        Packetization Layer Path MTU Discovery for Datagram Transports
                draft-ietf-tsvwg-datagram-plpmtud-01

Abstract

   This document describes a robust method for Path MTU Discovery
   (PMTUD) for datagram Packetization layers. The method allows a
   Packetization Layer (PL), or a datagram application that uses a PL,
   to probe an network path with progressively larger packets to
   determine a maximum packet size. The document describes an extension
   to RFC 1191 and RFC 8201, which specify ICMP-based Path MTU Discovery
   for IPv4 and IPv6. This provides functionally for datagram
   transports that is equivalent to the Packetization layer PMTUD
   specification for TCP, specified in RFC4821.

   When published, this specification updates RFC4821.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Table of Contents

1. Introduction

   The IETF has specified datagram transport using UDP, SCTP, and DCCP,
   as well as protocols layered on top of these transports (e.g., SCTP/
   UDP, DCCP/UDP).

1.1. Classical Path MTU Discovery

   Classical Path Maximum Transmission Unit Discovery (PMTUD) can be
   used with any transport that is able to process ICMP Packet Too Big
   (PTB) messages (e.g., [RFC1191] and [RFC8201]). The term PTB message
   is applied to both IPv4 ICMP Unreachable messages (type 3) that carry
   the error Fragmentation Needed (Type 3, Code 4) and ICMPv6 packet too
   big messages (Type 2). When a sender receives a PTB message, it
   reduces the effective Path MTU (PMTU) to the value reported as the
   Link MTU in the PTB message, and a method that from time-to-time
   increases the packet size in attempt to discover an increase in the
   supported PMTU. The packets sent with a size larger than the current
   effective PMTU are known as probe packets.

   Packets not intended as probe packets are either fragmented to the
   current effective PMTU, or the attempt to send fails with an error
   code. Applications are sometimes provided with a primitive to let
   them read the maximum packet size, derived from the current effective
   PMTU.

   Classical PMTUD is subject to protocol failures. One failure arises
   when traffic using a packet size larger than the actual supported
   PMTU is black-holed (all datagrams sent with this size are silently

discarded without the sender receiving ICMP PTB messages. This could
arise when the ICMP messages are not delivered back to the sender for
some reason [RFC2923]). For example, ICMP messages are increasingly
filtered by middleboxes (including firewalls) [RFC4890]. Also, in
some cases are not correctly processed by tunnel endpoints.

Another failure could result if a node not on the network path sends
a PTB that attempts to force the sender to change the effective PMTU
[RFC8201]. A sender can protect itself from reacting to such
messages by utilising the quoted packet within the PTB message
payload to verify that the received PTB message was generated in
response to a packet that had actually been sent. However, there are
situations where a sender would be unable to provide this
verification.

Examples where verification is not possible include:

o When the router issuing the ICMP message is acting on a tunneled
   packet the ICMP message is directed to the tunnel endpoint. This
   endpoint is responsible for processed in the quoted packet in the
   payload field to remove the effect of the tunnel, and return the
   ICMP message to the sender. Failure to do this results in black-
   holing.

o When the router issuing the ICMP message implements RFC792
   [RFC0792], which only requires the quoted payload to include the
   first 64 bits of the IP payload of the packet, and the ICMP
   message occurs within a tunnel. Even if the decpasulated message
   is processed by the tunnel endpoint, there could be insufficient
   bytes remaining for the sender to read the quoted transport
   information. RFC1812 [RFC1812] requires routers to return the
   full packet if possible, often the case for IPv4 when used the
   path includes tunnels; or where the packet has been encapsulated/
   tunneled over an encrypted transport and it is not possible to
   determine the original transport header).

o Even when the PTB message includes sufficient bytes of the quoted
   packet, the network layer could lack sufficient context to perform
   verification, because this depends on information about the active
   transport flows at an endpoint node (e.g., the socket/address
   pairs being used, and other protocol header information).

1.2. Packetization Layer Path MTU Discovery

The term Packetization Layer (PL) has been introduced to describe the
layer that is responsible for placing data blocks into the payload of
packets and selecting an appropriate maximum packet size. This
function is often performed by a transport protocol, but can also be

performed by other encapsulation methods working above the transport.
PTB verification is more straight forward at the PL or at a higher
layer.

In contrast to PMTUD, Packetization Layer Path MTU Discovery
(PLPMTUD) [RFC4821] does not rely upon reception and verification of
PTB messages. It is therefore more robust than Classical PMTUD.
This has become the recommended approach for implementing PMTU
discovery with TCP.

It uses a general strategy where the PL sends probe packet to search
for an appropriate PMTU. The probe packets are sent a progressively
larger packet size. If a probe packet is successfully delivered (as
determined by the PL), then the effective Path MTU is raised to the
size of the successful probe. If no response is received to a probe
packet, the method reduces the probe size.

PLPMTUD introduces flexibility in the implementation of PMTU
discovery. At one extreme, it can be configured to only perform PTB
black hole detection and recovery to increase the robustness of
Classical PMTUD, or at the other extreme, all PTB processing can be
disabled and PLPMTUD can completely replace Classical PMTUD. PLPMTUD
can also include additional consistency checks without increasing the
risk of increased black-holing.

1.3. Path MTU Discovery for Datagram Services

Section 4 of this document presents a set of algorithms for datagram
protocols to discover a maximum size for the effective PMTU across a
path. The methods described rely on features of the PL Section 3 and
apply to transport protocols over IPv4 and IPv6. It does not require
cooperation from the lower layers (except that they are consistent
about which packet sizes are acceptable). A method can utilise ICMP
PTB messages when these received messages are made available to the
PL.

The UDP-Guidelines [RFC8085] state "an application SHOULD either use
the Path MTU information provided by the IP layer or implement Path
MTU Discovery (PMTUD)", but does not provide a mechanism for
discovering the largest size of unfragmented datagram than can be
used on a path. Prior to this document, PLPMTUD had not been
specified for UDP.

Section 10.2 of [RFC4821] recommends a PLPMTUD probing method for the
Stream Control Transport Protocol (SCTP). SCTP utilises heartbeat
messages as probe packets, but RFC4821 does not provide a complete
specification. This document provides the details to complete that
specification.

The Datagram Congestion Control Protocol (DCCP) [RFC4340] requires
implementations to support Classical PMTUD and states that a DCCP
sender "MUST maintain the maximum packet size (MPS) allowed for each
active DCCP session". It also defines the current congestion control
maximum packet size (CCMPS) supported by a path. This recommends use
of PMTUD, and suggests use of control packets (DCCP-Sync) as path
probe packets, because they do not risk application data loss. The
method defined in this specification could be used with DCCP.

Section 5 specifies the method for a set of transports, and provides
information to enables the implementation of PLPMTUD with other
datagram transports and applications that use datagram transports.

2. 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 [RFC2119].

Other terminology is directly copied from [RFC4821], and the
definitions in [RFC1122].

Black-Holed: When the sender is unaware that packets are not
   delivered to the destination endpoint (e.g., when the sender
   transmits packets of a particular size with a previously known
   PMTU, but is unaware of a change to the path that resulted in a
   smaller PMTU).

Classical Path MTU Discovery: Classical PMTUD is a process described
   in [RFC1191] and [RFC8201], in which nodes rely on PTB messages to
   learn the largest size of unfragmented datagram than can be used
   across a path.

Datagram: A datagram is a transport-layer protocol data unit,
   transmitted in the payload of an IP packet.

Effective PMTU: The current estimated value for PMTU that is used by
   a Packetization Layer.

