

Limits on Sending and Processing IPv6 Extensions Headers

draft-herbert-6man-eh-limits-00

Tom Herbert <tom@sipanda.io>

Problem

- Deployment and support for IPv6 extension headers in the Internet is underwhelming
- A major reason that is that TLVs and VLHs are hard to process efficiently especially in hardware
- Problem exasperated by the fact that there few limits on the usage extension headers

To save extension headers, we need to limit them!

Solution

- Specify a set of limits that may be applied to various aspects for sending and receiving extension header
- Goals
 - Define practical limits to promote efficient and feasible implementation
 - Don't restrict functionality in extension headers to the point that they no longer useful!

Related work

- **RFC7045**: Cannot rely on routers to process HbH
- **RFC7872**: measurements on extension header drops
- **draft-ietf-v6ops-ipv6-ehs-packet-drops**: Provides some reasons why nodes may drop packets with EH
- **RFC8883**: ICMP errors when discarding packets due to processing limits being exceeded
- **RFC8504**: limits on EH for hosts
- **draft-hinden-6man-hbh-processing**: specific limit on HbH options
- **RFC9000**: QUIC limits EH to 32 bytes for initial packets

Amended Robust Principle

- Be conservative in what you send
- Liberal in what you receive
- If you possess explicit knowledge of receivers, you can be less conservative, more liberal, in what you send

Types of limits

- Limits on processing of extension headers
- Limits on length of the IPv6 header chain

Limiting processing of ext. headers

- Limits on number of extensions headers, number of options & padding in DestOps & HbH opts, etc.
- Processing long lists of TLVs and VLH and inherently serialized and difficult to do “fast-path”
- Example, a sender could send a packet with 700 HbH options: only purpose is DoS attack!

Limiting IPv6 header chain length

- Motivated by common practice that intermediate routers parse into the transport layer in a packet
- Implementations may have limited capabilities to parse into a packet (e.g. hardware parsing buffer)

Public Internet v. Limited Domains

Applying the amended robustness principle

- When sending into public Internet, apply restrictive limits
- When sending into limited domain, limits can be relaxed
 - Limited domains not normatively defined, work around that with “a priori” knowledge of the path
 - Attaining knowledge: sending in closed network, probing, historical information, mash-up map of capabilities in the Internet
 - Some EH & options, like SRv6 and IOAM HbH, are intended only for use in limited domains-- restrictive limits are not relevant to them

Application of limits

| | Identification | Limits (Hdr. chain applies to all receivers) | Behavior when limit exceeded |
|---------------------------------|--|--|--|
| Host: sending EH | Source of packet | Limit setting EH in packets | --- |
| Host: receiving EH | Final destination address | Limits on all extension headers | Drop packet |
| Router | Intermediate node not in destination address | Limits on processing HbH options | Ignore data beyond the limit and forward |
| Intermediate destination | Non-final destination in routing header | Limits on HbH, DO before RH, RH | HbH options limits: Router behavior DestOpt, RH limits: Host behavior |

Default Sending Host Limits

- Hosts **MUST NOT** send a packet with an EH longer than Hosts **MUST NOT** send more than eight non-padding DO or HbH options
- Hosts **MUST** not send consecutive pad options, padding **MUST** be ≤ 7 bytes
- Hosts **MUST** limit Data Length in HbH and DO options to sixty bytes

Default Sending Host Limits

- Hosts **MUST NOT** send a packet with length of EH header chain greater than 104 bytes
- Derivation:
 - 128 bytes minimal parsing buffer for interm. nodes
 - Evidence suggests minimum is commonly supported
 - Less than that is not viable for ext. hdrs.
 - $128 - (16 \text{ bytes L2 hdr.}, 8 \text{ bytes transport hdr.}) = 104$
- **Allows 64 bytes for extension headers**
 - $104 - 40 \text{ bytes IPv6 header} = 64 \text{ bytes}$

Optional Receiving Limits

- # non-padding DO or HbH options (default 8)
- Maximum length of DO or HbH options
- Maximum length of DO, HbH, RH EH
- Padding ≤ 7 bytes
- No consecutive pad options
- Limits on IPv6 header chain length

Notes on receive limits

- If behavior of a limit being exceeded in to drop, RX limits should be \geq sending limits
- Limit of HbH means an intermediate node can decide to process only first N options
 - In this draft **$\#_opts \geq N \geq 0$**
 - RFC246: **$N = \#_opts$**
 - RFC8200: **$N = 0$ or $N = \#_opts$**
 - In draft-hinden-6man-hbh-processing **$N > 0$**

Thank you!