IPv6 Oversized Packets Analysis

draft-vasilenko-v6ops-ipv6-oversized-analysis

Eduard Vasilenko vasilenko.eduard@huawei.com (Huawei)

Dmitriy Khaustov Dmitriy.Khaustov@rt.ru (Rostelecom)

The Problem Statement – oversized is becoming a problem again

Reasons:

- Intelligence is moving to the Data Plane,
  many new headers are specified to be added at transit,
  some new headers (like SRv6) could be very big (216 bytes)
- Massive ECMP (tens of thousands of paths) makes PMTU very dynamic
- Tunneling is limiting PMTU below 1500B very often
- Reassembly buffer (EMTU_R) is regulated as 1500B by default
- Less than 220B are available between 1500B and 1280B (IPv6 minimum)
- DC/Cloud could generate packets much bigger than typical MTU (up to jumbo)

Consequence:

- Many recent drafts to solve the oversized problem

IPv6 disallows fragmentation in transit. Hence, the source should decrease PMTU.
The source should be signaled.

Solutions for oversized packets:
1. Provision links with big enough MTU

Hindrances:

- Old routers and switches
- Most of the Middleboxes
- Other link-layer technologies (especially wireless) with small MTU
- Rented links from other carriers with limited MTU
- Buffer inefficiency may be bad if not sharing buffers for many small packets
- 1500B is the default, reconfiguration may need some additional efforts
- DC/Cloud could generate packets bigger than link MTU (2500B-9000B)

Conclusion: The best solution, if available.

Solutions for oversized packets:
2. Frugal usage of Extension Headers

The router could try to analyze and predict PMTU for a specific flow.
If all domain's MTUs are available on the router
then router could refrain from activation of some functionality for a specific flow.

Deficiencies:

- Some functionality should be blocked if oversized is predicted
- Difficult to predict PMTU with massive ECMP parallelism and local hash calculations
- Needs software upgrade but then bigger MTU is the better approach
- Needs protocol extension to collect MTUs of the domain
  (ISIS, BGP-LS, PCEP, SR-Policy)
- Some headers (iOAM, BIERv6, APN6) do not have interface structure at the point of header attachment – no easy point to attach checking mechanism and keep statistics
- Difficult to snoop ICMP PTB at transit for cases of failed prediction

Conclusion:
This solution has limited applicability but generated a lot of interest recently (many drafts).

Solutions for oversized packets:
3. Fragmentation and reassembly at the tunnel ends

Frequentation and reassembly in transit.

Deficiencies:

- Contradict to IPv6 architecture
- Contradict to the long list of RFCs: MPLS, L2TPv3, VxLAN, GRE, NVO3
- Expensive for reassembly (bigger buffer is needed)
- Many related problems: breaks ECMP, stateful processing, policy routing, and has many security attack vectors

Unfortunately, it is the only way if the source packet is already minimum (1280B) and overhead is big enough. IPv6 tunneling RFC 2473 and GRE assumes fragmentation for this case.

Conclusion: Fragmentation is the least probable solution for oversized packet drops.

Solutions for oversized packets:
4. PMTUD by original packet source

Proxy ICMP PTB as soon as possible (preferable after 1st ICMP PTB).

Advantages:

- Primary IPv6 architecture decision (“it is strongly recommended …”)
- Compatible with massive ECMP
- Requirement of IPv6 tunneling RFC 2473 to proxy 1st PTB
- Requirement of VxLAN, NVO3, IPSec, MPLS, or GRE
  is to set the MTU of the virtual interface
  then send PTB from it after the 2nd oversized packet from the host

Flow Label (if copied) could help to choose the same path for PTB.
A recommendation is possible here that does not contradict any RFC.

Conclusion: The best solution for the problem.

Solutions for oversized packets:
5. Packetization Layer MTU Discovery

[PLPMTUD]/[DPLPMTUD] advantages:

- more visibility (could see the size of transport layer buffers)
- could operate under the absence of ICMPv6 PTB (too much filtering)
- could be very granular (per-flow)

[PLPMTUD]/[DPLPMTUD] restrictions:

- not universal for all transport protocols
- need more resources from the host
- are challenging to share PMTU information between applications
- need much more round trip times to find suitable PMTU
- do not work well on congested paths

[PLPMTUD]: “Packetization Layer Path MTU Discovery (PLPMTUD) **is most efficient when used in conjunction with the ICMP-based Path MTU Discovery**”.

Conclusion: PLPMTUD could be considered as a redundancy mechanism for PMTUD.

Conclusion

It is better not to have a problem with oversized packets in the first place.
Upgrade all links to a bigger MTU, if possible.

The host could have MTU as big as a transit node. It would be never possible to deprecate PMTUD.
It is important to follow the recommendations of [PMTUD] and [IPv6 Tunneling] for ICMPv6 PTB message delivery to the original traffic source. Tunnel sources should perform the relay function to make sure that the original traffic source would get the PTB message faster.

The temporary 220B limit for all headers pushes us to the frugal implementation of new extension headers. This limit would be alleviated after all backbone links would be upgraded to a much bigger MTU than 1500B. Additional protocols to collect MTU information could help in the transition period to attach additional headers frugally. It is true for all new protocols: SRv6, SFC, BIERv6, iOAM, APN6.

[PLPMTUD] and [DPLPMTUD] are not the replacement for [PMTUD] but could help in some scenarios.

Fragmentation is not at all a solution for oversized packet drops.