EMTU_S: The Effective MTU for sending (EMTU_S) is defined in
   [RFC1122] as "the maximum IP datagram size that may be sent, for a
   particular combination of IP source and destination addresses...".

EMTU_R: The Effective MTU for receiving (EMTU_R) is designated in
   [RFC1122] as the largest datagram size that can be reassembled by
   EMTU_R ("Effective MTU to receive").

Link: A communication facility or medium over which nodes can
      communicate at the link layer, i.e., a layer below the IP layer.
      Examples are Ethernet LANs and Internet (or higher) layer and
      tunnels.

Link MTU: The Maximum Transmission Unit (MTU) is the size in bytes
      of the largest IP packet, including the IP header and payload,
      that can be transmitted over a link. Note that this could more
      properly be called the IP MT, to be consistent with how other
      standards organizations use the acronym MT. This includes the IP
      header, but excludes link layer headers and other framing that is
      not part of IP or the IP payload. Other standards organizations
      generally define link MTU to include the link layer headers.

MPS: The Maximum Packet Size (MPS), the largest size of application
      data block that can be sent unfragmented across a path. In
      PLPMTUD this quantity is derived from Effective PMTU by taking
      into consideration the size of the application and lower protocol
      layer headers, and can be limited by the application protocol.

Packet: An IP header plus the IP payload.

Packetization Layer (PL): The layer of the network stack that places
      data into packets and performs transport protocol functions.

Path: The set of link and routers traversed by a packet between a
      source node and a destination node.

Path MTU (PMTU): The minimum of the link MTU of all the links
      forming a path between a source node and a destination node.

PLPMTUD: Packetization Layer Path MTU Discovery, the method
      described in this document for datagram PLs, which is an extension
      to Classical PMTU Discovery.

Probe packet: A datagram sent with a purposely chosen size
      (typically larger than the current Effective PMTU or MPS) to
      detect if messages of this size can be successfully sent along the
      end-to-end path.

3. Features Required to Provide Datagram PLPMTUD

TCP PLPMTUD has been defined using standard TCP protocol mechanisms.
All of the requirements in [RFC4821] also apply to use of the
technique with a datagram PL. Unlike TCP, some datagram PLs require
additional mechanisms to implement PLPMTUD.

There are nine requirements for performing the datagram PLPMTUD method described in this specification:

1. PMTU parameters: A PLPMTUD sender is REQUIRED to provide information about the maximum size of packet that can be transmitted by the sender on the local link (the link MTU and MAY utilize similar information about the receiver when this is supplied (note this could be less than EMTU_R). Some applications also have a maximum transport protocol data unit (PDU) size, in which case there is no benefit from probing for a size larger than this (unless a transport allows multiplexing multiple applications PDUs into the same datagram).

2. Effective PMTU: A datagram application MUST be able to choose the size of datagrams sent to the network, up to the effective PMTU, or a smaller value (such as the MPS) derived from this. This value is managed by the PMTUD method. The effective PMTU (specified in Section 1 of [RFC1191]) is equivalent to the EMTU_S (specified in [RFC1122]).

3. Probe packets: On request, a PLPMTUD sender is REQUIRED to be able to transmit a packet larger than the current effective PMTU (but always with a total size less than the link MTU). The method can use this as a probe packet. In IPv4, a probe packet is always sent with the Don't Fragment (DF) bit set in the IP header, and without network layer endpoint fragmentation. In IPv6, a probe packet is always sent without source fragmentation (as specified in section 5.4 of [RFC8201]).

4. Processing PTB messages: A PLPMTUD sender MAY optionally utilize PTB messages received from the network layer to help identify when a path does not support the current size of packet probe. Any received PTB message SHOULD/MUST be verified before it is used to update the PMTU discovery information [RFC8201]. This verification confirms that the PTB message was sent in response to a packet originating by the sender, and needs to be performed before the PMTU discovery method reacts to the PTB message. When the router link MTU is indicated in the PTB message this MAY be used by datagram PLPMTUD to reduce the size of a probe, but MUST NOT be used to increase the effective PMTU ([RFC8201]).

5. Reception feedback: The destination PL endpoint is REQUIRED to provide a feedback method that indicates to the PLPMTUD sender when a probe packet has been received by the destination endpoint. The local PL endpoint at the sending node is REQUIRED to pass this feedback to the sender-side PLPMTUD method.

6. Probing and congestion control: The isolated loss of a probe
    packet SHOULD NOT be treated as an indication of congestion and
    its loss does not directly trigger a congestion control reaction
    [RFC4821].

7. Probe loss recovery: If the data block carried by a probe message
    needs to be sent reliably, the PL (or layers above) MUST arrange
    retransmission/repair of any resulting loss. This method MUST be
    robust in the case where probe packets are lost due to other
    reasons (including link transmission error, congestion). The
    PLPMTUD method treats isolated loss of a probe packet (with or
    without an PTB message) as a potential indication of a PMTU limit
    on the path. The PL MAY retransmit any data included in a lost
    probe packet without adjusting its congestion window [RFC4821].

8. Cached effective PMTU: The sender MUST cache the effective PMTU
    value used by an instance of the PL between probes and needs also
    to consider the disruption that could be incurred by an
    unsuccessful probe - both upon the flow that incurs a probe loss,
    and other flows that experience the effect of additional probe
    traffic.

9. Shared effective PMTU state: The PMTU value could also be stored
    with the corresponding entry in the destination cache and used by
    other PL instances. The specification of PLPMTUD [RFC4821]
    states: "If PLPMTUD updates the MTU for a particular path, all
    Packetization Layer sessions that share the path representation
    (as described in Section 5.2 of [RFC4821]) SHOULD be notified to
    make use of the new MTU and make the required congestion control
    adjustments". Such methods need to robust to the wide variety of
    underlying network forwarding behaviours. Section 5.2 of
    [RFC8201] provides guidance on the caching of PMTU information
    and also the relation to IPv6 flow labels.

In addition the following design principles are stated:

o Suitable MPS: The PLPMTUD method SHOULD avoid forcing an
   application to use an arbitrary small MPS (effective PMTU) for
   transmission while the method is searching for the currently
   supported PMTU. Datagram PLs do not necessarily support
   fragmentation of PDUs larger than the PMTU. A reduced MPS can
   adversely impact the performance of a datagram application.

o Path validation: The PLPMTUD method MUST be robust to path changes
   that could have occurred since the path characteristics were last
   confirmed.

   o Datagram reordering: A method MUST be robust to the possibility
      that a flow encounters reordering, or has the traffic (including
      probe packets) is divided over more than one network path.

   o When to probe: The PLPMTUD method SHOULD determine whether the
      path capacity has increased since it last measured the path. This
      determines when the path should again be probed.

3.1. PMTU Probe Packets

   PMTU discovery relies upon the sender being able to generate probe
   messages with a specific size. TCP is able to generate probe packets
   by choosing to appropriately segment data being sent [RFC4821].

   In contrast, a datagram PL that needs to construct a probe packet has
   to either request an application to send a data block that is larger
   than that generated by an application, or to utilise padding
   functions to extend a datagram beyond the size of the application
   data block. Protocols that permit exchange of control messages
   (without an application data block) could alternatively prefer to
   generate a probe packet by extending a control message with padding
   data.

   When the method fails to validate the PMTU for the path, it may be
   required to send a probe packet with a size less than the size of the
   data block generated by an application. In this case, the PL could
   provide a way to fragment a datagram at the PL, or could instead
   utilise a control packet with padding.

   A receiver needs to be able to distinguish an in-band data block from
   any added padding. This is needed to ensure that any added padding
   is not passed on to an application at the receiver.

