Link Adaptation for IPv6-in-(foo)-in-IPv4 Tunnels
draft-templin-linkadapt-05.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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/ietf/1id-abstracts.txt.

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

This Internet-Draft will expire on November 3, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

IPv6-in-(foo)-in-IPv4 tunnels support a minimum Maximum Transmission Unit (MTU) of 1280 bytes for IPv6 via static prearrangements and/or dynamic MTU determination based on ICMPv4 messages, but these methods have known operational limitations. This document specifies link adaptation mechanisms for IPv6-in-(foo)-in-IPv4 tunnels that present an assured MTU to the IPv6 layer using tunnel endpoint-based segmentation/reassembly and dynamic segment size probing.
Table of Contents

1. Introduction ............................................. 3
2. Terminology .............................................. 3
3. Link Adaptation for IPv6-in-(foo)-in-IPv4 Tunnels ................. 4
   3.1. Layering ............................................ 4
   3.2. Initial Negotiation Phase ............................. 4
   3.3. Tunnel MTU and MRU .................................. 5
   3.4. Ingress Tunnel Endpoint Specification .................. 5
       3.4.1. Segmentation and Encapsulation .................. 5
       3.4.2. IPv4 Fragmentation and Setting the DF Bit ......... 7
       3.4.3. Probing ........................................ 8
       3.4.4. Processing Errors ............................... 9
   3.5. Egress Tunnel Endpoint Specification ................... 10
       3.5.1. Decapsulation and Reassembly ..................... 10
       3.5.2. Sending Errors .................................. 10
       3.5.3. Sending Probe Replies ............................ 11
       3.5.4. Active Reassembly Buffer Management ............... 11
4. IANA Considerations ........................................ 11
5. Security Considerations ..................................... 12
6. Acknowledgments .......................................... 12
7. Appendix A: Additional Considerations .......................... 12
8. Appendix B: Changes ........................................ 13
9. References ................................................. 14
   9.1. Normative References ................................ 14
   9.2. Informative References ............................... 14
Author’s Address ............................................. 15
Intellectual Property and Copyright Statements ..................... 17
1. Introduction

IPv6-in-(foo)-in-IPv4 tunnels may span multiple IPv4 network hops yet are seen by IPv6 as ordinary links that must support the minimum IPv6 Maximum Transmission Unit (MTU) of 1280 bytes ([RFC2460], Section 5). Common IPv6-in-(foo)-in-IPv4 tunneling mechanisms meet this requirement through conservative static prearrangements at the expense of degraded performance over some paths due to excessive IPv4 network-based fragmentation and/or missed opportunities to discover larger MTUs. Optional dynamic MTU determination methods [RFC1191] are also available, but may not provide adequate robustness.

This document specifies link adaptation mechanisms for IPv6-in-(foo)-in-IPv4 tunnels that present an assured MTU to the IPv6 layer. It uses tunnel endpoint-based segmentation/reassembly and dynamic segment size probing with authenticated probe feedback. Thus, it provides greater robustness and efficiency by avoiding IPv4 network-based fragmentation and dependence on ICMPv4 feedback from IPv4 network middleboxes.

2. Terminology

The following terms are defined within the scope of this document:

Upper Layer Payload (ULP)
a whole IPv6 packet, or a fragment packet created by IPv6 fragmentation.

Ingress Tunnel Endpoint (ITE)
the tunnel interface endpoint that accepts ULPs from the IP layer and segments/packetizes them for transmission into a tunnel.

Egress Tunnel Endpoint (ETE)
the tunnel interface endpoint that receives packets from a tunnel and de-packetizes/reassembles them into ULPs for delivery to the IP layer.

IP Layer
the layer above the tunnel interface, i.e., the IPv6 layer.
The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

3. Link Adaptation for IPv6-in-(foo)-in-IPv4 Tunnels

The following subsections specify link adaptation mechanisms for IPv6-in-(foo)-in-IPv4 tunnels with properties similar to the link adaptation mechanisms defined for AAL5 [RFC2684] and IEEE 802.11 [WLAN]:

3.1. Layering

IPv6-in-(foo)-in-IPv4 tunnel endpoints operate at a logical midpoint between the IPv6 and IPv4 protocol modules. From the viewpoint of IPv6, the tunnel appears as an ordinary network interface module that delivers whole IPv6 packets and IPv6 fragment packets as ULPs to and from an underlying link. From the viewpoint of IPv4, the tunnel appears as a packetization layer protocol that segments and reassembles ULPs.

