INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: October 2019

T. Herbert
Quantonium

April 8, 2019

IPv4 Extension Headers and Flow Label
draft-herbert-ipv4-eh-00

Abstract

This specification defines extension headers for IPv4 and a definition of an IPv4 flow label. The goal is to provide a uniform and feasible method of extensibility that is shared between IPv4 and IPv6.

Table of Contents

1 Introduction

This specification defines extension headers for IPv4 as well as an IPv4 flow label. The motivation is to provide an extensible mechanism in IPv4 that is unified with IPv6 and thus facilitates leveraging common protocol and implementation for extensibility between the two versions of the Internet Protocol.

The extension headers defined for IPv6 in [RFC8200], specifically Hop-by-Hop Options, Destination Options, Routing Header, and Fragment Header are permitted for use with IPv4 (note that Authentication Header and Encapsulating Security Payload are already usable with IPv4). Additionally, No Next Header (protocol number 59) is defined to be usable in IPv4 packets.

The IPv4 flow label is similarly derived from the definition of the IPv6 flow label. There is no flow label defined in the IPv4 header [RFC791], however under specific circumstances the sixteen bit Identification field may safely be used as a flow label.

1.1 Motivation

IPv6 is intended to become the standard protocol of the Internet, however it is clear that there is a large segment of users that will be using IPv4 for the foreseeable future. This is particularly true in many enterprises where a business case for transitioning to IPv6 hasn't yet emerged [V6STATE].

In lieu of sun-setting IPv4 and expecting all users to move to IPv6 in some time frame that is unlikely to be met, this specification suggests an alternative which is to improve IPv4. However the nature of these improvements is very specific, the idea is to "backport" useful features of IPv6 into IPv4. Essentially, this makes IPv4 look more like IPv6. The rationale for this is two fold:

1) Users benefit from forward looking features being actively defined and developed for IPv6 without requiring them to transition to IPv6.

2) In making IPv4 look more like IPv6, the work required to complete a future transition to IPv6 at some site may be reduced or simplified.

Various proposals that would use IPv6 extensions are currently being discussed in IETF. These include Segment Routing [SRV6], Compressed Routing Header [CRH], Path MTU Option [MTUOPT], In-situ OAM [IOAM], Service-aware IPv6 Network [SAIN], and Firewall and Service Tickets [FAST]. These proposals leverage the extensibility mechanism of

extension headers defined for IPv6. All of these proposals, in some form, could be of value for use with IPv4. Unfortunately, IPv4 does not have an extensibility mechanism that meets the requirements for supporting them. IP options are quite limited and have long been considered obsolete. There have been proposal for encoding host to network signaling in UDP (e.g. [SPUD], IOAM over encapsulation like Geneve [IOAMGEN]), however these are shown to neither be generic nor robust especially in the case that encapsulated data must be modified in flight.

The proposal contained in this document is to enable IPv4 packets to carry the extension headers in the same manner that IPv6 packets can carry extension headers. In doing so, the various extensions for IPv6 can be used with IPv4 to the benefit of the user. In many cases (such as IOAM and Path MTU option), the extension being defined is protocol agnostic and would be applicable and usable with IPv4 with little or no change. In other cases, such as segment routing, the extension being defined might be IPv6 specific, for example the segment routing header contains a list of IPv6 addresses. With some modification to the extension definition, it is also conceivable that these may work with IPv4. For instance, in the case of segment routing the extension can be adapted for use with IPv4 by defining a routing header format that contains IPv4 addresses instead of IPv6 addresses.

1.2 IPv4 extension headers

IPv4 options were defined in [RFC0791] as the means of extending the IP protocol. IPv4 options have not been successful. Early router implementations, and even those today, either don't process IPv4 options or relegate them to a slow path effectively making them unusable for serious applications. IPv4 options are limited to forty bytes length and, unlike TCP options, no IP options have been defined that are critical to communications. The upshot is that IPv4 options have long not been considered an option for deployment [IPNOOP].