   This results in three possible ways that a sender can create a probe
   packet:

   Probing using appication data: A probe packet that contains a data
      block supplied by an application that matches the size required
      for the probe. This method requests the application to issue a
      data block of the desired probe size. If the application/
      transport needs protection from the loss of an unsuccessful probe
      packet, the application/transport needs then to perform transport-
      layer retransmission/repair of the data block (e.g., by
      retransmission after loss is detected or by duplicating the data
      block in a datagram without the padding).

   Probing using appication data and padding data: A probe packet that
      contains a data block supplied by an application that is combined

   with padding to inflate the length of the datagram to the size
   required for the probe. If the application/transport needs
   protection from the loss of this probe packet, the application/
   transport may perform transport-layer retransmission/repair of the
   data block (e.g., by retransmission after loss is detected or by
   duplicating the data block in a datagram without the padding
   data).

   Probing using padding data: A probe packet that contains only
   control information together with any padding needed to inflate
   the packet to the size required for the probe. Since these probe
   packets do not carry an application-supplied data block,they do
   not typically require retransmission, although they do still
   consume network capacity and incur endpoint processing.

   A datagram PLPMTUD MAY choose to use only one of these methods to
   simplify the implementation.

3.2. Validation of the Current Effective PMTU

   The PL needs a method to determine when probe packets have been
   successfully received end-to-end across a network path.

   Transport protocols can include end-to-end methods that detect and
   report reception of specific datagrams that they send (e.g., DCCP and
   SCTP provide keep-alive/heartbeat features). When supported, this
   mechanism SHOULD also be used by PLPMTUD to acknowledge reception of
   a probe packet.

   A PL that does not acknowledge data reception (e.g., UDP and UDP-
   Lite) is unable to detect when the packets it sends are discarded
   because their size is greater than the actual PMTUD. These PLs need
   to either rely on an application protocol to detect this, or make use
   of an additional transport method such as UDP-Options
   [I-D.ietf-tsvwg-udp-options]. In addition, they might need to send
   reachability probes (e.g., periodically solicit a response from the
   destination) to determine whether the current effective PMTU is still
   supported by the network path.

   Section Section 4 specifies this function for a set of IETF-specified
   protocols.

3.3. Reduction of the Effective PMTU

   When the current effective PMTU is no longer supported by the network
   path, the transport needs to detect this and reduce the effective
   PMTU.

   o A PL that sends a datagram larger than the actual PMTU that
      includes no application data block, or one that does not attempt
      to provide any retransmission, can send a new probe packet with an
      updated probe size.

   o A PL that wishes to resend the application data block, could then
      need to re-fragment the data block to a smaller packet size that
      is expected to traverse the end-to-end path. This could utilise
      network-layer or PL fragmentation when these are available. A
      fragmented datagram MUST NOT be used as a probe packet (see
      [RFC8201]).

   A method can additionally utilise PTB messages to detect when the
   actual PMTU supported by a network path is less than the current size
   of datagrams (or probe messages) that are being sent.

4. Datagram Packetization Layer PMTUD

   This section specifies Datagram PLPMTUD.

   The central idea of PLPMTU discovery is probing by a sender. Probe
   packets of increasing size are sent to find out the maximum size of a
   user message that is completely transferred across the network path
   from the sender to the destination.

4.1. Probing

   The PLPMTUD method utilises a timer to trigger the generation of
   probe packets. The probe_timer is started each time a probe packet
   is sent to the destination and is cancelled when receipt of the probe
   packet is acknowledged.

   The PROBE_COUNT is initialised to zero when a probe packet is first
   sent with a particular size. Each time the probe_timer expires, the
   PROBE_COUNT is incremented, and a probe packet of the same size is
   retransmitted. The maximum number of retransmissions per probing
   size is configured (MAX_PROBES). If the value of the PROBE_COUNT
   reaches MAX_PROBES, probing will be stopped and the last successfully
   probed PMTU is set as the effective PMTU.

   Once probing is completed, the sender continues to use the effective
   PMTU until either a PTB message is received or the PMTU_RAISE_TIMER
   expires. If the PL is unable to verify reachability to the
   destination endpoint after probing has completed, the method uses a
   REACHABILITY_TIMER to periodically repeat a probe packet for the
   current effective PMTU size, while the PMTU_RAISE_TIMER is running.
   If the resulting probe packet is not acknowledged (i.e. the
   PROBE_TIMER expires), the method re-starts probing for the PMTU.

4.2. Verification and Use of PTB Messages

   This section describes processing for both IPv4 ICMP Unreachable
   messages (type 3) and ICMPv6 packet too big messages.

   A node that receives a PTB message from a router or middlebox, MUST
   verify the PTB message. The node checks the protocol information in
   the quoted payload to verify that the message originated from the
   sending node. The node also checks that the reported MTU size is
   less than the size used by packet probes. PTB messages are discarded
   if they fail to pass these checks, or where there is insufficient
   ICMP payload to perform these checks. The checks are intended to
   provide protection from packets that originate from a node that is
   not on the network path or a node that attempts to report a larger
   MTU than the current probe size.

   PTB messages that have been verified can be utilised by the DPLPMTUD
   algorithm. A method that utilises these PTB messages can improve
   performance compared to one that relies solely on probing.

4.3. Timers

   This method utilises three timers:

   PROBE_TIMER: Configured to expire after a period longer than the
      maximum time to receive an acknowledgment to a probe packet. This
      value MUST be larger than 1 second, and SHOULD be larger than 15
      seconds. Guidance on selection of the timer value are provide in
      section 3.1.1 of the UDP Usage Guidelines [RFC8085].

   PMTU_RAISE_TIMER: Configured to the period a sender ought to
      continue use the current effective PMTU, after which it re-
      commences probing for a higher PMTU. This timer has a period of
      600 secs, as recommended by PLPMTUD [RFC4821].

   REACHABILITY_TIMER: Configured to the period a sender ought to wait
      before confirming the current effective PMTU is still supported.
      This is less than the PMTU_RAISE_TIMER.

      An application that needs to employ keep-alive messages to deliver
      useful service over UDP SHOULD NOT transmit them more frequently
      than once every 15 seconds and SHOULD use longer intervals when
      possible. DPLPMTUD ought to suspend reachability probes when no
      application data has been sent since the previous probe packet.
      Guidance on selection of the timer value are provide in section
      3.1.1 of the UDP Usage Guidelines[RFC8085].

An implementation could implement the various timers using a single
timer process.

4.4. Constants

The following constants are defined:

MAX_PROBES: The maximum value of the PROBE_ERROR_COUNTER. The
    default value of MAX_PROBES is 10.

MIN_PMTU: The smallest allowed probe packet size. This value is
    1280 bytes, as specified in [RFC2460]. For IPv4, the minimum
    value is 68 bytes. (An IPv4 routed is required to be able to
    forward a datagram of 68 octets without further fragmentation.
    This is the combined size of an IPv4 header and the minimum
    fragment size of 8 octets.)

BASE_PMTU: The BASE_PMTU is a considered a size that ought to work
    in most cases. The size is equal to or larger than the minimum
    permitted and smaller than the maximum allowed. In the case of
    IPv6, this value is 1280 bytes [RFC2460]. When using IPv4, a size
    of 1200 is RECOMMENDED.

MAX_PMTU: The MAX_PMTU is the largest size of PMTU that is probed.
    This has to be less than or equal to the minimum of the local MTU
    of the outgoing interface and the destination effective MTU for
    receiving. An application or PL may reduce this when it knows
    there is no need to send packets above a specific size.