This document refers to the IPv6 layer as the "IP Layer" (i.e., layer 3) and any sublayers that occur within the tunnel interface (i.e., any (foo) layers and including the upper portion of the IPv4 layer itself) as the "Sub-IP layer". Note that IPv4 is also viewed as the Layer 2 protocol from the perspective of the tunnel, so the Sub-IP layer begins below the IP layer and extends into Layer 2. Note also that (foo) may entail multiple nested sublayers or may even be NULL, i.e., in the case of IPv6-in-IPv4 tunnels.

3.2. Initial Negotiation Phase

IPv6-in-(foo)-in-IPv4 tunnel endpoints MUST first determine that the link adaptation mechanisms are implemented by both the ITE and ETE through an initial negotiation phase specified outside the scope of this document. ITEs/ETEs for which one or both ends of the tunnel do not implement the scheme MUST use the default MTU assurance mechanisms specified for the particular IPv6-in-(foo)-in-IPv4 tunneling mechanism, and do not implement any other aspects of this
3.3. Tunnel MTU and MRU

ITEs MUST configure a minimum IPv6 link MTU of 1280 bytes for all flows and SHOULD provide a configuration knob to set larger values. A nominal per-flow MTU of 9180 bytes (i.e., the same as defined in [RFC1626]) is RECOMMENDED, since it is large enough to accommodate frame sizes as large as Gigabit Ethernet Jumbo Frames [GIGE]. ITEs MAY set still larger MTU values, but are advised that this may lead to excessive packet loss and ICMPv6 "packet too big" messages.

ETEs MUST configure a minimum per-flow Sub-IP layer reassembly buffer size (i.e., a minimum Sub-IP layer Maximum Receive Unit (MRU)) of 1280 bytes, and SHOULD configure an MRU of 9180 bytes or larger to accommodate the recommended nominal MTU for ITEs. A maximum MRU of 11454 bytes is RECOMMENDED, since 11454 bytes is the maximum packet size for which a 32-bit CRC can provide Ethernet-quality bit error detection [JAIN][AARNET]. ETEs MAY set still larger MRU values, but are advised that larger values may lead to unacceptable levels of undetected errors unless all physical segments in the path provide assured error-free delivery for larger packets.

3.4. Ingress Tunnel Endpoint Specification

The following Tunnel Endpoint Specification

3.4.1. Segmentation and Encapsulation

ITEs maintain a per-flow MTU and per-flow segment size ("SEGSIZE") for the purpose of segmenting ULPs that are too large to traverse the tunnel. It is RECOMMENDED that ITEs configure an initial per-flow SEGSIZE such that (SEGSIZE + length((foo) headers) + length(IPv4 header)) yields an IPv4 datagram size between 256-576 bytes (since 256 bytes can safely accommodate the recommended nominal MTU (see below), and since IPv4 nodes are only required to accept datagrams of up to 576 bytes [RFC0791]). Since most IPv4 links in the Internet configure still larger MTUs [RFC3150][RFC3819], and since IPv4 nodes should accept packets as large as the underlying link MTU [RFC1122], ITEs MAY use a still larger initial per-flow SEGSIZE if there is assurance that it would not cause gratuitous IPv4 fragmentation and/or overrun the IPv4 reassembly buffer. ITEs probe the path to maintain SEGSIZE and/or discover larger SEGSIZEs during the lifetime of a flow (see: Section 3.4.3).

ITEs split each ULP they send into a tunnel into chains of segments for packetization and presentation to the IPv4 layer. For ULPs that will span multiple segments, the ITE first uses the 2’s compliment
Fletcher-32 checksum [STONE][RFC3385] to calculate a checksum across the entire ULP, then appends the A and B results as a trailing 32-bit checksum at the end of the ULP. For ULPs that fit within a single segment, the ITE omits the trailing checksum.

The ITE next splits the ULP into a chain of consecutive segments that MUST be created as contiguous and non-overlapping, i.e., the final byte of the (i)th segment MUST be the byte that immediately precedes the first byte of the (i+1)th segment. Non-final segments in the chain MUST be identical in length and no larger than SEGSIZE bytes; the final segment MAY be of different length. The ITE encapsulates each segment in Sub-IP layer headers (including any (foo) headers and an IPv4 header) to form a chain of IPv4 packets; each packet in the chain MUST include Sub-IP layer encapsulation headers of identical length. The ITE sets the DF bit in the IPv4 header according to the specification in Section 3.4.2, and encodes the following information in the 16-bit IPv4 "Identification" field of each segment:

0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ULPID      |  SEGID  |P|A|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
IPv4 Identification Field

ULPID:  8 bits
An identifying value assigned by the ITE to aid the ETE in reassembling the segments of a ULP.