IPv6 took a different approach. Extensibility of IPv6 is provided by extension headers. Optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet [RFC8200]. IPv6 extension headers have had mixed success in deployment in that some intermediate devices have trouble processing them [RFC7872], however there are several active proposals in IETF that would make use of them (e.g. [FAST], [MTUOPT], [IOAM], [SRV6EH]).

Using extension headers with IPv4 is logically straightforward. The IPv4 Protocol field is effectively re-designated to be a Next Header field with the same meaning and semantics as the IPv6 Next Header field. In this manner, an IPv4 packet can contain IPv6 extension

headers that are recast as IPv4 extension headers. These include Hop-by-Hop Options, Routing Header, Fragment, Destination Options, Authentication, and Encapsulating Security Payload. In cases where an extension header contains IPv6 specific information, the extension header can be adapted for use with IPv4. For instance, a Routing Header carrying IPv6 addresses to visit could be adapted to carry IPv4 addresses.

1.3 The IPv4 flow label

IPv6 [RFC8200] introduced the concept of a flow label that has proven quite convenient to perform flow classification, such as that needed by Equal-Cost Multipath (ECMP). The base IPv4 header does not have reserved bits that could be allocated as a flow label, however the sixteen bit Identification field can be used as a flow label in atomic datagrams [RFC6864].

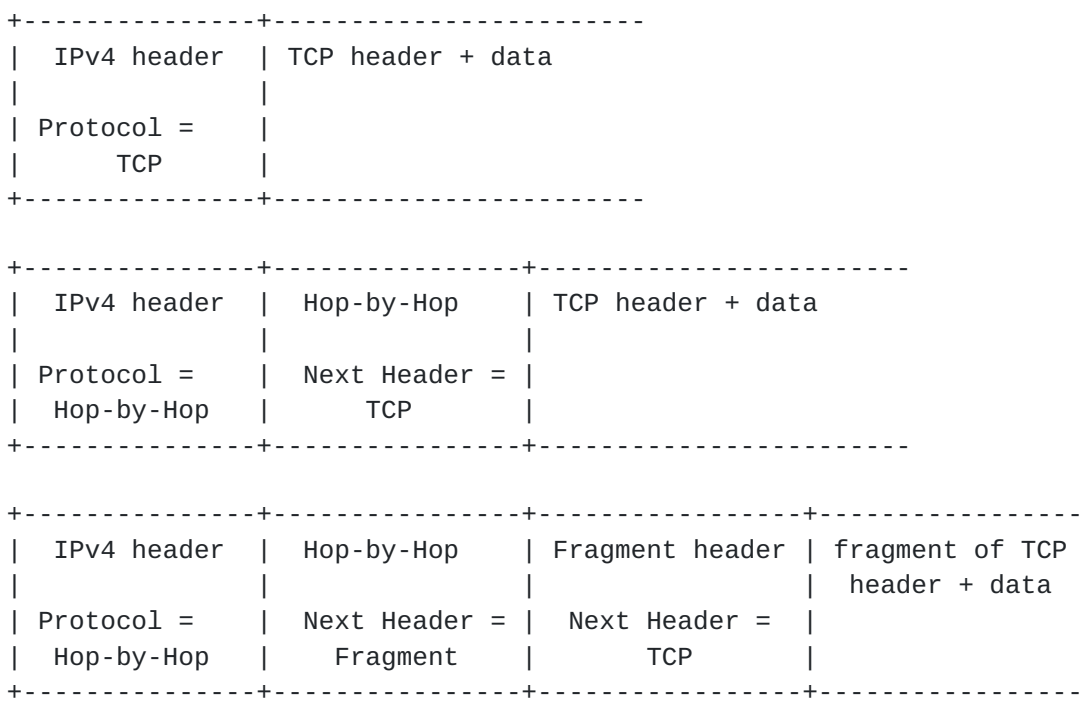
The IPv4 flow label will be most useful in scenarios for which the existing mechanisms used to classify IPv4 packets, such as parsing transport layer headers to extract port information, aren't available. Defining an IPv4 flow label is another instance of back porting a beneficial feature from IPv6 and further unifying the two protocols.