4.5. Variables

This method utilises a set of variables:

effective PMTU: The effective PMTU is the maximum size of datagram
    that the method has currently determined can be supported along
    the entire path.

PROBED_SIZE: The PROBED_SIZE is the size of the current probe
    packet. This is a tentative value for the effective PMTU, which
    is awaiting confirmation by an acknowledgment.

PROBE_COUNT: This is a count of the number of unsuccessful probe
    packets that have been sent with size PROBED_SIZE. The value is
    initialised to zero when a particular size of PROBED_SIZE is first
    attempted.

PTB_SIZE: The PTB_Size is value returned by a verified PTB message
    indicating the local MTU size of a router along the path.

4.6. Selecting PROBED_SIZE

Implementations discover the search range by validating the minimum
path MTU and then using the probe method to select a PROBED_SIZE less
than or equal to the maximum PMTU_MAX. Where PMTU_MAX is the minimum
of the the local link MTU and EMTU_R (learned from the remote
endpoint). The PMTU_MAX MAY be constrained by an application that
has a maximum to the size of datagrams it wishes to send.

Implementations use a search algorithm to choose probe sizes within
the search range. XXX The current method does not specify or
recommend a specific methods for selecting a probe size. One simple
method is to increase the size of probe in increments until it fails,
other methods may use tables to select probe sizes, or search
algorithms - this part to be expanded based on experience and
consideration of methods XXX

Implementations MAY optimizse the search procedure by selecting step
sizes from a table of common MTU sizes.

Implementations SHOULD select probe sizes to maximise the gain in
PMTU each search step. Implementations ought to take into
consideration useful probe size steps and a minimum useful gain in
PMTU.

4.7. State Machine

A state machine for Datagram PLPMTUD is depicted in Figure 1. If
multihoming is supported, a state machine is needed for each active
path.

```
                            PROBE_TIMER expiry
                          (PROBE_COUNT = MAX_PROBES)
                    +-------------+                +--------------+
                =->| PROBE_START |-------------->|PROBE_DISABLED|
   PROBE_TIMER expiry | +-------------+                +--------------+
  (PROBE_COUNT =      |    |       |
       MAX_PROBES) -------       |  Connectivity confirmed
                                 v
             ----------- +------------+ -- PROBE_TIMER expiry
  MAX_PMTU acked or |            | PROBE_BASE |  | (PROBE_COUNT <
  PTB (>= BASE_PMTU)|     -----> +------------+ <-          MAX_PROBES)
    ---------------     |          /\    |  |
    |                   |          |  |  | PTB
    |     PMTU_RAISE_TIMER|        |  |  | (PTB_SIZE < BASE_PMTU)
    |     or reachability |        |  |  |        or
    |      (PROBE_COUNT   |        |  |  |    PROBE_TIMER expiry
    |       = MAX_PROBES) |        |  |  | (PROBE_COUNT = MAX_PROBES)
    |        ------------          |  |  \
    |        |           PTB |     |  |   \
    |        |     (< PROBED_SIZE)|  |    \
    |        |                    |  |     ---------------
    |        |                    |  |                   |
    |        |                    |  | Probe             |
    |        |                    |  | acked             |
    v        |                    |  v                   v
  +------------+                +--------------+  Probe +-------------+
  | PROBE_DONE |<-------------- | PROBE_SEARCH |<-------| PROBE_ERROR |
  +------------+ MAX_PMTU acked +--------------+  acked +-------------+
   /\    |          or              /\       |
   |     |     PROBE_TIMER expiry    |       |
   |     |(PROBE_COUNT = MAX_PROBES) |       |
   |     |                          |       |
   ------                         --------
  Reachability probe acked       PROBE_TIMER expiry
   or PROBE_TIMER expiry       (PROBE_COUNT < MAX_PROBES)
  (PROBE_COUNT < MAX_PROBES)             or
                               Probe acked
```
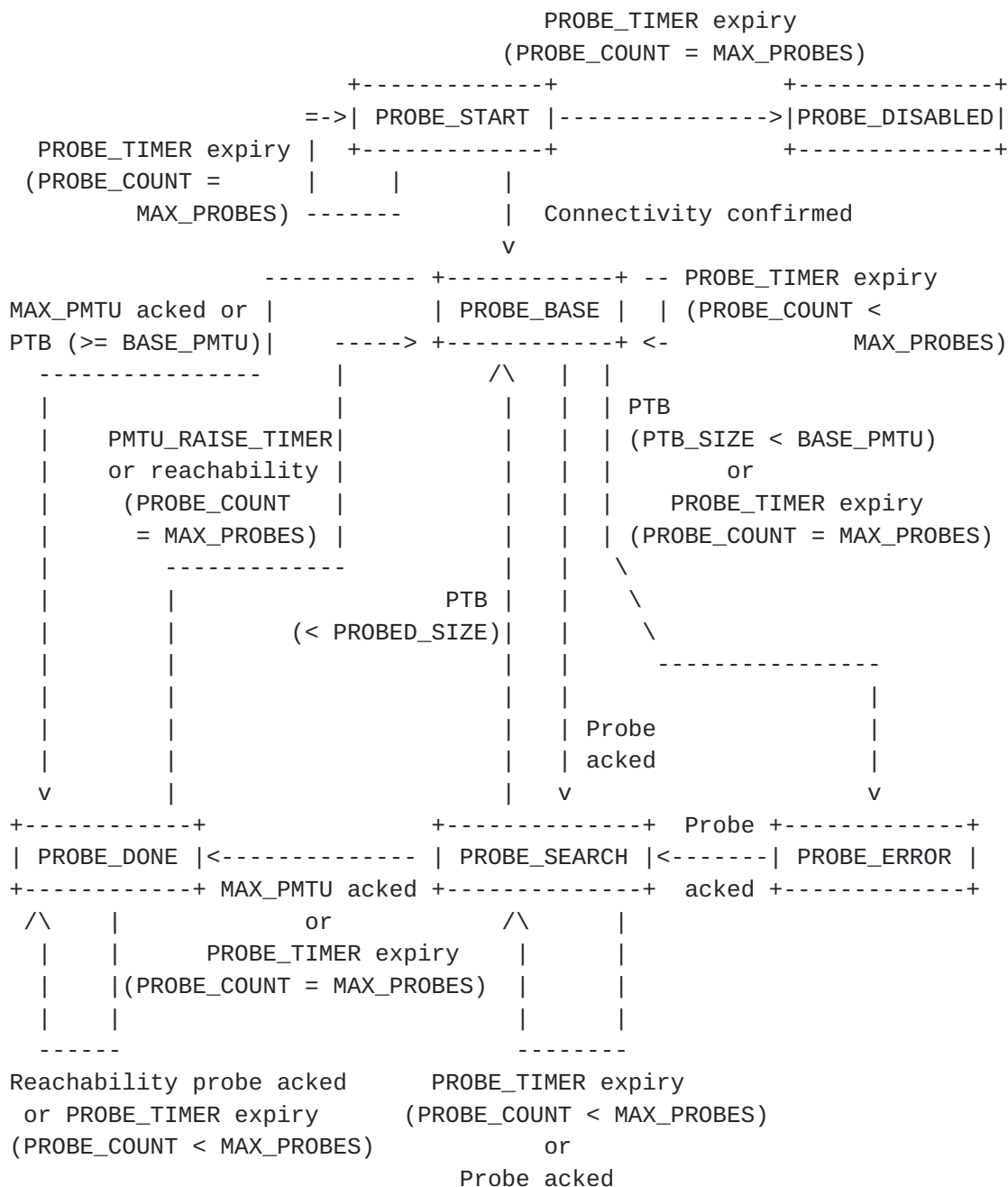
Figure 1: State machine for Datagram PLPMTUD

   XXX State machine to be updated to describe handling of validated PTB
   messages XXX

   XXX Method may be updated to clarify how probe sizes are used during
   probing XXX

   The following states are defined to reflect the probing process:

PROBE_START: The PROBE_START state is the initial state before
   probing has started. PLPMTUD is not performed in this state. The
   state transitions to PROBE_BASE, when a path has been confirmed,
   i.e. when a sent packet has been acknowledged on this path. The
   effective PMTU is set to the BASE_PMTU size. Probing ought to
   start immediately after connection setup to prevent the loss of
   user data.

