Guidance to limit size of protocol headers in packets
draft-herbert-int-area-limits-00

Abstract

This specification provides guidance for limiting the size of
protocol headers in packets. Intermediate nodes that perform deep
packet inspection (DPI), particularly hardware implementations, are
often limited as to how many bytes of protocol headers they can
process. We recommend that 256 bytes be established as the minimum
number of bytes of headers that intermediate nodes are expected to be
able to process. This limit is explicitly not intended to be a
standard requirement, however it can be used as a guideline in
hardware design, protocol design, as well as a useful constraint in
implementation or configuration.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF), its areas, and its working groups. Note that
other groups may also distribute working documents as
Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

Copyright and License Notice
1 Introduction

In this specification we recommend that 256 bytes be established as the minimum size of protocol headers in a packet that nodes are expected to be capable of processing. This limit can be taken as a guideline for hardware design, protocol design, as well as a potential constraint in implementation or configuration. This packet header size limit is a recommendation only, not a requirement.

Intermediate nodes often perform deep packet inspection (DPI) in order to implement various functions in the network. DPI occurs when an intermediate node parses a packet beyond network layer headers. Hardware devices often have constraints on how much of the headers in a packet can be parsed for DPI. A typical design is that some portion of the beginning of a received packet is loaded into a memory buffer for header parsing. The size of this buffer is often fixed.

To derive the size limit for protocol headers we need to take into account headers in a packet that might be subject to DPI which include the link layer header through the transport layer header. Additionally, we need to consider the headers associated with encapsulation as well as protocol extensions such as header options, extension headers, and network service headers (NSH). Note that this limit only applies to protocol headers that an intermediate device can parse, if the protocol layers are encrypted (as in transports over UDP [I-D.transports-over-udp]) they would not be subject to the limit.

This specification only provides a recommendation for a limit on the cumulative size of packet headers; it does not explicitly provide guidance on limiting the number of protocol headers or nested protocol encapsulations. Coarse limits for those can be inferred from a limit on the size of protocol headers.

2 Derivation of the limit

2.1 Limit without extensibility

To derive a limit for the size protocol headers we first consider a representative "simple" packet in network virtualization. Such a packet has two IP headers, a UDP header and an encapsulation header (using Generic UDP Encapsulation for example [I-D.ietf-nvo3-gue]). We assume that there are no extensions or protocol options present and that intermediate nodes may want to parse an encapsulated TCP header (excluding TCP options). IPv6 over IPv6 encapsulation creates the largest packet in this scenario.
The size of protocol headers for such a packet are:

- 14 bytes for Ethernet header
- 40 bytes for outer IPv6 header
- 8  bytes for UDP encapsulation header
- 8  bytes for encapsulation header
- 40 bytes for inner IPv6 header
- 20 bytes for TCP header

So the size of the headers for this representative packet is 130 bytes. The Ethernet header can be rounded up to sixteen for alignment and the last four bytes of the TCP header (containing checksum and urgent pointer) should not be particularly interesting to intermediate nodes. Hence 128 bytes seems like a reasonable minimum limit for simple packets that contain no extensions.

2.2 Limit with extensibility

To account for protocol extensions such as IP options, IPv6 extension headers, extensions within encapsulation protocols, and network service headers we propose that extensibility is allotted 128 bytes which brings the total limit for protocol headers up to 256 bytes.

The motivations for a 256 byte limit are:

* It is a tradeoff between a minimal limit too small to permit any extensibility in protocol headers and a limit too large that it can’t be practically implemented in hardware.

* It is a power of two. This is compatible with sizes of cache lines and memory buffers that are typically powers of two.

* The 256 byte limit implies a limit to the amount of protocol overhead. For instance, with the standard 1500 byte MTU protocol overhead is 17% with the 256 byte limit. For 512 bytes, the next power of two step, the maximum overhead would be 34%.

* The 256 byte limit implies a limit on the number of headers in header chains and the number of nested encapsulations. For instance, the minimum size of most IPv6 extension headers is eight bytes. With a 1500 byte MTU and no header size limit a packet can contain 182 extension headers; with a 256 byte header limit a packet contains up to twenty-five extension headers.
3 Application of the limit

3.1 Limits on extensibility in protocols

As pointed out the header limit applies to all protocol headers in a packet. When a protocol is being designed there is no means to determine how many bytes will be used in other protocol headers (for instance the designer of an encapsulation protocol can not predict the number of bytes used for extension headers in packets). Thus the recommendation for a protocol header or extension header with variable length is that header size should be limited to 128 bytes.

The 128 byte limit is a maximum value, however because of the possibility of other extensions being present in a packet, protocol designers should be judicious in the use of extensions and employ an efficient format of extension data to minimize overhead. An implementation may allow configuration of options and headers in packets to adhere to the cumulative 256 byte limit at runtime.

3.2 Limit on IPv6 Hop-by-hop options

Hop-by-hop options (HBH) [RFC2460] are required to be processed by intermediate nodes (2460bis relaxes this requirement [I-D.ietf-6man-rfc2460bis]). Processing HBH options is therefore not DPI. In order to facilitate processing of HBH options in intermediate nodes we specifically recommend that the HBH option should be limited to 128 bytes in size.

Note that the HBH options must be the first extension header in packet so that if the HBH option is less than 128 bytes in size it should be ensured that the HBH options lie within a 256 byte protocol header limit and can be processed.

3.3 Hardware implementation

Intermediate nodes that perform DPI or process IPv6 HBH options should be able to parse packets with a minimum of 256 bytes of headers. If a packet is received with an accumulative size of packet headers that exceeds its parsing limit, the packet may take a slow path or be processed with degraded functionality.

3.4 Limits on sender

Packet senders should try to minimize the amount of header overhead in a packet to respect the 256 byte limit. This can enforced in the implementation or as at runtime through configuration.
4 Security Considerations

The recommendation of a limit for size of protocol headers should not
impact security.

5 IANA Considerations

There are no IANA considerations in this document.

6 References

6.1 Normative References


6.2 Informative References


[I-D.ietf-nvo3-gue] Herbert, T., Yong, L., and Zia, O., "Generic UNP Encapsulation", draft-ietf-nvo3-gue-01

[I-D.transports-over-udp] Herbert, T., "Transport layer protocols over UDP", draft-herbert-transports-over-udp-00

[I-D.ietf-nvo3-gue] Herbert, T., Yong, L., and Zia, O., "Generic UNP Encapsulation", draft-ietf-nvo3-gue-01

Author’s Address

Tom Herbert
Facebook
Menlo Park, CA
USA

Email: tom@herbertland.com