2 IPv4 extension headers

IPv4 extension headers are optional internet-layer information encoded in separate headers that may be placed between the IPv4 header and the upper-layer header in a packet. IPv4 extension headers are based on IPv6 extension headers and share the same basic properties and semantics [RFC8200].

Extension headers are numbered from IANA IP Protocol Numbers [IANA-PN], the same values are used for IPv4 and IPv6. When processing a sequence of Next Header values in a packet, the first one that is not an extension header [IANA-EH] indicates that the next item in the packet is the corresponding upper-layer header. A special "No Next Header" value is used if there is no upper-layer header.

As illustrated in these examples, an IPv4 packet MAY carry zero, one, or more extension headers, each identified by Protocol field of the IPv4 header or the Next Header field of a preceding extension header:



2.1 Requirements

2.1.1 General requirements

IPv4 extension headers normatively assume the requirements of IPv6 extension headers as defined in [RFC8200] section 4, with the following modifications:

* References to the IPv6 header are replaced by references to the IPv4 header.

* ICMP errors sent in the course of processing extension headers use ICMPv4 instead of ICMPv6.

* The IPv4 header Protocol field assumes the same role and semantics with respect to extension headers as the IPv6 Next Header field.

* The Hop-by-Hop Options header is used to carry optional information that MAY be examined and processed by any node along a packet's delivery path.

* If a legacy IPv4 destination node, one that does not support IPv4 extension headers, receives a packet with extension headers then the packet will be processed as having an unknown protocol. It is expected that the packet will be discarded and an ICMP error may be generated.

* Extension headers or options that carry IPv6 specific data or are otherwise specific to IPv6 MUST NOT be used with IPv4 (Segment Routing [SRV6EH] for example). IPv4 variants of these might be defined if achieving the same functionality in IPv4 is desirable.

* References to the Payload Length, for instance in reassembly procedures, are reinterpreted as being the computed IPv4 payload length (i.e. IPv4 Total Length minus the length of the IPv4 header).

2.1.2 Fragmentation and reassembly requirements

The following are modifications to fragmentation and reassembly requirements:

* References to setting the Payload Length field in the IPv6 header are interpreted to be setting the Total Length in the IPv4 header taking into account the IPv4 header length.

* When creating or modifying IPv4 headers in packets, the IPv4 header checksum MUST be set correctly.

* Different fragment packets MAY contain different IPv4 options. In the reassembled packet, the IP options are taken from the first fragment packet (the one with offset of zero).

* Different fragment packets MAY contain different extension headers preceding the fragment header. In the reassembled packet, the extension headers preceding the fragment header are taken from the first fragment packet (the one with offset of zero).

* If the length and offset of a fragment are such that the Total Length of the packet reassembled from that fragment would exceed 65,535 octets, then that fragment must be discarded and an ICMP Parameter Problem, Code 0, message should be sent to the source of the fragment, pointing to the Fragment Offset field of the fragment packet.

2.2 Interaction with standard IPv4 mechanisms

IPv4 extension headers may be used concurrently with IPv4 mechanisms such as IPv4 options and IPv4 fragmentation. This section discusses the interactions.

2.2.1 IPv4 options and IPv4 extension headers

An IPv4 packet MAY contain both IPv4 options and extension headers. IPv4 options are completely independent of IPv4 extension headers. IPv4 options MUST be processed before processing any extension headers per normal requirements of processing the IP header before the IP payload.

2.2.2 IPv4 fragmentation and IPv4 extension headers

An IPv4 packet MAY be fragmented both by using a Fragment extension header as well as by standard IPv4 fragmentation. The Fragment header can only be set at the source, however intermediate devices can fragment packets using standard IPv4 fragmentation. Standard IPv4 fragmentation at a source node MUST be done only after any extension headers are set in a packet or the packet was fragmented using the Fragment header. Specifically, fragmentation using the extension header MUST NOT be done on packet fragments created by standard IPv4 fragmentation. However, a packet fragment that contains a Fragment header MAY itself be fragmented by standard IPv4 fragmentation. There is no correlation between normal IPv4 fragmentation and the IPv4 Fragment header, the identifier space for each are unrelated and reassembly procedures are independent.