PROBE_BASE: The PROBE_BASE state is the starting point for probing
   with datagram PLPMTUD. It is used to confirm whether the
   BASE_PMTU size is supported by the network path. On entry, the
   PROBED_SIZE is set to the BASE_PMTU size and the PROBE_COUNT is
   set to zero. A probe packet is sent, and the PROBE_TIMER is
   started. The state is left when the PROBE_COUNT reaches
   MAX_PROBES; a PTB message is verified, or a probe packet is
   acknowledged.

PROBE_SEARCH: The PROBE_SEARCH state is the main probing state.
   This state is entered either when probing for the BASE_PMTU was
   successful or when there is a successful reachability test in the
   PROBE_ERROR state. On entry, the effective PMTU is set to the
   last acknowledged PROBED_SIZE.

   The PROBE_COUNT is set to zero when the first probe packet is sent
   for each probed size. Each time a probe packet is acknowledged,
   the effective PMTU is set to the PROBED_SIZE, and then the
   PROBED_SIZE is increased.

   When a probe packet is sent and not acknowledged within the period
   of the PROBE_TIMER, the PROBE_COUNT is incremented and the probe
   packet is retransmitted. The state is exited when the PROBE_COUNT
   reaches MAX_PROBES; a PTB message is verified; or a probe of size
   PMTU_MAX is acknowledged.

PROBE_ERROR: The PROBE_ERROR state represents the case where the
   network path is not known to support an effective PMTU of at least
   the BASE_PMTU size. It is entered when either a probe of size
   BASE_PMTU has not been acknowledged or a verified PTB message
   indicates a smaller link MTU than the BASE_PMTU. On entry, the
   PROBE_COUNT is set to zero and the PROBED_SIZE is set to the
   MIN_PMTU size, and the effective PMTU is reset to MIN_PMTU size.
   In this state, a probe packet is sent, and the PROBE_TIMER is
   started. The state transitions to the PROBE_SEARCH state when a
   probe packet is acknowledged.

PROBE_DONE: The PROBE_DONE state indicates a successful end to a
   probing phase. Datagram PLPMTUD remains in this state until

either the PMTU_RAISE_TIMER expires or a received PTB message is
verified.

When PLPMTUD uses an unacknowledged PL and is in the PROBE_DONE
state, a REACHABILITY_TIMER periodically resets the PROBE_COUNT
and schedules a probe packet with the size of the effective PMTU.
If the probe packet fails to be acknowledged after MAX_PROBES
attempts, the method enters the PROBE_BASE state. When used with
an acknowledged PL (e.g., SCTP), DPLPMTUD SHOULD NOT continue to
probe in this state.

PROBE_DISABLED: The PROBE_DISABLED state indicates that connectivity
   could not be established. DPLPMTUD MUST NOT probe in this state.

Appendix A contains an informative description of key events.

5. Specification of Protocol-Specific Methods

This section specifies protocol-specific details for datagram PLPMTUD
for IETF-specified transports.

5.1. DPLPMTUD for UDP and UDP-Lite

The current specifications of UDP [RFC0768] and UDP-LIte [RFC3828] do
not define a method in the RFC-series that supports PLPMTUD. In
particular, these transports do not provide the transport layer
features needed to implement datagram PLPMTUD, and any support for
Datagram PLPMTUD would therefore need to rely on higher-layer
protocol features [RFC8085].

5.1.1. UDP Options

UDP-Options [I-D.ietf-tsvwg-udp-options] supply the additional
functionality required to implement datagram PLPMTUD. This enables
padding to be added to UDP datagrams and can be used to provide
feedback acknowledgement of received probe packets.

5.1.2. UDP Options Required for PLPMTUD

This subsection proposes two new UDP-Options that add support for
requesting a datagram response be sent and to mark this datagram as a
response to a request.

XXX Future versions of the spec may define a parameter in an Option
to indicate the EMTU_R to the peer that can be used to initialise
PMTU_MAX.   XXX

5.1.2.1. Echo Request Option

The Echo Request Option allows a sending endpoint to solicit a
response from a destination endpoint.

The Echo Request carries a four byte token set by the sender. This
token can be set to a value that is likely to be known only to the
sender (and becomes known to nodes along the end-to-end path). The
sender can then check the value returned in the response to provide
additional protection from off-path insertion of data [RFC8085].

```
           +---------+--------+-----------------+
           | Kind=9  | Len=6  |      Token      |
           +---------+--------+-----------------+
             1 byte    1 byte       4 bytes
```
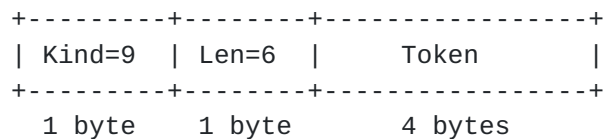
Figure 2: UDP ECHOREQ Option Format

5.1.2.2. Echo Response Option

The Echo Response Option is generated by the PL in response to
reception of a previously received Echo Request. The Token field
associates the response with the Token value carried in the most
recently-received Echo Request. The rate of generation of UDP
packets carrying an Echo Response Option MAY be rate-limited.

```
           +---------+--------+-----------------+
           | Kind=10 | Len=6  |      Token      |
           +---------+--------+-----------------+
             1 byte    1 byte       4 bytes
```
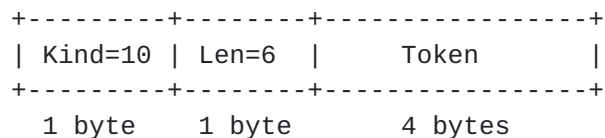
Figure 3: UDP ECHORES Option Format

5.1.3. Sending UDP-Option Probe Packets

This method specifies a probe packet that does not carry an
application data block. The probe packet consists of a UDP datagram
header followed by a UDP Option containing the ECHOREQ option, which
is followed by NOP Options to pad the remainder of the datagram
payload to the probe size. NOP padding is used to control the length
of the probe packet.

A UDP Option carrying the ECHORES option is used to provide feedback
when a probe packet is received at the destination endpoint.

5.1.4. Validating the Path with UDP Options

   Since UDP is an unacknowledged PL, a sender that does not have
   higher-layer information confirming correct delivery of datagrams
   SHOULD implement the REACHABILITY_TIMER to periodically send probe
   packets while in the PROBE_DONE state.

5.1.5. Handling of PTB Messages by UDP

   Normal ICMP verification MUST be performed as specified in
   Section 5.2 of [RFC8085]. This requires that the PL verifies each
   received PTB messages to verify these are received in response to
   transmitted traffic and that the reported LInk MTU is less than the
   current probe size. A verified PTB message MAY be used as input to
   the PLPMTUD algorithm.

