used MAY not be
constant for the duration of the session, and MAY not be a power of
2. When a TG size different from the one advertised in
OPT_PARITY_PRM is used, the TG size advertised in OPT_PARITY_PRM MUST
be interpreted as specifying the maximum effective size of the TG.

When the actual TG size is not a power of 2 or is smaller than the
max TG size, there will be sparse utilization of the sequence number
space since some of the sequence numbers that would have been
consumed in numbering a maximum sized TG will not be assigned to
packets in the smaller TG. The start of the next transmission group
will always begin on the boundary of the maximum TG size as though
each of the sequence numbers had been utilized.

When the source decides to use a smaller group size than that
advertised in OPT_PARITY_PRM, it appends OPT_CURR_TGSIZE to the last
data packet (ODATA) in the truncated transmission group. This lets
the receiver know that it should not expect any more packets in this
transmission group, and that it may start requesting repairs for any
missing packets. If the last data packet itself went missing, the
receiver will detect the end of the group when it receives a parity
packet for the group, an SPM with SPM_LEAD equal to OD_SQN of the
last data packet, or the first packet of the next group, whichever
comes first. In addition, any parity packet from this TG will also
carry the OPT_CURR_TGSIZE option as will any SPM sent with SPM_LEAD
equal to OD_SQN of the last data packet.

Variable TSDU length

If a non constant TSDU length is used within a given transmission
group, the size of parity packets in the corresponding FEC block MUST
be equal to the size of the largest original data packet in the
block. Parity packets MUST be computed by padding the original
packets with zeros up to the size of the largest data packet. Note
that original data packets are transmitted without padding.

Receivers using a combination of original packets and FEC packets to
rebuild missing packets MUST pad the original packets in the same way
as the source does. The receiver MUST then feed the padded original
packets plus the parity packets to the FEC decoder. The decoder
produces the original packets padded with zeros up to the size of the
largest original packet in the group. In order for the receiver to
eliminate the padding on the reconstructed data packets, the original
size of the packet MUST be known, and this is accomplished as
follows:

    The source, along with the packet payloads, encodes the TSDU
    length and appends the 2-byte encoded length to the padded FEC
    packets.

    Receivers pad the original packets that they received to the
    largest original packet size and then append the TSDU length to
    the padded packets. They then pass them and the FEC packets to
    the FEC decoder.

     The decoder produces padded original packets with their original
     TSDU length appended. Receivers MUST now use this length to get
     rid of the padding.

  A source that transmits variable size packets MUST take into account
  the fact that FEC packets will have a size equal to the maximum size
  of the original packets plus the size of the length field (2 bytes).

  If a fixed packet size is used within a transmission group, the
  encoded length is not appended to the parity packets. The presence
  of the fixed header option flag OPT_VAR_PKTLEN in parity packets
  allows receivers to distinguish between transmission groups with
  variable sized packets and fixed-size ones, and behave accordingly.

  Payload-specific options

  Some options present in data packet (ODATA and RDATA) are strictly
  associated with the packet content (PGM payload), OPT_FRAGMENT being
  an example. These options must be preserved even when the data
  packet that would normally contain them is not received, but its the
  payload is recovered though the use of FEC.

  To achieve this, PGM encodes the content of these options in special
  options that are inserted in parity packets. Two flags present in
  the the option common-header are used for this process: bit F
  (OP_ENCODED) and bit U (OP_ENCODED_NULL).

  Whenever at least one of the original packets of a TG contains a
  payload-specific option of a given type, the source MUST include an
  encoded version of that option type in all the parity packets it
  transmits. The encoded option is computed by applying FEC encoding
  to the whole option with the exception of the first three bytes of
  the option common-header (E, Option Type, Option Length, OP_ENCODED
  and OPX fields). The type, length and OPX of the encoded option are
  the same as the type, length and OPX in the original options.
  OP_ENCODED is set to 1 (all original option have OP_ENCODED = 0).

  The encoding is performed using the same process that is used to
  compute the payload of the parity packet. i.e. the FEC encoder is fed
  with one copy of that option type for each original packet in the TG.
  If one (or more) original packet of the TG does not contain that
  option type, an all zeroes option is used for the encoding process.
  To be able to distinguish this "dummy" option from valid options with
  all-zeroes payload, OP_ENCODED_NULL is used. OP_ENCODED_NULL is set
  to 0 in all the original options, but the value of 1 is used in the
  encoding process if the option did not exist in the original packet.
  On the receiver side, all option with OP_ENCODED_NULL equal to 1 are
  discarded after decoding.

When a receiver recovers a missing packet using FEC repair packets,
it MUST also recover payload-specific options, if any. The presence
of these can be unequivocally detected through the presence of
encoded options in parity packets (encoded options have OP_ENCODED
set to 1). Receivers apply FEC-recovery to encoded options and
possibly original options, as they do to recover packet payloads.
The FEC decoding is applied to the whole option with the exception of
the first three bytes of the option common-header (E, Option Type,
Option Length, OP_ENCODED and OPX fields). Each decoded option is
associated with the relative payload, unless OP_ENCODED_NULL turns
out to be 1, in which case the decoded option is discarded.

The decoding MUST be performed using the 1st occurrence of a given
option type in original/parity packets. If one or more original
packets do not contain that option type, an option of the same type
with zero value must be used. This option MUST have OP_ENCODED_NULL
equal to 1.

11.3. Packet Contents

This section just provides enough short-hand to make the Procedures
intelligible. For the full details of packet contents, please refer
to Packet Formats below.

    OPT_PARITY   indicated in pro-active (ODATA) and on-demand
                        (RDATA) parity packets to distinguish them from
                        data packets. This option is directly encoded in
                        the "Option" field of the fixed PGM header

    OPT_VAR_PKTLEN   MAY be present in pro-active (ODATA) and on-demand
                        (RDATA) parity packets to indicate that the
                        corresponding transmission group is composed of
                        variable size data packets. This option is
                        directly encoded in the "Option" field of the fixed
                        PGM header

    OPT_PARITY_PRM   appended by sources to SPMs to specify session-
                        specific parameters such as the transmission group
                        size and the availability of pro-active and/or on-
                        demand parity from the source

    OPT_PARITY_GRP   the number of the group (greater than 0) of h
                        parity packets to which the parity packet belongs
                        when more than k parity packets are provided by the
                        source

    OPT_CURR_TGSIZE   appended by sources to the last data packet and any
                        parity packets in a variable sized transmission
                        group to indicate to the receiver the actual size
                        of a transmission group. May also be appended to
                        certain SPMs

11.3.1. Parity NAKs

    NAK_TG_SQN   the high-order portion of NAK_SQN specifying the
                        transmission group for which parity packets are
                        requested

    NAK_PKT_CNT   the low-order portion of NAK_SQN specifying the
                        number of missing data packets for which parity
                        packets are requested

       Nota Bene: NAK_PKT_CNT (and NCF_PKT_CNT) are 0-based counters,
       meaning that NAK_PKT_CNT = 0 means that 1 FEC RDATA is being
       requested, and in general NAK_PKT_CNT = k - 1 means that k FEC
       RDATA are being requested.

11.3.2. Parity NCFs

    NCF_TG_SQN   the high-order portion of NCF_SQN specifying the
                        transmission group for which parity packets were
                        requested

    NCF_PKT_CNT   the low-order portion of NCF_SQN specifying the
                        number of missing data packets for which parity
                        packets were requested

       Nota Bene: NCF_PKT_CNT (and NAK_PKT_CNT) are 0-based counters,
       meaning that NAK_PKT_CNT = 0 means that 1 FEC RDATA is being
       requested, and in general NAK_PKT_CNT = k - 1 means that k FEC
       RDATA are being requested.

11.3.3. On-demand Parity

    RDATA_TG_SQN   the high-order portion of RDATA_SQN specifying the
                        transmission group to which the parity packet
                        belongs

    RDATA_PKT_SQN   the low-order portion of RDATA_SQN specifying the
                        parity packet sequence number within the
                        transmission group

11.3.4. Pro-active Parity

   ODATA_TG_SQN   the high-order portion of ODATA_SQN specifying the
                         transmission group to which the parity packet
                         belongs

   ODATA_PKT_SQN   the low-order portion of ODATA_SQN specifying the
                         parity packet sequence number within the
                         transmission group

11.4. Procedures - Sources

   If a source elects to provide parity for a given transport session,
   it MUST first provide the transmission group size PARITY_PRM_TGS in
   the OPT_PARITY_PRM option of its SPMs. This becomes the maximum
   effective transmission group size in the event that the source elects
   to send smaller size transmission groups. If a source elects to
   provide proactive parity for a given transport session, it MUST set
   PARITY_PRM_PRO in the OPT_PARITY_PRM option of its SPMs. If a source
   elects to provide on-demand parity for a given transport session, it
   MUST set PARITY_PRM_OND in the OPT_PARITY_PRM option of its SPMs.

   A source MUST send any pro-active parity packets for a given
   transmission group only after it has first sent all of the
   corresponding k data packets in that group. Pro-active parity
   packets MUST be sent as ODATA with OPT_PARITY in the fixed header.

   If a source elects to provide on-demand parity, it MUST respond to a
   parity NAK for a transmission group with a parity NCF. The source
   MUST complete the transmission of the k original data packets and the
   proactive parity packets, possibly scheduled, before starting the
   transmission of on-demand parity packets. Subsequently, the source
   MUST send the number of parity packets requested by that parity NAK.
   On-demand parity packets MUST be sent as RDATA with OPT_PARITY in the
   fixed header. Previously transmitted pro-active parity packets
   cannot be reused as on-demand parity packets, these MUST be computed
   with new, previously unused, indexes.

   In either case, the source MUST provide selective retransmissions
   only in response to selective NAKs from the leading partial
   transmission group. For any group that is full, the source SHOULD
   provide FEC on demand in response to a selective retransmission
   request.

   In the absence of data to transmit, a source SHOULD prematurely
   terminate the current transmission group by including OPT_CURR_TGSIZE
   to the last data packet or to any proactive parity packets provided.

If the last data packet has already been transmitted and there is no
provision for sending proactive parity packets, an SPM with
OPT_CURR_TGSIZE SHOULD be sent.

A source consolidates requests for on-demand parity in the same
transmission group according to the following procedures. If the
number of pending (i.e., unsent) parity packets from a previous
request for on-demand parity packets is equal to or greater than
NAK_PKT_CNT in a subsequent NAK, that subsequent NAK MUST be
confirmed but MAY otherwise be ignored. If the number of pending
(i.e., unsent) parity packets from a previous request for on-demand
parity packets is less than NAK_PKT_CNT in a subsequent NAK, that
subsequent NAK MUST be confirmed but the source need only increase
the number of pending parity packets to NAK_PKT_CNT.

When a source provides parity packets relative to a transmission
group with variable sized packets, it MUST compute parity packets by
padding the smaller original packets with zeroes out to the size of
the largest of the original packets. The source MUST also append the
encoded TSDU lengths at the end of any padding or directly to the end
of the largest packet, and add the OPT_VAR_PKTLEN option as specified
in the overview description.

When a source provides variable sized transmission groups, it SHOULD
append the OPT_CURR_TGSIZE option to the last data packet in the
shortened group, and it MUST append the OPT_CURR_TGSIZE option to any
parity packets it sends within that group. In case the the last data
packet is sent before a determination has been made to shorten the
group and there is no provision for sending proactive parity packets,
an SPM with OPT_CURR_TGSIZE SHOULD be sent. The source MUST also add
OPT_CURR_TGSIZE to any SPM that it sends with SPM_LEAD equal to
OD_SQN of the last data packet.

A receiver MUST NAK for the entire number of packets missing based on
the maximum TG size, even if it already knows that the actual TG size
is smaller. The source MUST take this into account and compute the
number of packets effectively needed as the difference between
NAK_PKT_CNT and an offset computed as the difference between the max
TG size and the effective TG size.

11.5. Procedures - Receivers

If a receiver elects to make use of parity packets for loss recovery,
it MUST first learn the transmission group size PARITY_PRM_TGS from
OPT_PARITY_PRM in the SPMs for the TSI. The transmission group size
is used by a receiver to determine the sequence number boundaries
between transmission groups.