At a destination, if a received packet was fragmented by standard IPv4 fragmentation, it MUST be reassembled before processing any IPv4 extension headers. This requirement ensures that standard IPv4 reassembly is done before reassembly for the Fragment header.

If an IPv4 packet containing Hop-by-Hop options is fragmented using standard IPv4 fragmentation, the Hop-by-Hop Options are not set in each of the packet fragments. An intermediate node MAY process the Hop-by-Hop options in the first fragment if the complete Hop-by-Hop extension header is contained within the fragment. If the Fragment header is used with IPv4 then the DF bit (Don't Fragment) bit SHOULD be set and Path MTU discovery mechanisms SHOULD be used.

2.2.3 Atomic datagram recommendation

It is RECOMMENDED to only use IPv4 extensions in atomic datagrams. Atomic datagrams [RFC6834] are IPv4 packets for which the Don't Fragment bit set, More Fragment bit is not set, and Fragment Offset is zero. In this case the packet will not be subject to IPv4 fragmentation, the Fragment header can alternatively be used for fragmentation.

3 The IPv4 flow label

The Identification field of the IPv4 header is re-purposed to be the IPv4 flow label in atomic datagrams. As stated in [RFC6864]:

> ">> Originating sources MAY set the IPv4 ID field of atomic datagrams to any value."

This specification allows the IPv4 ID to be used as a flow label in atomic datagrams where (DF==1)&&(MF==0)&&(frag_offset==0).

3.1 Sender requirements

An origin host MAY set the IPv4 Identification field as a flow label in atomic datagram packets. The IPv4 flow label is set following the same procedures for setting the IPv6 flow label as described in [RFC6437], with the following modifications:

* The Identification field MUST only be used as a flow label in atomic datagrams. That is Don't Fragment (DF) bit MUST be set, More Fragment (MF) bit MUST NOT be set, and Fragment Offset MUST be zero.

* If the IPv4 Identification field is not used as a flow label in atomic fragments, the Identification field MUST be set to zero.

* Only stateless flow labels can be set.

* The value to set, e.g. from a hash computation over packet headers, is truncated to sixteen bits (the size of the Identification field).

* Intermediate nodes MUST NOT set the Identification field in atomic datagrams.

3.2 Receiver requirements

Receivers, including intermediate hosts, MAY process a non-zero Identification field in the IPv4 header of atomic datagrams as being a flow label. The IPv4 flow label for instance can be used as input to ECMP as described in [RFC6438].

If the Identification field is zero or the packet is not an atomic datagram (either the More Fragment bit is set, the Don't Fragment bit is not set, or Fragment Offset is non-zero) then the Identification field MUST NOT be considered as a flow label.

4 Deployability

If a legacy host device receives an IPv4 packet with IPv4 extension headers, the packet will be treated as having an unknown protocol and should dropped. Intermediate devices might also see packets with a protocol unknown to them and will forward the packet inasmuch as they would forward any packet with an unknown protocol.

In the Internet, it is well known that there are some intermediate nodes that will drop packets with protocols that are unknown to them (firewalls would commonly to this for instance). Therefore, it is unlikely that packets with IPv4 extension headers can be ubiquitously deployed over the Internet. A workaround to this might be to encapsulate extension headers in UDP [EHUDPENCAP].

In a limited domain [LIMDOM], an operator would have control over intermediate nodes and could ensure that at a minimum they properly forward packets with IPv4 extension headers. Routers in a limited domain can be updated to process IPv4 Hop-by-Hop Options or Routing headers to provide the functionality of features like IOAM and Segment Routing in IPv4. Similarly, they could be updated to support the IPv4 flow label to provide flow based ECMP in the same manner that the IPv6 flow label is used for ECMP [RFC6438].