5.2. DPLPMTUD for SCTP

   Section 10.2 of [RFC4821] specifies a recommended PLPMTUD probing
   method for SCTP. It recommends the use of the PAD chunk, defined in
   [RFC4820] to be attached to a minimum length HEARTBEAT chunk to build
   a probe packet. This enables probing without affecting the transfer
   of user messages and without interfering with congestion control.
   This is preferred to using DATA chunks (with padding as required) as
   path probes.

   XXX Future versions of this specification might define a parameter
   contained in the INIT and INIT ACK chunk to indicate the MTU to the
   peer. However, multihoming makes this a bit complex, so it might not
   be worth doing. XXX

5.2.1. SCTP/IP4 and SCTP/IPv6

   The base protocol is specified in [RFC4960].

5.2.1.1. Sending SCTP Probe Packets

   Probe packets consist of an SCTP common header followed by a
   HEARTBEAT chunk and a PAD chunk. The PAD chunk is used to control
   the length of the probe packet. The HEARTBEAT chunk is used to
   trigger the sending of a HEARTBEAT ACK chunk. The reception of the
   HEARTBEAT ACK chunk acknowledges reception of a successful probe.

   The HEARTBEAT chunk carries a Heartbeat Information parameter which
   should include, besides the information suggested in [RFC4960], the
   probing size, which is the MTU size the complete datagram will add up
   to. The size of the PAD chunk is therefore computed by reducing the
   probing size by the IPv4 or IPv6 header size, the SCTP common header,

the HEARTBEAT request and the PAD chunk header. The payload of the
PAD chunk contains arbitrary data.

To avoid fragmentation of retransmitted data, probing starts right
after the handshake, before data is sent. Assuming normal behaviour
(i.e., the PMTU is smaller than or equal to the interface MTU), this
process will take a few round trip time periods depending on the
number of PMTU sizes probed. The Heartbeat timer can be used to
implement the PROBE_TIMER.

5.2.1.2. Validating the Path with SCTP

Since SCTP provides an acknowledged PL, a sender does MUST NOT
implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.2.1.3. PTB Message Handling by SCTP

Normal ICMP verification MUST be performed as specified in Appendix C
of [RFC4960]. This requires that the first 8 bytes of the SCTP
common header are quoted in the payload of the PTB message, which can
be the case for ICMPv4 and is normally the case for ICMPv6.

When a PTB message has been verified, the router Link MTU indicated
in the PTB message SHOULD be used with the PLPMTUD algorithm,
providing that the reported Link MTU is less than the current probe
size.

5.2.2. DPLPMTUD for SCTP/UDP

The UDP encapsulation of SCTP is specified in [RFC6951].

5.2.2.1. Sending SCTP/UDP Probe Packets

Packet probing can be performed as specified in Section 5.2.1.1. The
maximum payload is reduced by 8 bytes, which has to be considered
when filling the PAD chunk.

5.2.2.2. Validating the Path with SCTP/UDP

Since SCTP provides an acknowledged PL, a sender does MUST NOT
implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.2.2.3. Handling of PTB Messages by SCTP/UDP

Normal ICMP verification MUST be performed for PTB messages as
specified in Appendix C of [RFC4960]. This requires that the first 8
bytes of the SCTP common header are contained in the PTB message,
which can be the case for ICMPv4 (but note the UDP header also

consumes a part of the quoted packet header) and is normally the case
for ICMPv6. When the verification is completed, the router Link MTU
size indicated in the PTB message SHOULD be used with the PLPMTUD
algorithm providing that the reported LInk MTU is less than the
current probe size.

5.2.3. DPLPMTUD for SCTP/DTLS

The Datagram Transport Layer Security (DTLS) encapsulation of SCTP is
specified in [I-D.ietf-tsvwg-sctp-dtls-encaps]. It is used for data
channels in WebRTC implementations.

5.2.3.1. Sending SCTP/DTLS Probe Packets

Packet probing can be done as specified in Section 5.2.1.1.

5.2.3.2. Validating the Path with SCTP/DTLS

Since SCTP provides an acknowledged PL, a sender does MUST NOT
implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.2.3.3. Handling of PTB Messages by SCTP/DTLS

It is not possible to perform normal ICMP verification as specified
in [RFC4960], since even if the ICMP message payload contains
sufficient information, the reflected SCTP common header would be
encrypted. Therefore it is not possible to process PTB messages at
the PL.

5.3. PMTUD for QUIC

XXX New section XXX

Quick UDP Internet Connection (QUIC) is a UDP-based transport that
provides reception feedback [I-D.ietf-quic-transport].

Section 9.2 of [I-D.ietf-quic-transport] details the path
considerations when sending QUIC packets. It reccomends the use of
PADDING frames to buld the probe packet. This enables probing the
without affecting the transfer of other frames.

5.3.1. Sending QUIC Probe Packets

Probe packets consist of a QUIC Header and a payload containing only
PADDING Frames. PADDING Frames are a single octet (0x00) and
serveral of these can be used to create a probe packet of size
PROBED_SIZE.

A QUIC sender needs to pad initial packets to 1200 bytes to validate
the path can support packets of a useful size. If a QUIC sender
determines the PMTU on a path has fallen below 1280 octets it MUST
immediately stop sending on the affected path.

5.3.2. Validating the Path with QUIC

Since QUIC provides an acknowledged PL, a sender does MUST NOT
implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.3.3. Handling of PTB Messages by QUIC

QUIC does not specify any methods for validating ICMP responses, but
does provide some guidlines to make it harder for an off path
attacker to inject ICMP messages.

o Set the IPv4 Don't Fragment (DF) bit on a small proportion of
   packets, so that most invalid ICMP messages arrive when there are
   no DF packets outstanding, and can therefore be identified as
   spurious.

o Store additional information from the IP or UDP headers from DF
   packets (for example, the IP ID or UDP checksum) to further
   authenticate incoming Datagram Too Big messages.

o Any reduction in PMTU due to a report contained in an ICMP packet
   is provisional until QUIC's loss detection algorithm determines
   that the packet is actually lost.

XXX The above list was pulled whole from quic-transport XXX

5.4. Other IETF Transports

XXX This section to be updated in a later revision. XXX

5.5. DPLPMTUD by Applications

Applications that use the Datagram API (e.g., applications built
directly or indirectly on UDP) can implement DPLPMTUD. Some
primitives used by DPLPMTUD might not be available via this interface
(e.g., the ability to access the PMTU cache, or interpret received
ICMP PTB messages).

In addition, it is important that PMTUD is not performed by multiple
protocol layers.

XXX This section will be completed in a future revision of this ID
XXX

6. Acknowledgements

7. IANA Considerations

This memo includes no request to IANA.

XXX If new UDP Options are specified in this document, a request to
IANA will be included here. XXX

If there are no requirements for IANA, the section will be removed
during conversion into an RFC by the RFC Editor.

8. Security Considerations

The security considerations for the use of UDP and SCTP are provided
in the references RFCs. Security guidance for applications using UDP
is provided in the UDP-Guidelines [RFC8085].

PTB messages could potentially be used to cause a node to
inappropriately reduce the effective PMTU. A node supporting PLPMTUD
MUST appropriately verify the payload of PTB messages to ensure these
are received in response to transmitted traffic (i.e., a reported
error condition that corresponds to a datagram actually sent by the
path layer.

XXX Determine if parallel forwarding paths needs to be considered.
XXX

A node performing PLPMTUD could experience conflicting information
about the size of supported probe packets. This could occur when
there are multiple paths are concurrently in use and these exhibit a
different PMTU. If not considered, this could result in data being
blackholed when the effective PMTU is larger than the smallest PMTU
across the current paths.

