                     Path MTU discovery solution space
                  draft-troan-6man-pmtu-solution-space-00

Abstract

   Path MTU discovery has turned out to be a thorny problem that has
   haunted the Internet community for decades.  Lately there has been
   some work both at the transport layer and at the network layer.  This
   memo lists the solutions the author is aware of from the perspective
   of the network layer.

Status of This Memo

Copyright Notice

1.  Introduction

   Path MTU discovery has turned out to be harder than expeced.  In IPv6
   we set out following the same model as for IPv4.  The sending host
   maintains a MTU cache, that is updated based on received ICMP PMTUD
   messages.  That solution has a few short-comings:

   o  Sending of ICMP PMTUD messages is throttled in routers [RFC4443]

   o  It's not efficient if links along the path have decreasingly
      smaller MTU, then multiple rounds of large packet, resulting ICMP
      PMTUD happens.

   o  ICMP might be ignored by host stacks / applications

   o  As ICMP looks different than application traffic, it might be
      blocked by routers.

   o  Doesn't work well in an anycast scenario (but what does).

2.  Requirements / Goals

   1.  Avoid MTU black-holes [RFC2923].

   2.  Detect the Path MTU in single round trip.

   3.  Adapt to varying MTU over the connection life time.

   4.  The signalling of the MTU back to the sender must be
       indistinguishable from application traffic to lessen risk of
       filtering.

   5.  Design a mechanism that ensures that neither MTU probes nor MTU
       signalling back to sender are more likely to be dropped than
       other application traffic.

   6.  Must be deployable and anchored in transport / application areas.
       Otherwise https://xkcd.com/927/

   7.  [Optional?]  Support neighbors on the same link which support
       higher MTU than link MTU see [I-D.van-beijnum-multi-mtu]

3.  Network layer solutions for Path MTU discovery

   o  PMTUD [RFC8201]

   o  On-path fragmentation, IPv4 style.  We know this one.

   o  Packet truncation.  [I-D.leddy-6man-truncate].  The source sets a
      truncation elligble flag in the packet, routers on the path may
      truncate f the packet is too big, and sets a truncated done flag.
      Then the receiver signals the learnt forward MTU back to the
      sender.  Either via existing ICMP PMTUD or a transport layer
      option.  This is an example of a solution which does not require
      the sender having to accept packets from intermediate nodes.

   o  MTU recording.  Probe packets are sent, either as part of data
      packets, if those are guaranteed not to exceed MTU.  Some trigger
      in the header (ECN like flags) or a HBH option is required for the
      router to record the smallest MTU along the path.  Application /
      Transport would have to periodically include the probe trigger in
      data packets to detect changes in path MTU.

3.1.  Common problems

   How is the router along the path "triggered" to put this packet on
   the exception path?  For current and the truncation scheme it's a
   simple check in the forwarding path for the size of packet versus
   outgoing interface MTU.  For e.g. a recording MTU mechanism it would
   have to be flags in the IPv6 header or an HBH option.

   How should the forward path MTU be signalled back to the sender?  The
   signal should look like any other application traffic to avoid
   filtering or is it sufficient to avoid sending from intermitent
   nodes.

4.  Solutions at other layers

   In addition there are solutions at the transport layer, that work in
   co-hort or independently of the network layer soltusions.  [RFC4821]
   and [I-D.ietf-tsvwg-datagram-plpmtud].

   One could also imagine other solutions, e.g. to include MTU in router
   advertisements in BGP, so that a BGP speaker could calculate the end
   to end MTU across the set of administrative domains.

5.  Conclusion

   What are our options?  Even if we developed a new PMTU mechanism, IP
   stacks must deal with networks where the new mechanism isn't yet
   deployed.  Will a new mechanism be so much better that it provides
   enough value for it to be deployed?  Or should we at the network
   layer just punt this to transport?

6.  References

   [I-D.ietf-tsvwg-datagram-plpmtud]
             Fairhurst, G., Jones, T., Tuexen, M., and I. Ruengeler,
             "Packetization Layer Path MTU Discovery for Datagram
             Transports", draft-ietf-tsvwg-datagram-plpmtud-04 (work in
             progress), September 2018.

   [I-D.leddy-6man-truncate]
             Leddy, J. and R. Bonica, "IPv6 Packet Truncation", draft-
             leddy-6man-truncate-04 (work in progress), June 2018.

   [I-D.van-beijnum-multi-mtu]
             Beijnum, I., "Extensions for Multi-MTU Subnets", draft-
             van-beijnum-multi-mtu-05 (work in progress), March 2016.

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

   [RFC4443]  Conta, A., Deering, S., and M. Gupta, Ed., "Internet
             Control Message Protocol (ICMPv6) for the Internet
             Protocol Version 6 (IPv6) Specification", STD 89,
             RFC 4443, DOI 10.17487/RFC4443, March 2006,
             <https://www.rfc-editor.org/info/rfc4443>.

   [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>.

   [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>.

Author's Address

   Ole Troan
   Cisco Systems
   Philip Pedersens vei 1
   Lysaker  1366
   Norway

   Email: ot@cisco.com