Thereafter, if PARITY_PRM_PRO is also set in the SPMs for the TSI, a receiver SHOULD use any pro-active parity packets it receives for loss recovery, and if PARITY_PRM_OND is also set in the SPMs for the TSI, it MAY solicit on-demand parity packets upon loss detection. If PARITY_PRM_OND is set, a receiver MUST NOT send selective NAKs, except in partial transmission groups if the source does not use the variable transmission-group size option. Parity packets are ODATA (pro-active) or RDATA (on-demand) packets distinguished by OPT_PARITY which lets receivers know that ODATA/RDATA_TG_SQN identifies the group of PARITY_PRM_TGS packets to which the parity may be applied for loss recovery in the corresponding transmission group, and that ODATA/RDATA_PKT_SQN is being reused to number the parity packets within that group. Receivers order parity packets and eliminate duplicates within a transmission group based on ODATA/RDATA_PKT_SQN and on OPT_PARITY_GRP if present.

To solicit on-demand parity packets, a receiver MUST send parity NAKs upon loss detection. For the purposes of soliciting on-demand parity, loss detection occurs at transmission group boundaries, i.e. upon receipt of the last data packet in a transmission group, upon receipt of any data packet in any subsequent transmission group, or upon receipt of any parity packet in the current or a subsequent transmission group.

A parity NAK is simply a NAK with OPT_PARITY and NAK_PKT_CNT set to the count of the number of packets detected to be missing from the transmission group specified by NAK_TG_SQN. Note that this constrains the receiver to request no more parity packets than there are data packets in the transmission group.

A receiver SHOULD bias the value of NAK_BO_IVL for parity NAKs inversely proportional to NAK_PKT_CNT so that NAKs for larger losses are likely to be scheduled ahead of NAKs for smaller losses in the same receiver population.

A confirming NCF for a parity NAK is a parity NCF with NCF_PKT_CNT equal to or greater than that specified by the parity NAK.

A receiver's NAK_RDATA_IVL timer is not cancelled until all requested parity packets have been received.

In the absence of data (detected from SPMs bearing SPM_LEAD equal to RXW_LEAD) on non-transmission-group boundaries, receivers MAY resort to selective NAKs for any missing packets in that partial transmission group.

When a receiver handles parity packets belonging to a transmission
group with variable sized packets, (detected from the presence of the
OPT_VAR_PKTLEN option in the parity packets), it MUST decode them as
specified in the overview description and use the decoded TSDU length
to get rid of the padding in the decoded packet.

If the source was using a variable sized transmission group via the
OPT_CURR_TGSIZE, the receiver might learn this before having
requested (and received) any retransmission. The above happens if it
sees OPT_CURR_TGSIZE in the last data packet of the TG, in any
proactive parity packet or in a SPM. If the receivers learns this
and determines that it has missed one or more packets in the
shortened transmission group, it MAY then NAK for them without
waiting for the start of the next transmission group. Otherwise it
will start NAKing at the start of the next transmission group.

In both cases, the receiver MUST NAK for the number of packets
missing assuming that the size of the transmission group is the
maximum effective transmission group. In other words, the receivers
cannot exploit the fact that it might already know that the
transmission group was smaller but MUST always NAK for the number of
packets it believes are missing, plus the number of packets required
to bring the total packets up to the maximum effective transmission
group size.

After the first parity packet has been delivered to the receiver, the
actual TG size is known to him, either because already known or
because discovered via OPT_CURR_TGSIZE contained in the parity
packet. Hence the receiver can decode the whole group as soon as the
minimum number of parity packets needed is received.

11.6. Procedures - Network Elements

Pro-active parity packets (ODATA with OPT_PARITY) are switched by
network elements without transport-layer intervention.

On-demand parity packets (RDATA with OPT_PARITY) necessitate modified
request, confirmation and repair constraint procedures for network
elements. In the context of these procedures, repair state is
maintained per NAK_TSI and NAK_TG_SQN, and in addition to recording
the interfaces on which corresponding NAKs have been received,
records the largest value of NAK_PKT_CNT seen in corresponding NAKs
on each interface. This value is referred to as the known packet
count. The largest of the known packet counts recorded for any
interface in the repair state for the transmit group or carried by an
NCF is referred to as the largest known packet count.

Upon receipt of a parity NAK, a network element responds with the
corresponding parity NCF. The corresponding parity NCF is just an
NCF formed in the usual way (i.e., a multicast copy of the NAK with
the packet type changed), but with the addition of OPT_PARITY and
with NCF_PKT_CNT set to the larger of NAK_PKT_CNT and the known
packet count for the receiving interface. The network element then
creates repair state in the usual way with the following
modifications.

If repair state for the receiving interface does not exist, the
network element MUST create it and additionally record NAK_PKT_CNT
from the parity NAK as the known packet count for the receiving
interface.

If repair state for the receiving interface already exists, the
network element MUST eliminate the NAK only if NAK_ELIM_IVL has not
expired and NAK_PKT_CNT is equal to or less than the largest known
packet count. If NAK_PKT_CNT is greater than the known packet count
for the receiving interface, the network element MUST update the
latter with the larger NAK_PKT_CNT.

Upon either adding a new interface or updating the known packet count
for an existing interface, the network element MUST determine if
NAK_PKT_CNT is greater than the largest known packet count. If so or
if NAK_ELIM_IVL has expired, the network element MUST forward the
parity NAK in the usual way with a value of NAK_PKT_CNT equal to the
largest known packet count.

Upon receipt of an on-demand parity packet, a network element MUST
locate existing repair state for the corresponding RDATA_TSI and
RDATA_TG_SQN. If no such repair state exists, the network element
MUST discard the RDATA as usual.

If corresponding repair state exists, the largest known packet count
MUST be decremented by one, then the network element MUST forward the
RDATA on all interfaces in the existing repair state, and decrement
the known packet count by one for each. Any interfaces whose known
packet count is thereby reduced to zero MUST be deleted from the
repair state. If the number of interfaces is thereby reduced to
zero, the repair state itself MUST be deleted.

Upon reception of a parity NCF, network elements MUST cancel pending
NAK retransmission only if NCF_PKT_CNT is greater or equal to the
largest known packet count. Network elements MUST use parity NCFs to
anticipate NAKs in the usual way with the addition of recording
NCF_PKT_CNT from the parity NCF as the largest known packet count
with the anticipated state so that any subsequent NAKs received with
NAK_PKT_CNT equal to or less than NCF_PKT_CNT will be eliminated, and

any with NAK_PKT_CNT greater than NCF_PKT_CNT will be forwarded.
Network elements which receive a parity NCF with NCF_PKT_CNT larger
than the largest known packet count MUST also use it to anticipate
NAKs, increasing the largest known packet count to reflect
NCF_PKT_CNT (partial anticipation).

Parity NNAKs follow the usual elimination procedures with the
exception that NNAKs are eliminated only if existing NAK state has a
NAK_PKT_CNT greater than NNAK_PKT_CNT.

Network elements must take extra precaution when the source is using
a variable sized transmission group. Network elements learn that the
source is using a TG size smaller than the maximum from
OPT_CURR_TGSIZE in parity RDATAs or in SPMs. When this happens, they
compute a TG size offset as the difference between the maximum TG
size and the actual TG size advertised by OPT_CURR_TGSIZE. Upon
reception of parity RDATA, the TG size offset is used to update the
repair state as follows:

    Any interface whose known packet count is reduced to the TG size
    offset is deleted from the repair state.

This replaces the normal rule for deleting interfaces that applies
when the TG size is equal to the maximum TG size.

11.7. Procedures - DLRs

A DLR with the ability to provide FEC repairs MUST indicate this by
setting the OPT_PARITY bit in the redirecting POLR. It MUST then
process any redirected FEC NAKs in the usual way.

11.8. Packet Formats

11.8.1. OPT_PARITY_PRM - Packet Extension Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |E| Option Type | Option Length |Reserved |F|OPX|U|         |P O|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                     Transmission Group Size                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
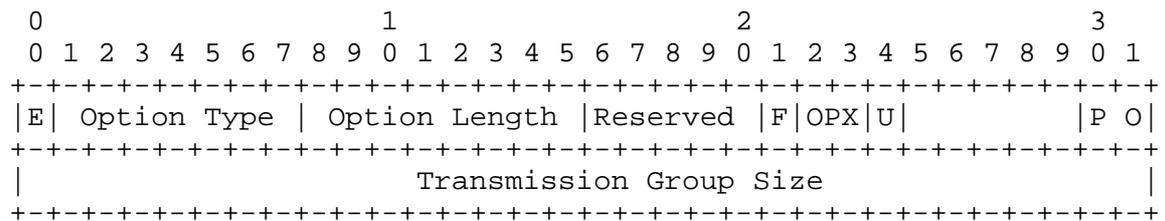
Option Type = 0x08

Option Length = 8 octets

P-bit (PARITY_PRM_PRO)

      Indicates when set that the source is providing pro-active parity
      packets.

   O-bit (PARITY_PRM_OND)

      Indicates when set that the source is providing on-demand parity
      packets.

   At least one of PARITY_PRM_PRO and PARITY_PRM_OND MUST be set.

   Transmission Group Size (PARITY_PRM_TGS)

      The number of data packets in the transmission group over which
      the parity packets are calculated. If a variable transmission
      group size is being used, then this becomes the maximum effective
      transmission group size across the session.

   OPT_PARITY_PRM MAY be appended only to SPMs.

   OPT_PARITY_PRM is network-significant.

11.8.2. OPT_PARITY_GRP - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Parity Group Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
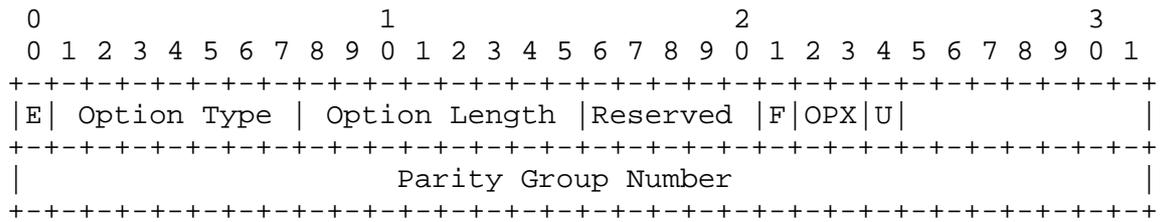
   Option Type = 0x09

   Option Length = 8 octets

   Parity Group Number (PRM_GROUP)

      The number of the group of k parity packets amongst the h parity
      packets within the transmission group to which the parity packet
      belongs, where the first k parity packets are in group zero.
      PRM_GROUP MUST NOT be zero.

   OPT_PARITY_GRP MAY be appended only to parity packets.

   OPT_PARITY_GRP is NOT network-significant.

11.8.3. OPT_CURR_TGSIZE - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Actual Transmission Group Size             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
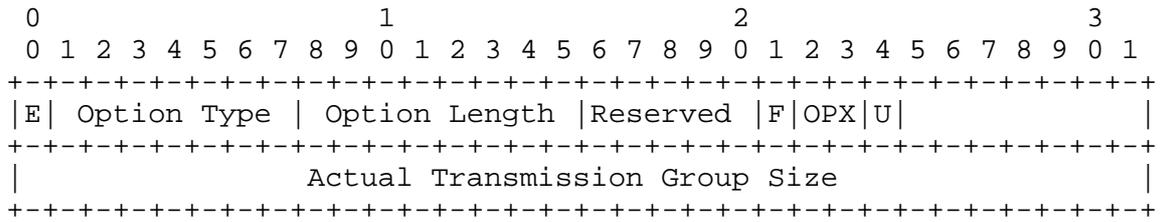
   Option Type = 0x0A

   Option Length = 8 octets

   Actual Transmission Group Size (PRM_ATGSIZE)

      The actual number of data packets in this transmission group.
      This MUST be less than or equal to the maximum transmission group
      size PARITY_PRM_TGS in OPT_PARITY_PRM.

   OPT_CURR_TGSIZE MAY be appended to data and parity packets (ODATA or
   RDATA) and to SPMs.

   OPT_CURR_TGSIZE is network-significant except when appended to ODATA.

12. Appendix B - Support for Congestion Control

12.1. Introduction

   A source MUST implement strategies for congestion avoidance, aimed at
   providing overall network stability, fairness among competing PGM
   flows, and some degree of fairness towards coexisting TCP flows [13].
   In order to do this, the source must be provided with feedback on the
   status of the network in terms of traffic load. This appendix
   specifies NE procedures that provide such feedback to the source in a
   scalable way. (An alternative TCP-friendly scheme for congestion
   control that does not require NE support can be found in [16]).