9. References

9.1. Normative References

[I-D.ietf-quic-transport]
          Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
          and Secure Transport", draft-ietf-quic-transport-04 (work
          in progress), June 2017.

   [I-D.ietf-tsvwg-sctp-dtls-encaps]
              Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "DTLS
              Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-
              dtls-encaps-09 (work in progress), January 2015.

   [I-D.ietf-tsvwg-udp-options]
              Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-
              udp-options-01 (work in progress), June 2017.

   [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768,
              DOI 10.17487/RFC0768, August 1980, <https://www.rfc-
              editor.org/info/rfc768>.

   [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5,
              RFC 792, DOI 10.17487/RFC0792, September 1981,
              <https://www.rfc-editor.org/info/rfc792>.

   [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts -
              Communication Layers", STD 3, RFC 1122,
              DOI 10.17487/RFC1122, October 1989, <https://www.rfc-
              editor.org/info/rfc1122>.

   [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers",
              RFC 1812, DOI 10.17487/RFC1812, June 1995,
              <https://www.rfc-editor.org/info/rfc1812>.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997, <https://www.rfc-
              editor.org/info/rfc2119>.

   [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6
              (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
              December 1998, <https://www.rfc-editor.org/info/rfc2460>.

   [RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., Ed.,
              and G. Fairhurst, Ed., "The Lightweight User Datagram
              Protocol (UDP-Lite)", RFC 3828, DOI 10.17487/RFC3828, July
              2004, <https://www.rfc-editor.org/info/rfc3828>.

   [RFC4820] Tuexen, M., Stewart, R., and P. Lei, "Padding Chunk and
              Parameter for the Stream Control Transmission Protocol
              (SCTP)", RFC 4820, DOI 10.17487/RFC4820, March 2007,
              <https://www.rfc-editor.org/info/rfc4820>.

   [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol",
              RFC 4960, DOI 10.17487/RFC4960, September 2007,
              <https://www.rfc-editor.org/info/rfc4960>.

   [RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream
              Control Transmission Protocol (SCTP) Packets for End-Host
              to End-Host Communication", RFC 6951,
              DOI 10.17487/RFC6951, May 2013, <https://www.rfc-
              editor.org/info/rfc6951>.

   [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage
              Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085,
              March 2017, <https://www.rfc-editor.org/info/rfc8085>.

   [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed.,
              "Path MTU Discovery for IP version 6", STD 87, RFC 8201,
              DOI 10.17487/RFC8201, July 2017, <https://www.rfc-
              editor.org/info/rfc8201>.

9.2. Informative References

   [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191,
              DOI 10.17487/RFC1191, November 1990, <https://www.rfc-
              editor.org/info/rfc1191>.

   [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery",
              RFC 2923, DOI 10.17487/RFC2923, September 2000,
              <https://www.rfc-editor.org/info/rfc2923>.

   [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram
              Congestion Control Protocol (DCCP)", RFC 4340,
              DOI 10.17487/RFC4340, March 2006, <https://www.rfc-
              editor.org/info/rfc4340>.

   [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU
              Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007,
              <https://www.rfc-editor.org/info/rfc4821>.

   [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering
              ICMPv6 Messages in Firewalls", RFC 4890,
              DOI 10.17487/RFC4890, May 2007, <https://www.rfc-
              editor.org/info/rfc4890>.
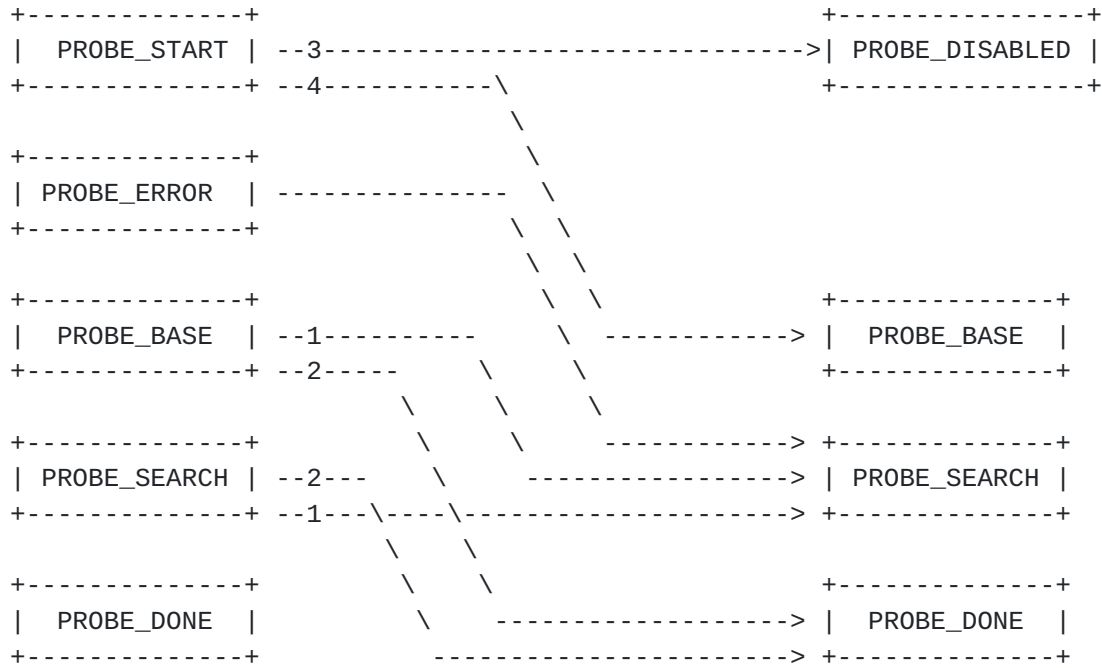
Appendix A. Event-driven state changes

   This appendix contains an informative description of key events:

   Path Setup: When a new path is initiated, the state is set to
      PROBE_START. As soon as the path is confirmed, the state changes
      to PROBE_BASE and the probing mechanism for this path is started.
      the first probe packet is sent with the size of the BASE_PMTU.

Arrival of an Acknowledgment: Depending on the probing state, the
    reaction differs according to Figure 4, which is just a
    simplification of Figure 1 focusing on this event.

```
+--------------+                                   +---------------+
|  PROBE_START | --3----------------------------->| PROBE_DISABLED |
+--------------+ --4-----------\                   +---------------+
                                \
+--------------+                 \
| PROBE_ERROR  | --------------   \
+--------------+              \  \
                              \  \
+--------------+              \  \                 +--------------+
|  PROBE_BASE  | --1----------     \  ----------->  |  PROBE_BASE  |
+--------------+ --2-----      \     \              +--------------+
                       \        \     \
+--------------+        \        \      ----------->  +--------------+
| PROBE_SEARCH | --2---    \      ---------------->  | PROBE_SEARCH |
+--------------+ --1---\----\------------------->  +--------------+
                       \     \
+--------------+        \     \                    +--------------+
|  PROBE_DONE  |          \    ------------------>  |  PROBE_DONE  |
+--------------+           ----------------------> +--------------+
```

Condition 1: The maximum PMTU size has not yet been reached.
Condition 2: The maximum PMTU size has been reached. Conition 3:
Probe Timer expires and PROBE_COUNT = MAX_PROBEs. Condition 4:
PROBE_ACK received.

