



Limits on Sending and Processing IPv6 Extensions Headers

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

Tom Herbert

IETF112

Overview

- 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

(Related goal: Processing TLVs in hardware fast path)

Types of limits

- Limits on processing of extension headers
- Limits on length of the IPv6 header chain
- Limits on senders and receivers-- need to adhere to robustness principle for both

Default Sending Host Limits

- Hosts **MUST NOT** send a packet with an EH longer than 64 bytes
- 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
- Hosts **MUST NOT** send a packet with length of EH header chain greater than 104 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

Requirements

	Identification	Applicable imits (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

Exceeding default sending limits

- Note that all sending limits are defaults that can be exceeded if a sender has knowledge that it is safe to do so
- How to obtain such knowledge?
 - Sending within a limit domain (static route configuration)
 - More advanced: “Happy eyeballs like probing
 - Send packets with EH exceeding limits, if seeing responses continue, else fallback
 - Extrapolation: Probing could create database that maps `<src, dst>` to supported features or parameters

Implementation: TLV loop for receive

```
while (offset < len) {
    if (cp[offset] == EOL) return 0;
    if (cp[offset] == PAD1) tlv_len = 1;
    else {
        if (len - offset < 2) return -1;
        tlvlen = cp[offset + 1];
        if (cp[offset] != PADN) {
            handler = Lookup(cp[offset], TLV_Table);
            if !(handler(&cp[offset])) return -1;
        }
    }
    offset += tlvlen;
}
return 0;
```

Implementation: TLV loop for receive

```
while (offset < len) {  
    if (cp[offset] == EOL) return 0;  
    if (cp[offset] == PAD1) tlv_len = 1;  
    else {  
        if (len - offset < 2) return -1;  
        tlvlen = cp[offset + 1];  
        if (cp[offset] != PADN) {  
            handler = Lookup(cp[offset], TLV_Table);  
            if !(handler(&cp[offset])) return -1;  
        }  
    }  
    offset += tlvlen;  
}  
return 0;
```

Limits on EH header chain, limits on size, number of EH

Limit for all options count

Limits on padding

Limits on options length

Limits on non-padding options count

Implementation: TLV loop for receive

```
while (offset < len) {  
    if (cp[offset] == EOL) return 0;  
    if (cp[offset] == PAD1) tlv_len = 1;  
    else {  
        if (len - offset < 2) return -1;  
        tlvlen = cp[offset + 1];  
        if (cp[offset] != PADN) {  
            handler = Lookup(cp[offset], TLV_Table);  
            if !(handler(&cp[offset])) return -1;  
        }  
    }  
    offset += tlvlen;  
}  
return 0;
```

Limits on EH header chain, limits on size, number of EH

Limit for all options count

Limits on padding

Limits on options length

Limits on non-padding options count

TLV_LOOP(PAD1, PADN, EOL, TLV_Table, Limits)

Next steps

WG adoption in 6man?

Thank you!