   The procedures specified in this section enable the collection and
   selective forwarding of three types of feedback to the source:

      o Worst link load as measured in network elements.

      o Worst end-to-end path load as measured in network elements.

      o Worst end-to-end path load as reported by receivers.

This specification defines in detail NE procedures, receivers
procedures and packet formats. It also defines basic procedures in
receivers for generating congestion reports. This specification does
not define the procedures used by PGM sources to adapt their
transmission rates in response of congestion reports. Those
procedures depend upon the specific congestion control scheme.

PGM defines a header option that PGM receivers may append to NAKs
(OPT_CR). OPT_CR carries congestion reports in NAKs that propagate
upstream towards the source.

During the process of hop-by-hop reverse NAK forwarding, NEs examine
OPT_CR and possibly modify its contents prior to forwarding the NAK
upstream. Forwarding CRs also has the side effect of creating
congestion report state in the NE. The presence of OPT_CR and its
contents also influences the normal NAK suppression rules. Both the
modification performed on the congestion report and the additional
suppression rules depend on the content of the congestion report and
on the congestion report state recorded in the NE as detailed below.

OPT_CR contains the following fields:

OPT_CR_NE_WL   Reports the load in the worst link as detected though
                NE internal measurements

OPT_CR_NE_WP   Reports the load in the worst end-to-end path as
                detected though NE internal measurements

OPT_CR_RX_WP   Reports the load in the worst end-to-end path as
                detected by receivers

A load report is either a packet drop rate (as measured at an NE's
interfaces) or a packet loss rate (as measured in receivers). Its
value is linearly encoded in the range 0-0xFFFF, where 0xFFFF
represents a 100% loss/drop rate. Receivers that send a NAK bearing
OPT_CR determine which of the three report fields are being reported.

OPT_CR also contains the following fields:

OPT_CR_NEL   A bit indicating that OPT_CR_NE_WL is being reported.

OPT_CR_NEP   A bit indicating that OPT_CR_NE_WP is being reported.

OPT_CR_RXP   A bit indicating that OPT_CR_RX_WP is being reported.

OPT_CR_LEAD   A SQN in the ODATA space that serves as a temporal
                   reference for the load report values. This is
                   initialized by receivers with the leading edge of the
                   transmit window as known at the moment of transmitting
                   the NAK. This value MAY be advanced in NEs that
                   modify the content of OPT_CR.

OPT_CR_RCVR   The identity of the receiver that generated the worst
                   OPT_CR_RX_WP.

   The complete format of the option is specified later.

12.2. NE-Based Worst Link Report

   To permit network elements to report worst link, receivers append
   OPT_CR to a NAK with bit OPT_CR_NEL set and OPT_CR_NE_WL set to zero.
   NEs receiving NAKs that contain OPT_CR_NE_WL process the option and
   update per-TSI state related to it as described below. The ultimate
   result of the NEs' actions ensures that when a NAK leaves a sub-tree,
   OPT_CR_NE_WL contains a congestion report that reflects the load of
   the worst link in that sub-tree. To achieve this, NEs rewrite
   OPT_CR_NE_WL with the worst value among the loads measured on the
   local (outgoing) links for the session and the congestion reports
   received from those links.

   Note that the mechanism described in this sub-section does not permit
   the monitoring of the load on (outgoing) links at non-PGM-capable
   multicast routers. For this reason, NE-Based Worst Link Reports
   SHOULD be used in pure PGM topologies only. Otherwise, this
   mechanism might fail in detecting congestion. To overcome this
   limitation PGM sources MAY use a heuristic that combines NE-Based
   Worst Link Reports and Receiver-Based Reports.

12.3. NE-Based Worst Path Report

   To permit network elements to report a worst path, receivers append
   OPT_CR to a NAK with bit OPT_CR_NEP set and OPT_CR_NE_WP set to zero.
   The processing of this field is similar to that of OPT_CR_NE_WL with
   the difference that, on the reception of a NAK, the value of
   OPT_CR_NE_WP is adjusted with the load measured on the interface on
   which the NAK was received according to the following formula:

   OPT_CR_NE_WP = if_load + OPT_CR_NE_WP * (100% - if_loss_rate)

   The worst among the adjusted OPT_CR_NE_WP is then written in the
   outgoing NAK. This results in a hop-by-hop accumulation of link loss
   rates into a path loss rate.

As with OPT_CR_NE_WL, the congestion report in OPT_CR_NE_WP may be invalid if the multicast distribution tree includes non-PGM-capable routers.

12.4. Receiver-Based Worst Report

To report a packet loss rate, receivers append OPT_CR to a NAK with bit OPT_CR_RXP set and OPT_CR_RX_WP set to the packet loss rate. NEs receiving NAKs that contain OPT_CR_RX_WP process the option and update per-TSI state related to it as described below. The ultimate result of the NEs' actions ensures that when a NAK leaves a sub-tree, OPT_CR_RX_WP contains a congestion report that reflects the load of the worst receiver in that sub-tree. To achieve this, NEs rewrite OTP_CR_RE_WP with the worst value among the congestion reports received on its outgoing links for the session. In addition to this, OPT_CR_RCVR MUST contain the NLA of the receiver that originally measured the value of OTP_CR_RE_WP being forwarded.

12.5. Procedures - Receivers

To enable the generation of any type of congestion report, receivers MUST insert OPT_CR in each NAK they generate and provide the corresponding field (OPT_CR_NE_WL, OPT_CR_NE_WP, OPT_CR_RX_WP). The specific fields to be reported will be advertised to receivers in OPT_CRQST on the session's SPMs. Receivers MUST provide only those options requested in OPT_CRQST.

Receivers MUST initialize OPT_CR_NE_WL and OPT_CR_NE_WP to 0 and they MUST initialize OPT_CR_RCVR to their NLA. At the moment of sending the NAK, they MUST also initialize OPT_CR_LEAD to the leading edge of the transmission window.

Additionally, if a receiver generates a NAK with OPT_CR with OPT_CR_RX_WP, it MUST initialize OPT_CR_RX_WP to the proper value, internally computed.

12.6. Procedures - Network Elements

Network elements start processing each OPT_CR by selecting a reference SQN in the ODATA space. The reference SQN selected is the highest SQN known to the NE. Usually this is OPT_CR_LEAD contained in the NAK received.

They use the selected SQN to age the value of load measurement as follows:

    o locally measured load values (e.g. interface loads) are
       considered up-to-date

o load values carried in OPT_CR are considered up-to-date and are
   not aged so as to be independent of variance in round-trip
   times from the network element to the receivers

o old load values recorded in the NE are exponentially aged
   according to the difference between the selected reference SQN
   and the reference SQN associated with the old load value.

The exponential aging is computed so that a recorded value gets
scaled down by a factor exp(-1/2) each time the expected inter-NAK
time elapses. Hence the aging formula must include the current loss
rate as follows:

    aged_loss_rate = loss_rate * exp(- SQN_difference * loss_rate /
    2)

Note that the quantity 1/loss_rate is the expected SQN_lag between
two NAKs, hence the formula above can also be read as:

    aged_loss_rate = loss_rate * exp(- 1/2 * SQN_difference /
    SQN_lag)

which equates to (loss_rate * exp(-1/2)) when the SQN_difference is
equal to expected SQN_lag between two NAKs.

All the subsequent computations refer to the aged load values.

Network elements process OPT_CR by handling the three possible types
of congestion reports independently.

For each congestion report in an incoming NAK, a new value is
computed to be used in the outgoing NAK:

o The new value for OPT_CR_NE_WL is the maximum of the load
   measured on the outgoing interfaces for the session, the value
   of OPT_CR_NE_WL of the incoming NAK, and the value previously
   sent upstream (recorded in the NE). All these values are as
   obtained after the aging process.

o The new value for OPT_CR_NE_WP is the maximum of the value
   previously sent upstream (after aging) and the value of
   OPT_CR_NE_WP in the incoming NAK adjusted with the load on the
   interface upon which the NAK was received (as described above).

o The new value for OPT_CR_RX_WP is the maximum of the value
   previously sent upstream (after aging) and the value of
   OPT_CR_RX_WP in the incoming NAK.

   o If OPT_CR_RX_WP was selected from the incoming NAK, the new
      value for OPT_CR_RCVR is the one in the incoming NAK.
      Otherwise it is the value previously sent upstream.

   o The new value for OPT_CR_LEAD is the reference SQN selected for
      the aging procedure.

12.6.1. Overriding Normal Suppression Rules

   Normal suppression rules hold to determine if a NAK should be
   forwarded upstream or not. However if any of the outgoing congestion
   reports has changed by more than 5% relative to the one previously
   sent upstream, this new NAK is not suppressed.

12.6.2. Link Load Measurement

   PGM routers monitor the load on all their outgoing links and record
   it in the form of per-interface loss rate statistics. "load" MUST be
   interpreted as the percentage of the packets meant to be forwarded on
   the interface that were dropped. Load statistics refer to the
   aggregate traffic on the links and not to PGM traffic only.

   This document does not specify the algorithm to be used to collect
   such statistics, but requires that such algorithm provide both
   accuracy and responsiveness in the measurement process. As far as
   accuracy is concerned, the expected measurement error SHOULD be
   upper-limited (e.g. less than than 10%). As far as responsiveness is
   concerned, the measured load SHOULD converge to the actual value in a
   limited time (e.g. converge to 90% of the actual value in less than
   200 milliseconds), when the instantaneous offered load changes.
   Whenever both requirements cannot be met at the same time, accuracy
   SHOULD be traded for responsiveness.