Figure 4: State changes at the arrival of an acknowledgment

Probing timeout: The PROBE_COUNT is initialised to zero each time
    the value of PROBED_SIZE is changed. The PROBE_TIMER is started
    each time a probe packet is sent. It is stopped when an
    acknowledgment arrives that confirms delivery of a probe packet.
    If the probe packet is not acknowledged before the PROBE_TIMER
    expires, the PROBE_ERROR_COUNTER is incremented. When the
    PROBE_COUNT equals the value MAX_PROBES, the state is changed,
    otherwise a new probe packet of the same size (PROBED_SIZE) is
    resent. The state transitions are illustrated in Figure 5. This
    shows a simplification of Figure 1 with a focus only on this
    event.

```
  +--------------+                                        +---------------+
  |  PROBE_START |--------------------------------------->| PROBE_DISABLED |
  +--------------+                                        +---------------+


  +--------------+                                        +--------------+
  | PROBE_ERROR  |                  ---------------> | PROBE_ERROR  |
  +--------------+                 /                      +--------------+
                                  /
  +--------------+ --2----------/                         +--------------+
  |  PROBE_BASE  | --1----------------------------> |  PROBE_BASE  |
  +--------------+                                        +--------------+


  +--------------+                                        +--------------+
  | PROBE_SEARCH | --1----------------------------> | PROBE_SEARCH |
  +--------------+ --2---------                           +--------------+
                              \
  +--------------+             \                          +--------------+
  |  PROBE_DONE  |              ------------------> |  PROBE_DONE  |
  +--------------+                                        +--------------+
```
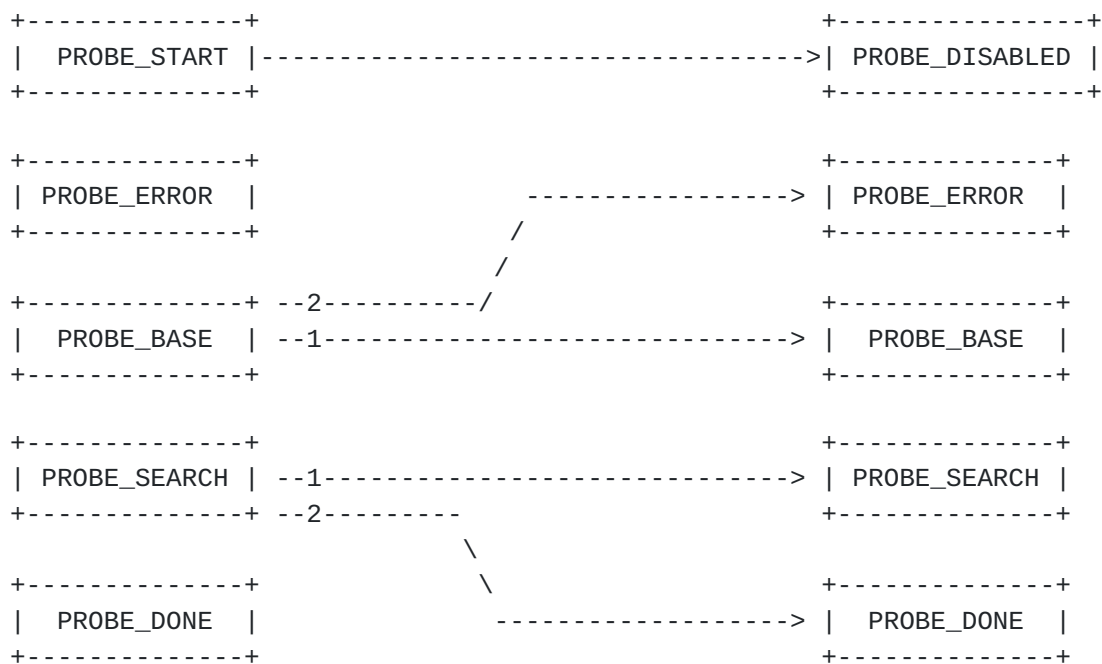
   Condition 1: The maximum number of probe packets has not been
   reached. Condition 2: The maximum number of probe packets has been
   reached.

      Figure 5: State changes at the expiration of the probe timer

   PMTU raise timer timeout: The path through the network can change
      over time. It impossible to discover whether a path change has
      increased the actual PMTU by exchanging packets less than or equal
      to the effective PMTU. This requires PLPMTUD to periodically send
      a probe packet to detect whether a larger PMTU is possible. This
      probe packet is generated by the PMTU_RAISE_TIMER. When the timer
      expires, probing is restarted with the BASE_PMTU and the state is
      changed to PROBE_BASE.

   Arrival of an ICMP message: The active probing of the path can be
      supported by the arrival of PTB messages sent by routers or
      middleboxes with a link MTU that is smaller than the probe packet
      size. If the PTB message includes the router link MTU, three
      cases can be distinguished:

      1. The indicated link MTU in the PTB message is between the
          already probed and effective MTU and the probe that triggered
          the PTB message.

2. The indicated link MTU in the PTB message is smaller than the
    effective PMTU.

3. The indicated link MTU in the PTB message is equal to the
    BASE_PMTU.

In first case, the PROBE_BASE state transitions to the PROBE_ERROR
state. In the PROBE_SEARCH state, a new probe packet is sent with
the sized reported by the PTB message. Its result is handled
according to the former events.

The second case could be a result of a network re-configuration.
If the reported link MTU in the PTB message is greater than the
BASE_MTU, the probing starts again with a value of PROBE_BASE.
Otherwise, the method enters the state PROBE_ERROR.

In the third case, the maximum possible PMTU has been reached.
This ought to be probed again, because there could be a link
further along the path with a still smaller MTU.

Note: Not all routers include the link MTU size when they send a
PTB message. If the PTB message does not indicate the link MTU,
the probe is handled in the same way as condition 2 of Figure 5.

Appendix B. Revision Notes

Note to RFC-Editor: please remove this entire section prior to
publication.

Individual draft -00:

o Comments and corrections are welcome directly to the authors or
   via the IETF TSVWG working group mailing list.

o This update is proposed for WG comments.

Individual draft -01:

o Contains the first representation of the algorithm, showing the
   states and timers

o This update is proposed for WG comments.

Individual draft -02:

o Contains updated representation of the algorithm, and textual
   corrections.

   o The text describing when to set the effective PMTU has not yet
      been verified by the authors

   o To determine security to off-path-attacks: We need to decide
      whether a received PTB message SHOULD/MUST be verified? The text
      on how to handle a PTB message indicating a link MTU larger than
      the probe has yet not been verified by the authors

   o No text currently describes how to handle inconsistent results
      from arbitrary re-routing along different parallel paths

   o This update is proposed for WG comments.

   Working Group draft -00:

   o This draft follows a successful adoption call for TSVWG

   o There is still work to complete, please comment on this draft.

   Working Group draft -01:

   o This draft includes improved introduction.

   o The draft is updated to require ICMP validation prior to accepting
      PTB messages - this to be confirmed by WG

   o Section added to discuss Selection of Probe Size - methods to be
      evlauated and recommendations to be considered

   o Section added to align with work proposed in the QUIC WG.

Authors' Addresses

   Godred Fairhurst
   University of Aberdeen
   School of Engineering
   Fraser Noble Building
   Aberdeen AB24 3U
   UK

   Email: gorry@erg.abdn.ac.uk

   Tom Jones
   University of Aberdeen
   School of Engineering
   Fraser Noble Building
   Aberdeen   AB24 3U
   UK

   Email: tom@erg.abdn.ac.uk

   Michael Tuexen
   Muenster University of Applied Sciences
   Stegerwaldstrasse 39
   Stein fart   48565
   DE

   Email: tuexen@fh-muenster.de

   Irene Ruengeler
   Muenster University of Applied Sciences
   Stegerwaldstrasse 39
   Stein fart   48565
   DE

   Email: i.ruengeler@fh-muenster.de