5 Security Considerations

This specification enables use of IPv6 extension headers in IPv4. Related security mechanisms of IPv6 extension headers can be applied for use with IPv4 extension headers.

The IPv4 flow label has similar security properties as the IPv6 flow label. If the security intent of the sender is to prevent intermediate nodes in the network from classifying its traffic into flows then the IPv4 flow label SHOULD NOT be used.

6 IANA Considerations

IANA is requested to change the descriptions of IPv6 extension headers and No Next Header protocol numbers to reflect that they are not IPv4 specific.

In the Assigned Internet Protocol Numbers Registry, the modified protocols descriptions are:

Decimal	Keyword	Protocol	IPv6 Extension header	Reference
0	HOPOPT	Hop-by-Hop Option		[RFC8200][RFCXXXX]
43	Route	Routing Header		[Steve_Deering]
44	Frag	Fragment Header		[Steve_Deering]
59	NoNxt	No Next Header		[RFC8200][RFCXXXX]
60	Opts	Destination Options		[RFC8200][RFCXXXX]

IANA is requested to update "Internet Protocol Version 4 (IPv4) Parameters" to include sections for "IPv6 Extension Header Types", "Destination Options and "Hop-by-Hop Options", and "Routing Types". These are based on the similarly named sections in "Internet Protocol Version 6 (IPv6) Parameters" with appropriate modifications for IPv4.

7 References

7.1 Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, DOI 10.17487/RFC6864, February 2013, <https://www.rfc-editor.org/info/rfc6864>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

7.2 Informative References

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[RFC7605] Touch, J., "Recommendations on Using Assigned Transport Port Numbers", BCP 165, RFC 7605, DOI 10.17487/RFC7605, August 2015, <https://www.rfc-editor.org/info/rfc7605>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[V6STATE] B. Kuerbis and M. Mueller, Internet Governance Project, "The Hidden Standards War: Economic Factors Affecting IPv6 Deployment", February, 2019.

[SRV6EH] C. Filsfils, Ed., S. Previdi, J. Leddy, S. Matsushima, D. Voyer, Ed., "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16

[CRH] Bonica, R., So, N., Xu, F., Chen, G., Zhu, Y., Yang, G., Zhou, Y., "The IPv6 Compressed Routing Header (CRH)" draft-bonica-6man-comp-rtg-hdr-03

[MTUOPT] Hinden, R. and Fairhurst, G., "IPv6 Minimum Path MTU Hop-by-Hop Option", draft-hinden-6man-mtu-option-00

[IOAM] F. Brockners, S. Bhandari, V. Govindan, C. Pignataro, H. Gredler, J. Leddy, S. Youell, T. Mizrahi, D. Mozes, P. Lapukhov, R. Chang, "Encapsulations for In-situ OAM Data"

draft-brockners-inband-oam-transport-05

[SAIN] Li, Z. and Peng, S., "Service-aware IPv6 Network", draft-li-6man-service-aware-ipv6-network-00

[FAST] Herbert, T., "Firewall and Service Tickets", draft-herbert-fast-03

[SPUD] Hildebrbrand, J. and Trammell, B., Substrate Protocol for User Datagrams (SPUD) Prototype, draft-hildebrand-spud-prototype-03

[IOAMGEN] Brockners, F. et al., "Geneve encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-geneve-01

[IPNOOP] Rodrigo Fonseca, George Manning Porter, Randy H. Katz, Scott Shenker and Ion Stoica, "IP Options are not an option", <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-24.html>

[IANA-PN] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers>.

[IANA-EH] IANA, "IPv6 Extension Header Types", <https://www.iana.org/assignments/ipv6-parameters>.

[LIMDOM] Carpenter, B., and Liu, B., "Limited Domains and Internet Protocols", draft-carpenter-limited-domains-06

Author's Address

Tom Herbert
Quantonium
Santa Clara, CA

USA

Email: tom@quantonium.net