12.7. Packet Formats

12.7.1. OPT_CR - Packet Extension Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |E| Option Type | Option Length |Reserved |F|OPX|U|      L P R|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Congestion Report Reference SQN               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         NE Worst Link          |         NE Worst Path        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       Rcvr Worst Path          |           Reserved           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          NLA AFI               |           Reserved           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Worst Receiver's NLA            ...         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+
```

Option Type = 0x10

Option Length = 20 octets + NLA length

    L OPT_CR_NEL bit : set indicates OPT_CR_NE_WL is being reported

    P OPT_CR_NEP bit : set indicates OPT_CR_NE_WP is being reported

    R OPT_CR_RXP bit : set indicates OPT_CR_RX_WP is being reported

Congestion Report Reference SQN (OPT_CR_LEAD).

    A SQN in the ODATA space that serves as a temporal reference point
    for the load report values.

NE Worst Link (OPT_CR_NE_WL).

    Reports the load in the worst link as detected though NE internal
    measurements

NE Worst Path (OPT_CR_NE_WP).

    Reports the load in the worst end-to-end path as detected though
    NE internal measurements

Rcvr Worst Path (OPT_CR_RX_WP).

   Reports the load in the worst end-to-end path as detected by
   receivers

Worst Receiver's NLA (OPT_CR_RCVR).

   The unicast address of the receiver that generated the worst
   OPT_CR_RX_WP.

OPT_CR MAY be appended only to NAKs.

OPT-CR is network-significant.

12.7.2. OPT_CRQST - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|      L P R|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
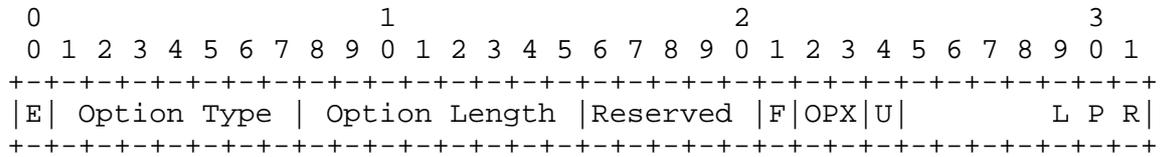
Option Type = 0x11

Option Length = 4 octets

   L OPT_CRQST_NEL bit : set indicates OPT_CR_NE_WL is being
   requested

   P OPT_CRQST_NEP bit : set indicates OPT_CR_NE_WP is being
   requested

   R OPT_CRQST_RXP bit : set indicates OPT_CR_RX_WP is being
   requested

OPT_CRQST MAY be appended only to SPMs.

OPT-CRQST is network-significant.

13. Appendix C - SPM Requests

13.1. Introduction

SPM Requests (SPMRs) MAY be used to solicit an SPM from a source in a
non-implosive way. The typical application is for late-joining
receivers to solicit SPMs directly from a source in order to be able
to NAK for missing packets without having to wait for a regularly
scheduled SPM from that source.

13.2. Overview

   Allowing for SPMR implosion protection procedures, a receiver MAY
   unicast an SPMR to a source to solicit the most current session,
   window, and path state from that source any time after the receiver
   has joined the group. A receiver may learn the TSI and source to
   which to direct the SPMR from any other PGM packet it receives in the
   group, or by any other means such as from local configuration or
   directory services. The receiver MUST use the usual SPM procedures
   to glean the unicast address to which it should direct its NAKs from
   the solicited SPM.

13.3. Packet Contents

   This section just provides enough short-hand to make the Procedures
   intelligible. For the full details of packet contents, please refer
   to Packet Formats below.

13.3.1. SPM Requests

   SPMRs are transmitted by receivers to solicit SPMs from a source.

   SPMs are unicast to a source and contain:

   SPMR_TSI   the source-assigned TSI for the session to which the
                   SPMR corresponds

13.4. Procedures - Sources

   A source MUST respond immediately to an SPMR with the corresponding
   SPM rate limited to once per IHB_MIN per TSI. The corresponding SPM
   matches SPM_TSI to SPMR_TSI and SPM_DPORT to SPMR_DPORT.

13.5. Procedures - Receivers

   To moderate the potentially implosive behavior of SPMRs at least on a
   densely populated subnet, receivers MUST use the following back-off
   and suppression procedure based on multicasting the SPMR with a TTL
   of 1 ahead of and in addition to unicasting the SPMR to the source.
   The role of the multicast SPMR is to suppress the transmission of
   identical SPMRs from the subnet.

   More specifically, before unicasting a given SPMR, receivers MUST
   choose a random delay on SPMR_BO_IVL (~250 msecs) during which they
   listen for a multicast of an identical SPMR. If a receiver does not
   see a matching multicast SPMR within its chosen random interval, it
   MUST first multicast its own SPMR to the group with a TTL of 1 before
   then unicasting its own SPMR to the source. If a receiver does see a

matching multicast SPMR within its chosen random interval, it MUST
refrain from unicasting its SPMR and wait instead for the
corresponding SPM.

In addition, receipt of the corresponding SPM within this random
interval SHOULD cancel transmission of an SPMR.

In either case, the receiver MUST wait at least SPMR_SPM_IVL before
attempting to repeat the SPMR by choosing another delay on
SPMR_BO_IVL and repeating the procedure above.

The corresponding SPMR matches SPMR_TSI to SPMR_TSI and SPMR_DPORT to
SPMR_DPORT. The corresponding SPM matches SPM_TSI to SPMR_TSI and
SPM_DPORT to SPMR_DPORT.

13.6. SPM Requests

SPMR:

   SPM Requests are sent by receivers to request the immediate
   transmission of an SPM for the given TSI from a source.

The network-header source address of an SPMR is the unicast NLA of
the entity that originates the SPMR.

The network-header destination address of an SPMR is the unicast NLA
of the source from which the corresponding SPM is requested.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Source Port         |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Options    |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Global Source ID                  ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...     Global Source ID      |           TSDU Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Option Extensions when present ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+- ...
```
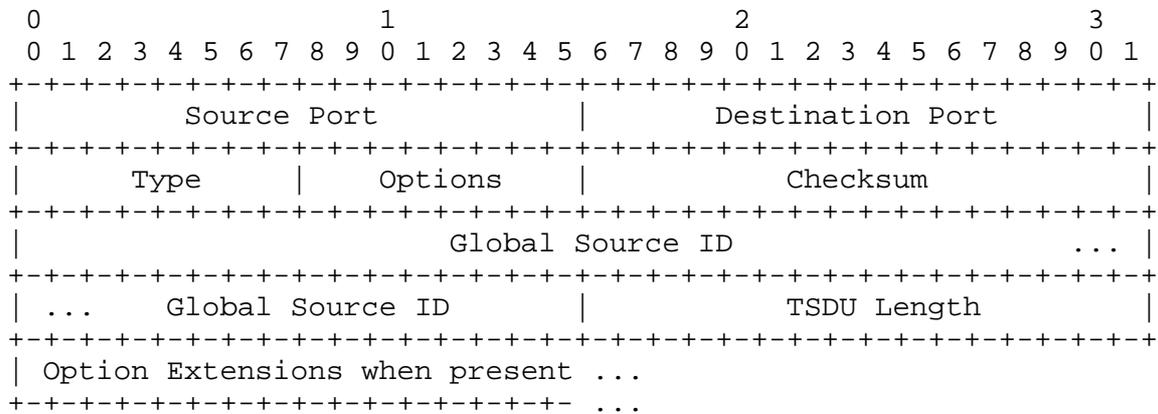
Source Port:

   SPMR_SPORT

   Data-Destination Port

    Destination Port:

        SPMR_DPORT

        Data-Source Port, together with Global Source ID forms SPMR_TSI

    Type:

        SPMR_TYPE = 0x0C

    Global Source ID:

        SPMR_GSI

        Together with Source Port forms

            SPMR_TSI

14. Appendix D - Poll Mechanism

14.1. Introduction

    These procedures provide PGM network elements and sources with the
    ability to poll their downstream PGM neighbors to solicit replies
    in an implosion-controlled way.

    Both general polls and specific polls are possible. The former
    provide a PGM (parent) node with a way to check if there are any
    PGM (children) nodes connected to it, both network elements and
    receivers, and to estimate their number. The latter may be used
    by PGM parent nodes to search for nodes with specific properties
    among its PGM children. An example of application for this is DLR
    discovery.

    Polling is implemented using two additional PGM packets:

  POLL a Poll Request that PGM parent nodes multicast to the group to
        perform the poll. Similarly to NCFs, POLL packets stop at the
        first PGM node they reach, as they are not forwarded by PGM
        network elements.

  POLR a Poll Response that PGM children nodes (either network elements
        or receivers) use to reply to a Poll Request by addressing it
        to the NLA of the interface from which the triggering POLL was
        sent.

The polling mechanism dictates that PGM children nodes that receive a POLL packet reply to it only if certain conditions are satisfied and ignore the POLL otherwise. Two types of condition are possible: a random condition that defines a probability of replying for the polled child, and a deterministic condition. Both the random condition and the deterministic condition are controlled by the polling PGM parent node by specifying the probability of replying and defining the deterministic condition(s) respectively. Random-only poll, deterministic-only poll or a combination of the two are possible.

The random condition in polls allows the prevention of implosion of replies by controlling their number. Given a probability of replying P and assuming that each receiver makes an independent decision, the number of expected replies to a poll is $P*N$ where N is the number of PGM children relative to the polling PGM parent. The polling node can control the number of expected replies by specifying P in the POLL packet.

14.2. Packet Contents

This section just provides enough short-hand to make the Procedures intelligible. For the full details of packet contents, please refer to Packet Formats below.

14.2.1. POLL (Poll Request)

POLL_SQN   a sequence number assigned sequentially by the polling
                parent in unit increments and scoped by POLL_PATH and
                the TSI of the session.

POLL_ROUND   a poll round sequence number. Multiple poll rounds
                are possible within a POLL_SQN.

POLL_S_TYPE   the sub-type of the poll request

POLL_PATH   the network-layer address (NLA) of the interface on
                the PGM network element or source on which the POLL is
                transmitted

POLL_BO_IVL   the back-off interval that MUST be used to compute the
                random back-off time to wait before sending the
                response to a poll. POLL_BO_IVL is expressed in
                microseconds.

POLL_RAND   a random string used to implement the randomness in
                replying

    POLL_MASK   a bit-mask used to determine the probability of random
                     replies

    Poll request MAY also contain options which specify deterministic
    conditions for the reply. No options are currently defined.

14.2.2. POLR (Poll Response)

    POLR_SQN   POLL_SQN of the poll request for which this is a reply

    POLR_ROUND   POLL_ROUND of the poll request for which this is a
                     reply

    Poll response MAY also contain options. No options are currently
    defined.

14.3. Procedures - General

14.3.1. General Polls

    General Polls may be used to check for and count PGM children that
    are 1 PGM hop downstream of an interface of a given node. They have
    POLL_S_TYPE equal to PGM_POLL_GENERAL. PGM children that receive a
    general poll decide whether to reply to it only based on the random
    condition present in the POLL.

    To prevent response implosion, PGM parents that initiate a general
    poll SHOULD establish the probability of replying to the poll, P, so
    that the expected number of replies is contained. The expected
    number of replies is N * P, where N is the number of children. To be
    able to compute this number, PGM parents SHOULD already have a rough
    estimate of the number of children. If they do not have a recent
    estimate of this number, they SHOULD send the first poll with a very
    low probability of replying and increase it in subsequent polls in
    order to get the desired number of replies.

    To prevent poll-response implosion caused by a sudden increase in the
    children population occurring between two consecutive polls with
    increasing probability of replying, PGM parents SHOULD use poll
    rounds. Poll rounds allow PGM parents to "freeze" the size of the
    children population when they start a poll and to maintain it
    constant across multiple polls (with the same POLL_SQN but different
    POLL_ROUND). This works by allowing PGM children to respond to a
    poll only if its POLL_ROUND is zero or if they have previously
    received a poll with the same POLL_SQN and POLL_ROUND equal to zero.

In addition to this PGM children MUST observe a random back-off in
replying to a poll. This spreads out the replies in time and allows
a PGM parent to abort the poll if too many replies are being
received. To abort an ongoing poll a PGM parent MUST initiate
another poll with different POLL_SQN. PGM children that receive a
POLL MUST cancel any pending reply for POLLs with POLL_SQN different
from the one of the last POLL received.

For a given poll with probability of replying P, a PGM parent
estimates the number of children as M / P, where M is the number of
responses received. PGM parents SHOULD keep polling periodically and
use some average of the result of recent polls as their estimate for
the number of children.

14.3.2. Specific Polls

Specific polls provide a way to search for PGM children that comply
to specific requisites. As an example specific poll could be used to
search for down-stream DLRs. A specific poll is characterized by a
POLL_S_TYPE different from PGM_POLL_GENERAL. PGM children decide
whether to reply to a specific poll or not based on the POLL_S_TYPE,
on the random condition and on options possibly present in the POLL.
The way options should be interpreted is defined by POLL_S_TYPE. The
random condition MUST be interpreted as an additional condition to be
satisfied. To disable the random condition PGM parents MUST specify
a probability of replying P equal to 1.

PGM children MUST ignore a POLL packet if they do not understand
POLL_S_TYPE. Some specific POLL_S_TYPE may also require that the
children ignore a POLL if they do not fully understand all the PGM
options present in the packet.

14.4. Procedures - PGM Parents (Sources or Network Elements)

A PGM parent (source or network element), that wants to poll the
first PGM-hop children connected to one of its outgoing interfaces
MUST send a POLL packet on that interface with:

POLL_SQN   equal to POLL_SQN of the last POLL sent incremented by
                one. If poll rounds are used, this must be equal to
                POLL_SQN of the last group of rounds incremented by
                one.

POLL_ROUND   The round of the poll. If the poll has a single
                round, this must be zero. If the poll has multiple
                rounds, this must be one plus the last POLL_ROUND for
                the same POLL_SQN, or zero if this is the first round
                within this POLL_SQN.

    POLL_S_TYPE   the type of the poll. For general poll use
                      PGM_POLL_GENERAL

    POLL_PATH   set to the NLA of the outgoing interface

    POLL_BO_IVL   set to the wanted reply back-off interval. As far as
                      the choice of this is concerned, using NAK_BO_IVL is
                      usually a conservative option, however a smaller value
                      MAY be used, if the number of expected replies can be
                      determined with a good confidence or if a
                      conservatively low probability of reply (P) is being
                      used (see POLL_MASK next). When the number of
                      expected replies is unknown, a large POLL_BO_IVL
                      SHOULD be used, so that the poll can be effectively
                      aborted if the number of replies being received is too
                      large.

    POLL_RAND   MUST be a random string re-computed each time a new
                      poll is sent on a given interface

    POLL_MASK   determines the probability of replying, P, according
                      to the relationship $P = 1 / (2 \char94 B)$, where B is the
                      number of bits set in POLL_MASK [15]. If this is a
                      deterministic poll, B MUST be 0, i.e. POLL_MASK MUST
                      be a all-zeroes bit-mask.

       Nota Bene: POLLs transmitted by network elements MUST bear the
       ODATA source's network-header source address, not the network
       element's NLA. POLLs MUST also be transmitted with the IP

       Router Alert Option [6], to be allow PGM network element to
       intercept them.

A PGM parent that has started a poll by sending a POLL packet SHOULD
wait at least POLL_BO_IVL before starting another poll. During this
interval it SHOULD collect all the valid response (the one with
POLR_SQN and POLR_ROUND matching with the outstanding poll) and
process them at the end of the collection interval.

A PGM parent SHOULD observe the rules mentioned in the description of
general procedures, to prevent implosion of response. These rules
should in general be observed both for generic polls and specific
polls. The latter however can be performed using deterministic poll
(with no implosion prevention) if the expected number of replies is
known to be small. A PGM parent that issue a generic poll with the
intent of estimating the children population size SHOULD use poll
rounds to "freeze" the children that are involved in the measure
process. This allows the sender to "open the door wider" across

subsequent rounds (by increasing the probability of response),
without fear of being flooded by late joiners. Note the use of
rounds has the drawback of introducing additional delay in the
estimate of the population size, as the estimate obtained at the end
of a round-group refers to the condition present at the time of the
first round.

A PGM parent that has started a poll SHOULD monitor the number of
replies during the collection phase. If this become too large, the
PGM parent SHOULD abort the poll by immediately starting a new poll
(different POLL_SQN) and specifying a very low probability of
replying.

When polling is being used to estimate the receiver population for
the purpose of calculating NAK_BO_IVL, OPT_NAK_BO_IVL (see 16.4.1
below) MUST be appended to SPMs, MAY be appended to NCFs and POLLs,
and in all cases MUST have NAK_BO_IVL_SQN set to POLL_SQN of the most
recent complete round of polls, and MUST bear that round's
corresponding derived value of NAK_BAK_IVL. In this way,
OPT_NAK_BO_IVL provides a current value for NAK_BO_IVL at the same
time as information is being gathered for the calculation of a future
value of NAK_BO_IVL.

14.5. Procedures - PGM Children (Receivers or Network Elements)

PGM receivers and network elements MUST compute a 32-bit random node
identifier (RAND_NODE_ID) at startup time. When a PGM child
(receiver or network element) receives a POLL it MUST use its
RAND_NODE_ID to match POLL_RAND of incoming POLLs. The match is
limited to the bits specified by POLL_MASK. If the incoming POLL
contain a POLL_MASK made of all zeroes, the match is successful
despite the content of POLL_RAND (deterministic reply). If the match
fails, then the receiver or network element MUST discard the POLL
without any further action, otherwise it MUST check the fields
POLL_ROUND, POLL_S_TYPE and any PGM option included in the POLL to
determine whether it SHOULD reply to the poll.

If POLL_ROUND is non-zero and the PGM receiver has not received a
previous poll with the same POLL_SQN and a zero POLL_ROUND, it MUST
discard the poll without further action.

If POLL_S_TYPE is equal to PGM_POLL_GENERAL, the PGM child MUST
schedule a reply to the POLL despite the presence of PGM options on
the POLL packet.

If POLL_S_TYPE is different from PGM_POLL_GENERAL, the decision on
whether a reply should be scheduled depends on the actual type and on
the options possibly present in the POLL.

If POLL_S_TYPE is unknown to the recipient of the POLL, it MUST NOT
reply and ignore the poll. Currently the only POLL_S_TYPE defined
are PGM_POLL_GENERAL and PGM_POLL_DLR.

If a PGM receiver or network element has decided to reply to a POLL,
it MUST schedule the transmission of a single POLR at a random time
in the future. The random delay is chosen in the interval [0,
POLL_BO_IVL]. POLL_BO_IVL is the one contained in the POLL received.
When this timer expires, it MUST send a POLR using POLL_PATH of the
received POLL as destination address. POLR_SQN MUST be equal to
POLL_SQN and POLR_ROUND must be equal to POLL_ROUND. The POLR MAY
contain PGM options according to the semantic of POLL_S_TYPE or the
semantic of PGM options possibly present in the POLL. If POLL_S_TYPE
is PGM_POLL_GENERAL no option is REQUIRED.

A PGM receiver or network element MUST cancel any pending
transmission of POLRs if a new POLL is received with POLL_SQN
different from POLR_SQN of the poll that scheduled POLRs.

14.6. Constant Definition

The POLL_S_TYPE values currently defined are:

     PGM_POLL_GENERAL 0

     PGM_POLL_DLR   1

14.7. Packet Formats

The packet formats described in this section are transport-layer
headers that MUST immediately follow the network-layer header in the
packet.

The descriptions of Data-Source Port, Data-Destination Port, Options,
Checksum, Global Source ID (GSI), and TSDU Length are those provided
in Section 8.

14.7.1. Poll Request

POLL are sent by PGM parents (sources or network elements) to
initiate a poll among their first PGM-hop children.

POLLs are sent to the ODATA multicast group. The network-header
source address of a POLL is the ODATA source's NLA. POLL MUST be
transmitted with the IP Router Alert Option.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Source Port         |        Destination Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Options     |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Global Source ID                    ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...     Global Source ID      |           TSDU Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    POLL's Sequence Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          POLL's Round          |         POLL's Sub-type       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            NLA AFI             |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Path NLA                         ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...-+-+
|                  POLL's  Back-off Interval                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Random String                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Matching Bit-Mask                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Option Extensions when present ...                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ... -+-+-+-+-+-+-+-+-+-+-+-+-+
```
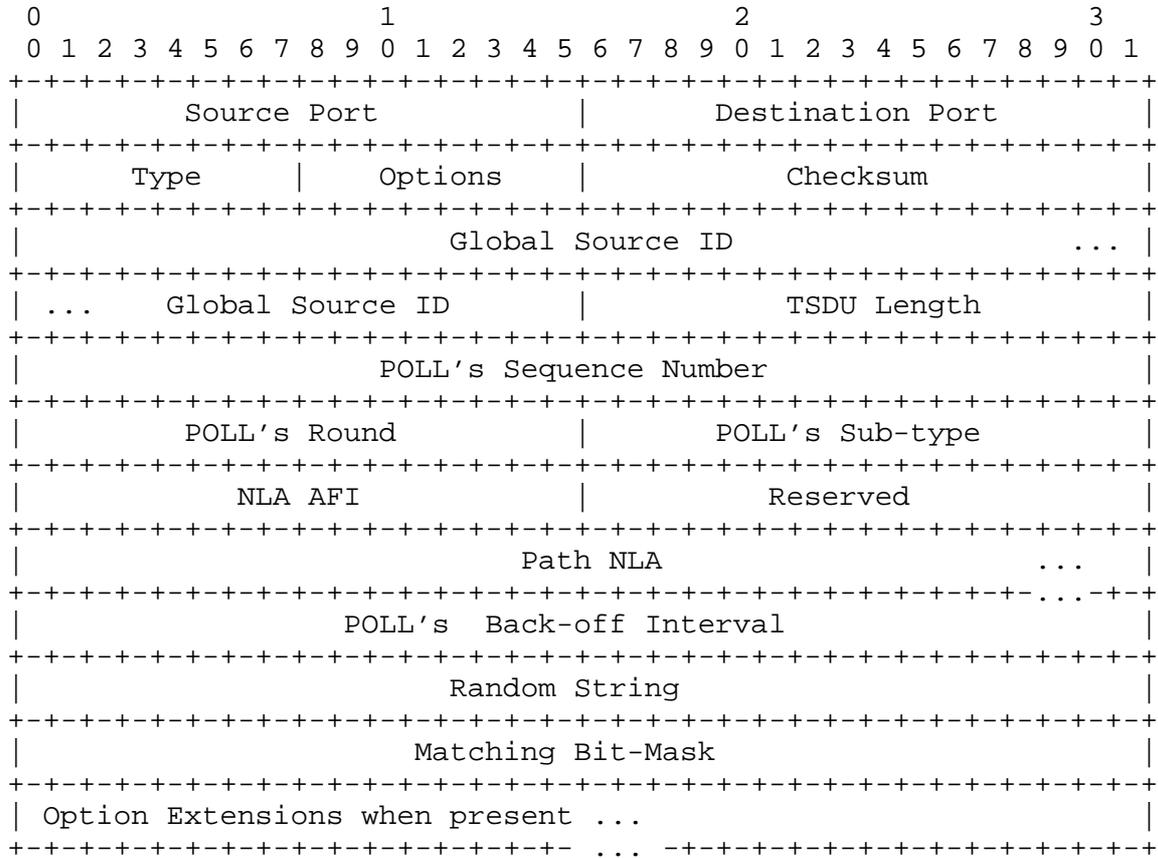
Source Port:

    POLL_SPORT

    Data-Source Port, together with POLL_GSI forms POLL_TSI

Destination Port:

    POLL_DPORT

    Data-Destination Port

Type:

    POLL_TYPE = 0x01

   Global Source ID:

      POLL_GSI

      Together with POLL_SPORT forms POLL_TSI

   POLL's Sequence Number

      POLL_SQN

      The sequence number assigned to the POLL by the originator.

   POLL's Round

      POLL_ROUND

      The round number within the poll sequence number.

   POLL's Sub-type

      POLL_S_TYPE

      The sub-type of the poll request.

   Path NLA:

      POLL_PATH

      The NLA of the interface on the source or network element on which
      this POLL was forwarded.

   POLL's Back-off Interval

      POLL_BO_IVL

      The back-off interval used to compute a random back-off for the
      reply, expressed in microseconds.

   Random String

      POLL_RAND

      A random string used to implement the random condition in
      replying.

Matching Bit-Mask

   POLL_MASK

   A bit-mask used to determine the probability of random replies.

14.7.2. Poll Response

   POLR are sent by PGM children (receivers or network elements) to
   reply to a POLL.

   The network-header source address of a POLR is the unicast NLA of the
   entity that originates the POLR. The network-header destination
   address of a POLR is initialized by the originator of the POLL to the
   unicast NLA of the upstream PGM element (source or network element)
   known from the POLL that triggered the POLR.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Source Port        |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Options    |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Global Source ID                     ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...   Global Source ID        |          TSDU Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   POLR's Sequence Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         POLR's Round          |           reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Option Extensions when present ...                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ... -+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
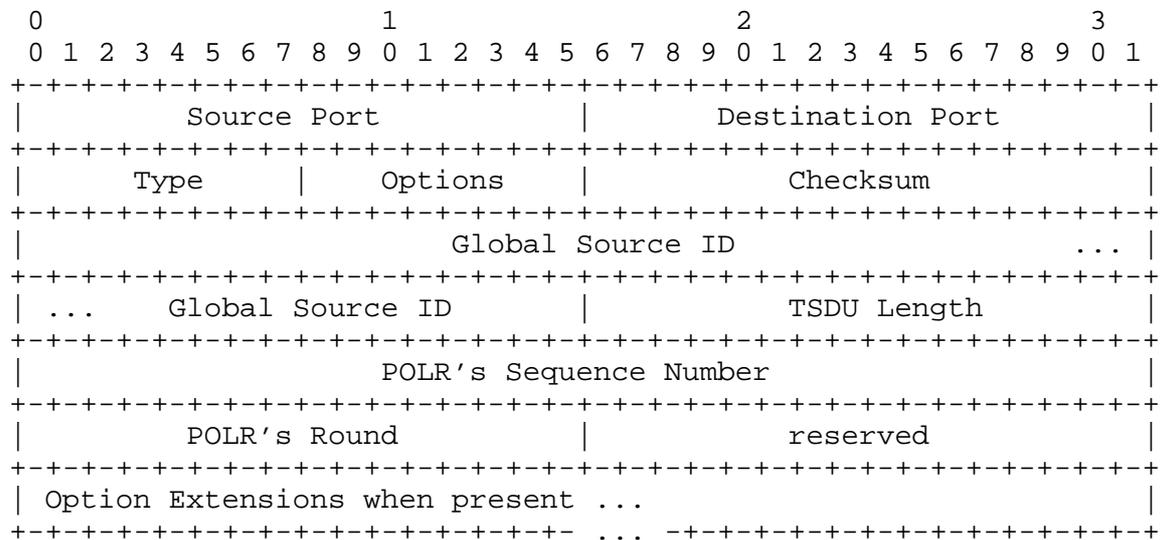
   Source Port:

      POLR_SPORT

      Data-Destination Port

   Destination Port:

      POLR_DPORT

      Data-Source Port, together with Global Source ID forms POLR_TSI

Type:

POLR_TYPE = 0x02

Global Source ID:

POLR_GSI

Together with POLR_DPORT forms POLR_TSI

POLR's Sequence Number

POLR_SQN

The sequence number (POLL_SQN) of the POLL packet for which this
is a reply.

POLR's Round

POLR_ROUND

The round number (POLL_ROUND) of the POLL packet for which this is
a reply.

15. Appendix E - Implosion Prevention

15.1. Introduction

These procedures are intended to prevent NAK implosion and to limit
its extent in case of the loss of all or part of the suppressing
multicast distribution tree. They also provide a means to adaptively
tune the NAK back-off interval, NAK_BO_IVL.

The PGM virtual topology is established and refreshed by SPMs.
Between one SPM and the next, PGM nodes may have an out-of-date view
of the PGM topology due to multicast routing changes, flapping, or a
link/router failure. If any of the above happens relative to a PGM
parent node, a potential NAK implosion problem arises because the
parent node is unable to suppress the generation of duplicate NAKs as
it cannot reach its children using NCFs. The procedures described
below introduce an alternative way of performing suppression in this
case. They also attempt to prevent implosion by adaptively tuning
NAK_BO_IVL.

15.2. Tuning NAK_BO_IVL

   Sources and network elements continuously monitor the number of
   duplicated NAKs received and use this observation to tune the NAK
   back-off interval (NAK_BO_IVL) for the first PGM-hop receivers
   connected to them. Receivers learn the current value of NAK_BO_IVL
   through OPT_NAK_BO_IVL appended to NCFs or SPMs.

15.2.1. Procedures - Sources and Network Elements

   For each TSI, sources and network elements advertise the value of
   NAK_BO_IVL that their first PGM-hop receivers should use. They
   advertise a separate value on all the outgoing interfaces for the TSI
   and keep track of the last values advertised.

   For each interface and TSI, sources and network elements count the
   number of NAKs received for a specific repair state (i.e., per
   sequence number per TSI) from the time the interface was first added
   to the repair state list until the time the repair state is
   discarded. Then they use this number to tune the current value of
   NAK_BO_IVL as follows:

      Increase the current value NAK_BO_IVL when the first duplicate NAK
      is received for a given SQN on a particular interface.

   Decrease the value of NAK_BO_IVL if no duplicate NAKs are received on
   a particular interface for the last NAK_PROBE_NUM measurements where
   each measurement corresponds to the creation of a new repair state.

   An upper and lower limit are defined for the possible value of
   NAK_BO_IVL at any time. These are NAK_BO_IVL_MAX and NAK_BO_IVL_MIN
   respectively. The initial value that should be used as a starting
   point to tune NAK_BO_IVL is NAK_BO_IVL_DEFAULT. The policies
   RECOMMENDED for increasing and decreasing NAK_BO_IVL are multiplying
   by two and dividing by two respectively.

   Sources and network elements advertise the current value of
   NAK_BO_IVL through the OPT_NAK_BO_IVL that they append to NCFs. They
   MAY also append OPT_NAK_BO_IVL to outgoing SPMs.

   In order to avoid forwarding the NAK_BO_IVL advertised by the parent,
   network elements must be able to recognize OPT_NAK_BO_IVL. Network
   elements that receive SPMs containing OPT_NAK_BO_IVL MUST either
   remove the option or over-write its content (NAK_BO_IVL) with the
   current value of NAK_BO_IVL for the outgoing interface(s), before
   forwarding the SPMs.

Sources MAY advertise the value of NAK_BO_IVL_MAX and NAK_BO_IVL_MIN
to the session by appending a OPT_NAK_BO_RNG to SPMs.

15.2.2. Procedures - Receivers

Receivers learn the value of NAK_BO_IVL to use through the option
OPT_NAK_BO_IVL, when this is present in NCFs or SPMs. A value for
NAK_BO_IVL learned from OPT_NAK_BO_IVL MUST NOT be used by a receiver
unless either NAK_BO_IVL_SQN is zero, or the receiver has seen
POLL_RND == 0 for POLL_SQN =< NAK_BO_IVL_SQN within half the sequence
number space. The initial value of NAK_BO_IVL is set to
NAK_BO_IVL_DEFAULT.

Receivers that receive an SPM containing OPT_NAK_BO_RNG MUST use its
content to set the local values of NAK_BO_IVL_MAX and NAK_BO_IVL_MIN.

15.2.3. Adjusting NAK_BO_IVL in the absence of NAKs

Monitoring the number of duplicate NAKs provides a means to track
indirectly the change in the size of first PGM-hop receiver
population and adjust NAK_BO_IVL accordingly. Note that the number
of duplicate NAKs for a given SQN is related to the number of first
PGM-hop children that scheduled (or forwarded) a NAK and not to the
absolute number of first PGM-hop children. This mechanism, however,
does not work in the absence of packet loss, hence a large number of
duplicate NAKs is possible after a period without NAKs, if many new
receivers have joined the session in the meanwhile. To address this
issue, PGM Sources and network elements SHOULD periodically poll the
number of first PGM-hop children using the "general poll" procedures
described in Appendix D. If the result of the polls shows that the
population size has increased significantly during a period without
NAKs, they SHOULD increase NAK_BO_IVL as a safety measure.

15.3. Containing Implosion in the Presence of Network Failures

15.3.1. Detecting Network Failures

In some cases PGM (parent) network elements can promptly detect the
loss of all or part of the suppressing multicast distribution tree
(due to network failures or route changes) by checking their
multicast connectivity, when they receive NAKs. In some other cases
this is not possible as the connectivity problem might occur at some
other non-PGM node downstream or might take time to reflect in the
multicast routing table. To address these latter cases, PGM uses a
simple heuristic: a failure is assumed for a TSI when the count of
duplicated NAKs received for a repair state reaches the value
DUP_NAK_MAX in one of the interfaces.

15.3.2. Containing Implosion

   When a PGM source or network element detects or assumes a failure for
   which it looses multicast connectivity to down-stream PGM agents
   (either receivers or other network elements), it sends unicast NCFs
   to them in response to NAKs. Downstream PGM network elements which
   receive unicast NCFs and have multicast connectivity to the multicast
   session send special SPMs to prevent further NAKs until a regular SPM
   sent by the source refreshes the PGM tree.

   Procedures - Sources and Network Elements

   PGM sources or network elements which detect or assume a failure that
   prevents them from reaching down-stream PGM agents through multicast
   NCFs revert to confirming NAKs through unicast NCFs for a given TSI
   on a given interface. If the PGM agent is the source itself, than it
   MUST generate an SPM for the TSI, in addition to sending the unicast
   NCF.

   Network elements MUST keep using unicast NCFs until they receive a
   regular SPM from the source.

   When a unicast NCF is sent for the reasons described above, it MUST
   contain the OPT_NBR_UNREACH option and the OPT_PATH_NLA option.
   OPT_NBR_UNREACH indicates that the sender is unable to use multicast
   to reach downstream PGM agents. OPT_PATH_NLA carries the network
   layer address of the NCF sender, namely the NLA of the interface
   leading to the unreachable subtree.

   When a PGM network element receives an NCF containing the
   OPT_NBR_UNREACH option, it MUST ignore it if OPT_PATH_NLA specifies
   an upstream neighbour different from the one currently known to be
   the upstream neighbor for the TSI. Assuming the network element
   matches the OPT_PATH_NLA of the upstream neighbour address, it MUST
   stop forwarding NAKs for the TSI until it receives a regular SPM for
   the TSI. In addition, it MUST also generate a special SPM to prevent
   downstream receivers from sending more NAKs. This special SPM MUST
   contain the OPT_NBR_UNREACH option and SHOULD have a SPM_SQN equal to
   SPM_SQN of the last regular SPM forwarded. The OPT_NBR_UNREACH
   option invalidates the windowing information in SPMs (SPM_TRAIL and
   SPM_LEAD). The PGM network element that adds the OPT_NBR_UNREACH
   option SHOULD invalidate the windowing information by setting
   SPM_TRAIL to 0 and SPM_LEAD to 0x80000000.

   PGM network elements which receive an SPM containing the
   OPT_NBR_UNREACH option and whose SPM_PATH matches the currently known
   PGM parent, MUST forward them in the normal way and MUST stop

forwarding NAKs for the TSI until they receive a regular SPM for the
TSI. If the SPM_PATH does not match the currently known PGM parent,
the SPM containing the OPT_NBR_UNREACH option MUST be ignored.

Procedures - Receivers

PGM receivers which receive either an NCF or an SPM containing the
OPT_NBR_UNREACH option MUST stop sending NAKs until a regular SPM is
received for the TSI.

On reception of a unicast NCF containing the OPT_NBR_UNREACH option
receivers MUST generate a multicast copy of the packet with TTL set
to one on the RPF interface for the data source. This will prevent
other receivers in the same subnet from generating NAKs.

Receivers MUST ignore windowing information in SPMs which contain the
OPT_NBR_UNREACH option.

Receivers MUST ignore NCFs containing the OPT_NBR_UNREACH option if
the OPT_PATH_NLA specifies a neighbour different than the one
currently know to be the PGM parent neighbour. Similarly receivers
MUST ignore SPMs containing the OPT_NBR_UNREACH option if SPM_PATH
does not match the current PGM parent.

15.4. Packet Formats

15.4.1. OPT_NAK_BO_IVL - Packet Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E| Option Type | Option Length |Reserved |F|OPX|U|            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    NAK Back-Off Interval                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  NAK Back-Off Interval SQN                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
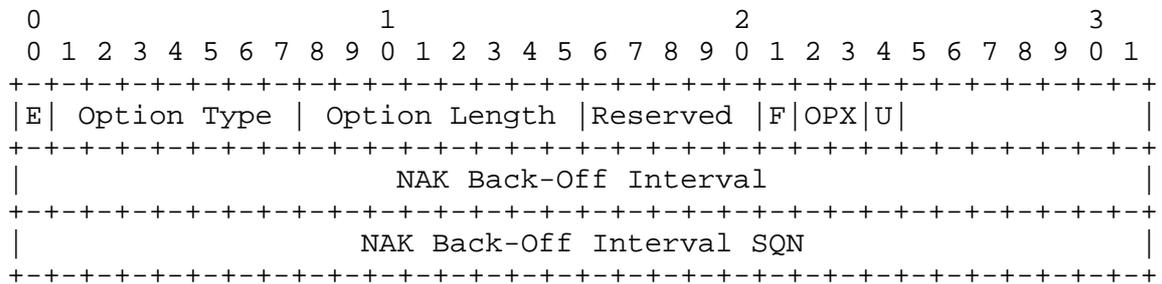
Option Type = 0x04

NAK Back-Off Interval

    The value of NAK-generation Back-Off Interval in microseconds.

   NAK Back-Off Interval Sequence Number

      The POLL_SQN to which this value of NAK_BO_IVL corresponds. Zero
      is reserved and means NAK_BO_IVL is NOT being determined through
      polling (see Appendix D) and may be used immediately. Otherwise,
      NAK_BO_IVL MUST NOT be used unless the receiver has also seen
      POLL_ROUND = 0 for POLL_SQN =< NAK_BO_IVL_SQN within half the
      sequence number space.

   OPT_NAK_BO_IVL MAY be appended to NCFs, SPMs, or POLLs.

   OPT_NAK_BO_IVL is network-significant.

15.4.2. OPT_NAK_BO_RNG - Packet Extension Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |E| Option Type | Option Length |Reserved |F|OPX|U|           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Maximum  NAK Back-Off Interval              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Minimum  NAK Back-Off Interval              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
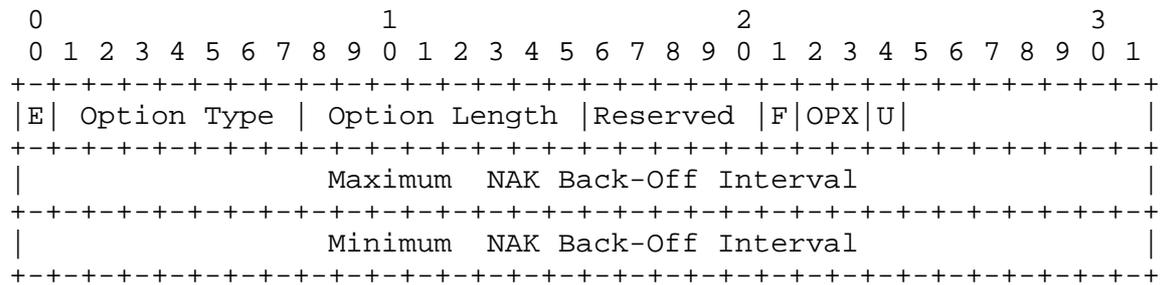
   Option Type = 0x05

   Maximum NAK Back-Off Interval

      The maximum value of NAK-generation Back-Off Interval in
      microseconds.

   Minimum NAK Back-Off Interval

      The minimum value of NAK-generation Back-Off Interval in
      microseconds.

   OPT_NAK_BO_RNG MAY be appended to SPMs.

   OPT_NAK_BO_RNG is network-significant.

15.4.3. OPT_NBR_UNREACH - Packet Extension Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |E| Option Type | Option Length |Reserved |F|OPX|U|           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
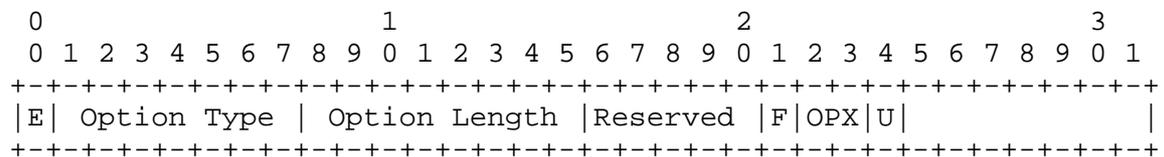
    Option Type = 0x0B

    When present in SPMs, it invalidates the windowing information.

  OPT_NBR_UNREACH MAY be appended to SPMs and NCFs.

  OPT_NBR_UNREACH is network-significant.

15.4.4. OPT_PATH_NLA - Packet Extension Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |E| Option Type | Option Length |Reserved |F|OPX|U|            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            Path NLA                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
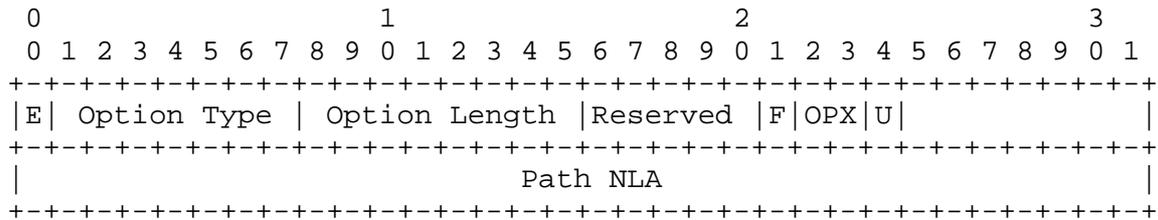
  Option Type = 0x0C

  Path NLA

    The NLA of the interface on the originating PGM network element
    that it uses to send multicast SPMs to the recipient of the packet
    containing this option.

  OPT_PATH_NLA MAY be appended to NCFs.

  OPT_PATH_NLA is network-significant.

16. Appendix F - Transmit Window Example

    Nota Bene: The concept of and all references to the increment
    window (TXW_INC) and the window increment (TXW_ADV_SECS)
    throughout this document are for illustrative purposes only. They
    provide the shorthand with which to describe the concept of
    advancing the transmit window without also implying any particular
    implementation or policy of advancement.

  The size of the transmit window in seconds is simply TXW_SECS. The
  size of the transmit window in bytes (TXW_BYTES) is (TXW_MAX_RTE *
  TXW_SECS). The size of the transmit window in sequence numbers
  (TXW_SQNS) is (TXW_BYTES / bytes-per-packet).

  The fraction of the transmit window size (in seconds of data) by
  which the transmit window is advanced (TXW_ADV_SECS) is called the
  window increment. The trailing (oldest) such fraction of the
  transmit window itself is called the increment window.

In terms of sequence numbers, the increment window is the range of
sequence numbers that will be the first to be expired from the
transmit window. The trailing (or left) edge of the increment window
is just TXW_TRAIL, the trailing (or left) edge of the transmit
window. The leading (or right) edge of the increment window
(TXW_INC) is defined as one less than the sequence number of the
first data packet transmitted by the source TXW_ADV_SECS after
transmitting TXW_TRAIL.

A data packet is described as being "in" the transmit or increment
window, respectively, if its sequence number is in the range defined
by the transmit or increment window, respectively.

The transmit window is advanced across the increment window by the
source when it increments TXW_TRAIL to TXW_INC. When the transmit
window is advanced across the increment window, the increment window
is emptied (i.e., TXW_TRAIL is momentarily equal to TXW_INC), begins
to refill immediately as transmission proceeds, is full again
TXW_ADV_SECS later (i.e., TXW_TRAIL is separated from TXW_INC by
TXW_ADV_SECS of data), at which point the transmit window is advanced
again, and so on.

16.1. Advancing across the Increment Window

In anticipation of advancing the transmit window, the source starts a
timer TXW_ADV_IVL_TMR which runs for time period TXW_ADV_IVL.
TXW_ADV_IVL has a value in the range (0, TXW_ADV_SECS). The value
MAY be configurable or MAY be determined statically by the strategy
used for advancing the transmit window.

When TXW_ADV_IVL_TMR is running, a source MAY reset TXW_ADV_IVL_TMR
if NAKs are received for packets in the increment window. In
addition, a source MAY transmit RDATA in the increment window with
priority over other data within the transmit window.

When TXW_ADV_IVL_TMR expires, a source SHOULD advance the trailing
edge of the transmit window from TXW_TRAIL to TXW_INC.

Once the transmit window is advanced across the increment window,
SPM_TRAIL, OD_TRAIL and RD_TRAIL are set to the new value of
TXW_TRAIL in all subsequent transmitted packets, until the next
window advancement.

PGM does not constrain the strategies that a source may use for
advancing the transmit window. The source MAY implement any scheme
or number of schemes. Three suggested strategies are outlined here.

Consider the following example:

    Assuming a constant transmit rate of 128kbps and a constant data
    packet size of 1500 bytes, if a source maintains the past 30
    seconds of data for repair and increments its transmit window in 5
    second increments, then

        TXW_MAX_RTE = 16kBps
        TXW_ADV_SECS = 5 seconds,
        TXW_SECS = 35 seconds,
        TXW_BYTES = 560kB,
        TXW_SQNS = 383 (rounded up),

    and the size of the increment window in sequence numbers
    (TXW_MAX_RTE * TXW_ADV_SECS / 1500) = 54 (rounded down).

Continuing this example, the following is a diagram of the transmit
window and the increment window therein in terms of sequence numbers.

    TXW_TRAIL   TXW_LEAD
        | |
        | |
    |--|--------------- Transmit Window -------------|----|
    v | | v
       v   v
n-1 | n | n+1 | ... | n+53 | n+54 | ... | n+381 | n+382 | n+383
                              ^
    ^ | ^
    |--- Increment Window|---|
                              |
                              |
                         TXW_INC
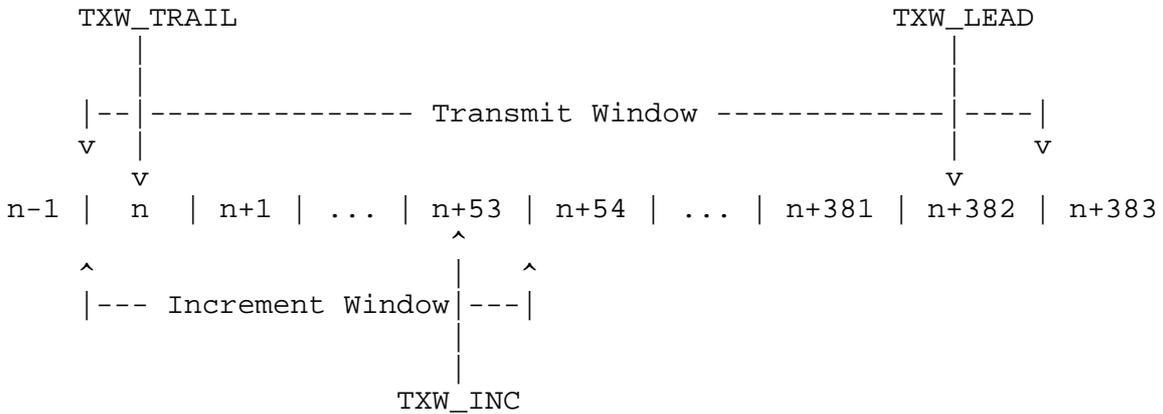
    So the values of the sequence numbers defining these windows are:

        TXW_TRAIL = n
        TXW_INC = n+53
        TXW_LEAD = n+382

    Nota Bene: In this example the window sizes in terms of sequence
    numbers can be determined only because of the assumption of a
    constant data packet size of 1500 bytes. When the data packet
    sizes are variable, more or fewer sequence numbers MAY be consumed
    transmitting the same amount (TXW_BYTES) of data.

So, for a given transport session identified by a TSI, a source
maintains:

TXW_MAX_RTE   a maximum transmit rate in kBytes per second, the
                 cumulative transmit rate of some combination of SPMs,
                 ODATA, and RDATA depending on the transmit window
                 advancement strategy

TXW_TRAIL   the sequence number defining the trailing edge of the
                 transmit window, the sequence number of the oldest
                 data packet available for repair

TXW_LEAD   the sequence number defining the leading edge of the
                 transmit window, the sequence number of the most
                 recently transmitted ODATA packet

TXW_INC   the sequence number defining the leading edge of the
                 increment window, the sequence number of the most
                 recently transmitted data packet amongst those that
                 will expire upon the next increment of the transmit
                 window

PGM does not constrain the strategies that a source may use for
advancing the transmit window. A source MAY implement any scheme or
number of schemes. This is possible because a PGM receiver must obey
the window provided by the source in its packets. Three strategies
are suggested within this document.

In the first, called "Advance with Time", the transmit window
maintains the last TXW_SECS of data in real-time, regardless of
whether any data was sent in that real time period or not. The
actual number of bytes maintained at any instant in time will vary
between 0 and TXW_BYTES, depending on traffic during the last
TXW_SECS. In this case, TXW_MAX_RTE is the cumulative transmit rate
of SPMs and ODATA.

In the second, called "Advance with Data", the transmit window
maintains the last TXW_BYTES bytes of data for repair. That is, it
maintains the theoretical maximum amount of data that could be
transmitted in the time period TXW_SECS, regardless of when they were
transmitted. In this case, TXW_MAX_RTE is the cumulative transmit
rate of SPMs, ODATA, and RDATA.

The third strategy leaves control of the window in the hands of the
application. The API provided by a source implementation for this,
could allow the application to control the window in terms of APDUs
and to manually step the window. This gives a form of Application
Level Framing (ALF). In this case, TXW_MAX_RTE is the cumulative
transmit rate of SPMs, ODATA, and RDATA.

16.2. Advancing with Data

   In the first strategy, TXW_MAX_RTE is calculated from SPMs and both
   ODATA and RDATA, and NAKs reset TXW_ADV_IVL_TMR. In this mode of
   operation the transmit window maintains the last TXW_BYTES bytes of
   data for repair. That is, it maintains the theoretical maximum
   amount of data that could be transmitted in the time period TXW_SECS.
   This means that the following timers are not treated as real-time
   timers, instead they are "data driven". That is, they expire when
   the amount of data that could be sent in the time period they define
   is sent. They are the SPM ambient time interval, TXW_ADV_SECS,
   TXW_SECS, TXW_ADV_IVL, TXW_ADV_IVL_TMR and the join interval. Note
   that the SPM heartbeat timers still run in real-time.

   While TXW_ADV_IVL_TMR is running, a source uses the receipt of a NAK
   for ODATA within the increment window to reset timer TXW_ADV_IVL_TMR
   to TXW_ADV_IVL so that transmit window advancement is delayed until
   no NAKs for data in the increment window are seen for TXW_ADV_IVL
   seconds. If the transmit window should fill in the meantime, further
   transmissions would be suspended until the transmit window can be
   advanced.

   A source MUST advance the transmit window across the increment window
   only upon expiry of TXW_ADV_IVL_TMR.

   This mode of operation is intended for non-real-time, messaging
   applications based on the receipt of complete data at the expense of
   delay.

16.3. Advancing with Time

   This strategy advances the transmit window in real-time. In this
   mode of operation, TXW_MAX_RTE is calculated from SPMs and ODATA only
   to maintain a constant data throughput rate by consuming extra
   bandwidth for repairs. TXW_ADV_IVL has the value 0 which advances
   the transmit window without regard for whether NAKs for data in the
   increment window are still being received.

   In this mode of operation, all timers are treated as real-time
   timers.

   This mode of operation is intended for real-time, streaming
   applications based on the receipt of timely data at the expense of
   completeness.

16.4. Advancing under explicit application control

   Some applications may wish more explicit control of the transmit
   window than that provided by the advance with data / time strategies
   above. An implementation MAY provide this mode of operation and
   allow an application to explicitly control the window in terms of
   APDUs.

17. Appendix G - Applicability Statement

   As stated in the introduction, PGM has been designed with a specific
   class of applications in mind in recognition of the fact that a
   general solution for reliable multicast has proven elusive. The
   applicability of PGM is narrowed further, and perhaps more
   significantly, by the prototypical nature of at least four of the
   transport elements the protocol incorporates. These are congestion
   control, router assist, local retransmission, and a programmatic API
   for reliable multicast protocols of this class. At the same time as
   standardization efforts address each of these elements individually,
   this publication is intended to foster experimentation with these
   elements in general, and to inform that standardization process with
   results from practise.

   This section briefly describes some of the experimental aspects of
   PGM and makes non-normative references to some examples of current
   practise based upon them.

   At least 3 different approaches to congestion control can be explored
   with PGM: a receiver-feedback based approach, a router-assist based
   approach, and layer-coding based approach. The first is supported by
   the negative acknowledgement mechanism in PGM augmented by an
   application-layer acknowledgement mechanism. The second is supported
   by the router exception processing mechanism in PGM. The third is
   supported by the FEC mechanisms in PGM. An example of a receiver-
   feedback based approach is provided in [16], and a proposal for a
   router-assist based approach was proposed in [17]. Open issues for
   the researchers include how do each of these approaches behave in the
   presence of multiple competing sessions of the same discipline or of
   different disciplines, TCP most notably; how do each of them behave
   over a particular range of topologies, and over a particular range of
   loads; and how do each of them scale as a function of the size of the
   receiver population.

   Router assist has applications not just to implosion control and
   retransmit constraint as described in this specification, but also to
   congestion control as described above, and more generally to any
   feature which may be enhanced by access to per-network-element state
   and processing. The full range of these features is as yet

unexplored, but a general mechanism for providing router assist in a
transport-protocol independent way (GRA) is a topic of active
research [18]. That effort has been primarily informed by the router
assist component of PGM, and implementation and deployment experience
with PGM will continue to be fed back into the specification and
eventual standardization of GRA. Open questions facing the
researchers ([19], [20], [21]) include how router-based state scales
relative to the feature benefit obtained, how system-wide factors
(such as throughput and retransmit latency) vary relative to the
scale and topology of deployed router assistance, and how incremental
deployment considerations affect the tractability of router-assist
based features. Router assist may have additional implications in
the area of congestion control to the extent that it may be applied
in multi-group layered coding schemes to increase the granularity and
reduce the latency of receiver based congestion control.

GRA itself explicitly incorporates elements of active networking, and
to the extent that the router assist component of PGM is reflected in
GRA, experimentation with the narrowly defined network-element
functionality of PGM will provide some of the first real world
experience with this promising if controversial technology.

Local retransmission is not a new idea in general in reliable
multicast, but the specific approach taken in PGM of locating re-
transmitters on the distribution tree for the session, diverting
repair requests from network elements to the re-transmitters, and
then propagating repairs downward from the repair point on the
distribution tree raises interesting questions concerning where to
locate re-transmitters in a given topology, and how network elements
locate those re-transmitters and evaluate their efficiency relative
to other available sources of retransmissions, most notably the
source itself. This particular aspect of PGM, while fully specified,
has only been implemented on the network element side, and awaits a
host-side implementation before questions like these can be
addressed.

PGM presents the opportunity to develop a programming API for
reliable multicast applications that reflects both those
applications' service requirements as well as the services provided
by PGM in support of those applications that may usefully be made
visible above the transport interface. At least a couple of host-
side implementations of PGM and a concomitant API have been developed
for research purposes ([22], [23]), and are available as open source
explicitly for the kind of experimentation described in this section.

Perhaps the broadest experiment that PGM can enable in a community of
researchers using a reasonable scale experimental transport protocol
is simply in the definition, implementation, and deployment of IP

multicast applications for which the reliability provided by PGM is a
significant enabler. Experience with such applications will not just
illuminate the value of reliable multicast, but will also provoke
practical examination of and responses to the attendant policy issues
(such as peering, billing, access control, firewalls, NATs, etc.),
and, if successful, will ultimately encourage more wide spread
deployment of IP multicast itself.

18. Abbreviations

     ACK   Acknowledgment
     AFI   Address Family Indicator
     ALF   Application Level Framing
     APDU   Application Protocol Data Unit
     ARQ   Automatic Repeat reQuest
     DLR   Designated Local Repairer
     GSI   Globally Unique Source Identifier
     FEC   Forward Error Correction
     MD5   Message-Digest Algorithm
     MTU   Maximum Transmission Unit
     NAK   Negative Acknowledgment
     NCF   NAK Confirmation
     NLA   Network Layer Address
     NNAK   Null Negative Acknowledgment
     ODATA   Original Data
     POLL   Poll Request
     POLR   Poll Response
     RDATA   Repair Data
     RSN   Receive State Notification
     SPM   Source Path Message
     SPMR   SPM Request
     TG   Transmission Group
     TGSIZE   Transmission Group Size
     TPDU   Transport Protocol Data Unit
     TSDU   Transport Service Data Unit
     TSI   Transport Session Identifier
     TSN   Transmit State Notification

19. Acknowledgements

   The design and specification of PGM has been substantially influenced
   by reviews and revisions provided by several people who took the time
   to read and critique this document. These include, in alphabetical
   order:

   Bob Albrightson
   Joel Bion
   Mark Bowles
   Steve Deering
   Tugrul Firatli
   Dan Harkins
   Dima Khoury
   Gerard Newman
   Dave Oran
   Denny Page
   Ken Pillay
   Chetan Rai
   Yakov Rekhter
   Dave Rossetti
   Paul Stirpe
   Brian Whetten
   Kyle York

20. References

   [1]   B. Whetten, T. Montgomery, S. Kaplan, "A High Performance
         Totally Ordered Multicast Protocol", in "Theory and Practice in
         Distributed Systems", Springer Verlag LCNS938, 1994.

   [2]   S. Floyd, V. Jacobson, C. Liu, S. McCanne, L. Zhang, "A
         Reliable Multicast Framework for Light-weight Sessions and
         Application Level Framing", ACM Transactions on Networking,
         November 1996.

   [3]   J. C. Lin, S. Paul, "RMTP: A Reliable Multicast Transport
         Protocol", ACM SIGCOMM August 1996.

   [4]   Miller, K., Robertson, K., Tweedly, A. and M. White, "Multicast
         File Transfer Protocol (MFTP) Specification", Work In Progress.

   [5]   Deering, S., "Host Extensions for IP Multicasting", STD 5, RFC
         1112, August 1989.

   [6]   Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

   [7]   C. Partridge, "Gigabit Networking", Addison Wesley 1994.

   [8]   H. W. Holbrook, S. K. Singhal, D. R. Cheriton, "Log-Based
         Receiver-Reliable Multicast for Distributed Interactive
         Simulation", ACM SIGCOMM 1995.

   [9]   Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April
         1992.

   [10] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC
         1700, October 1994.

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

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

   [13] V. Jacobson, "Congestion Avoidance and Control", ACM SIGCOMM
         August 1988.

   [14] Bradner, S., "Key words for use in RFCs to Indicate Requirement
         Levels", BCP, 14, RFC 2119, March 1997.

   [15] J. Bolot, T. Turletti, I. Wakeman, "Scalable Feedback Control
         for Multicast Video Distribution in the Internet", Proc.
         ACM/Sigcomm 94, pp. 58-67.

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

   [17] M. Luby, L. Vicisano, T. Speakman. "Heterogeneous multicast
         congestion control based on router packet filtering", RMT
         working group, June 1999, Pisa, Italy.

   [18] Cain, B., Speakman, T. and D. Towsley, "Generic Router Assist
         (GRA) Building Block, Motivation and Architecture", Work In
         Progress.

   [19] C. Papadopoulos, and E. Laliotis,"Incremental Deployment of a
         Router-assisted Reliable Multicast Scheme,", Proc. of Networked
         Group Communications (NGC2000), Stanford University, Palo Alto,
         CA. November 2000.

    [20] C. Papadopoulos, "RAIMS: an Architecture for Router-Assisted
          Internet Multicast Services." Presented at ETH, Zurich,
          Switzerland, October 23 2000.

    [21] J. Chesterfield, A. Diana, A. Greenhalgh, M. Lad, and M. Lim,
          "A BSD Router Implementation of PGM",
          http://www.cs.ucl.ac.uk/external/m.lad/rpgm/

    [22] L. Rizzo, "A PGM Host Implementation for FreeBSD",
          http://www.iet.unipi.it/~luigi/pgm.html

    [23] M. Psaltaki, R. Araujo, G. Aldabbagh, P. Kouniakis, and A.
          Giannopoulos, "Pragmatic General Multicast (PGM) host
          implementation for FreeBSD.",
          http://www.cs.ucl.ac.uk/research/darpa/pgm/PGM_FINAL.html

21. Authors' Addresses

    Tony Speakman
    EMail: speakman@cisco.com

    Dino Farinacci
    Procket Networks
    3850 North First Street
    San Jose, CA 95134
    USA
    EMail: dino@procket.com

    Steven Lin
    Juniper Networks
    1194 N. Mathilda Ave.
    Sunnyvale, CA 94086
    USA
    EMail: steven@juniper.net

    Alex Tweedly
    EMail: agt@cisco.com

    Nidhi Bhaskar
    EMail: nbhaskar@cisco.com

    Richard Edmonstone
    EMail: redmonst@cisco.com

    Rajitha Sumanasekera
    EMail: rajitha@cisco.com

Lorenzo Vicisano
Cisco Systems, Inc.
170 West Tasman Drive,
San Jose, CA 95134
USA
EMail: lorenzo@cisco.com

Jon Crowcroft
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
UK
EMail: j.crowcroft@cs.ucl.ac.uk

Jim Gemmell
Microsoft Bay Area Research Center
301 Howard Street, #830
San Francisco, CA 94105
USA
EMail: jgemmell@microsoft.com

Dan Leshchiner
Tibco Software
3165 Porter Dr.
Palo Alto, CA 94304
USA
EMail: dleshc@tibco.com

Michael Luby
Digital Fountain, Inc.
39141 Civic Center Drive
Fremont CA 94538
USA
EMail: luby@digitalfountain.com

Todd L. Montgomery
Talarian Corporation
124 Sherman Ave.
Morgantown, WV 26501
USA
EMail: todd@talarian.com

Luigi Rizzo
Dip. di Ing. dell'Informazione
Universita' di Pisa
via Diotisalvi 2
56126 Pisa
Italy
EMail: luigi@iet.unipi.it

22. Full Copyright Statement

   Copyright (C) The Internet Society (2001). All Rights Reserved.

   This document and translations of it may be copied and furnished to
   others, and derivative works that comment on or otherwise explain it
   or assist in its implementation may be prepared, copied, published
   and distributed, in whole or in part, without restriction of any
   kind, provided that the above copyright notice and this paragraph are
   included on all such copies and derivative works. However, this
   document itself may not be modified in any way, such as by removing
   the copyright notice or references to the Internet Society or other
   Internet organizations, except as needed for the purpose of
   developing Internet standards in which case the procedures for
   copyrights defined in the Internet Standards process must be
   followed, or as required to translate it into languages other than
   English.

   The limited permissions granted above are perpetual and will not be
   revoked by the Internet Society or its successors or assigns.

   This document and the information contained herein is provided on an
   "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
   TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
   BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
   HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
   MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

   Funding for the RFC Editor function is currently provided by the
   Internet Society.