SEGID:  6 bits
A value that identifies a specific segment within a ULP.

P:  1 bit
Probe flag; 0 = Ordinary Segment, 1 = Probe Segment.

A:  1 bit
Additional Segments flag; 0 = Last Segment, 1 = Additional Segments.

The ITE encodes an identical value in the "ULPID" field (bits 0 - 7 of the IPv4 Identification field) of each IPv4 packet in a chain to identify the segments of a specific ULP; it encodes different ULPID values in IPv4 packets that encapsulate segments of different ULPs. The ITE also encodes an increasing Segment ID value between 0 - 62 in
the "SEGID" field (bits 8 - 13 of the IPv4 Identification field) of consecutive packets in a chain, i.e., it encodes the value '0' in the first packet, encodes the value '1' in the second packet, etc.

The ITE then sets the "Additional Segments - A" bit (bit 15 of the IPv4 Identification field) in each packet in the chain except the final one to indicate that additional segments follow. Finally, it delivers each packet in the chain to the link layer (i.e., the IPv4 layer) in increasing SEGID order, i.e., SEGID 0 first, followed by SEGID 1, etc., up to the final packet. The IPv4 layer SHOULD NOT reorder the packets in a chain, but rather SHOULD deliver them to the underlying link in the order in which the tunnel interface produced them.

Note that IPv4 fragmentation in the network could theoretically result in silent packet loss along certain paths even for packets with the smallest recommended initial SEGSIZE (see: Section 3.4.2). As such, a robust ITE implementation could reduce its IPv4 packet sizes to as small as 68 bytes if it suspects that larger packets are disappearing into a fragmentation-related black hole, but such small packets might not satisfy the nominal tunnel MTU of 9180 bytes. ITEs SHOULD therefore return locally-generated IPv6 "packet too big" messages for IPv6 packets that cannot be segmented and encapsulated within current IPv4 packet size and chain length limitations for the tunnel.

3.4.2. IPv4 Fragmentation and Setting the DF Bit

When an ITE segments a ULP (see: Section 3.4.1), it can optionally set or clear the "Don't Fragment - DF" bit in the encapsulating IPv4 headers of packets in the chain. If the DF bit is cleared, gratuitous network-based IPv4 fragmentation could result in well-known operational issues [FRAG] [I-D.heffner-frag-harmful]. Also, some middleboxes (such as IPv4 NATs and firewalls) may only be capable of passing the first fragment of a multi-fragment IPv4 datagram, and large multi-fragment datagrams could result in IPv4 reassembly buffer overruns. Finally, the minimum IPv4 MTU is only 68 bytes (i.e., the size required to encapsulate a maximum-length (60 byte) IPv4 header and a minimum-length (8 byte) fragment [RFC0791]) such that a limited amount of IPv4 fragmentation may occur in the network even for relatively small packets.

Nonetheless, clearing the DF bit can in some circumstances increase the packet delivery ratio when setting the DF bit would otherwise result in excessive packet loss due to temporal link MTU restrictions. In view of the above considerations, the ITE:
o SHOULD set the DF bit in probe packets (see: Section 3.4.3) larger than 576 bytes.

o SHOULD set the DF bit in all packets larger than 576 bytes if it will not perform active probing (see: Section 3.4.3).

o MAY clear the DF bit in any packets larger than 576 bytes if it will perform active probing.

o MAY clear the DF bit in any packets of 576 bytes or smaller.

3.4.3. Probing

To increase efficiency and avoid excessive packet chain lengths, ITEs SHOULD probe the path periodically to increase a flow’s SEGSIZE to larger values. ITEs probe a candidate SEGSIZE value 'N' by setting the "Probe Segment - P" bit (bit 14 of the IPv4 Identification field) in packets that encapsulate a probe segment of size N. For probe segments that contain valid data for reassembly as part of a packet chain, the ITE sets the appropriate SEGID value in the IPv4 packet header as for ordinary segmentation. For probe segments that are to be discarded by the ETE, the ITE sets the value 63 in the SEGID field.

When the ITE sends a probe packet, it marks the probe as "pending" for a period of 'MaxProbeDelay' msec (i.e., a per-flow round-trip time estimate for the tunnel) and caches the probe packet’s IPv4 destination, length and identification field values, as well as the IPv6 flow label value [RFC3697]. If the ITE receives a valid Node Information Query reply (NI Reply) [RFC4620] from the ETE (see: Section 3.5.3) before the probe period expires, it marks the probe as successful; otherwise, it marks the probe as failed. A valid NI Reply MUST have:

o the Type, Code, Qtype and Flags fields set as specified for a NOOP reply in ([RFC4620], Section 6.1), and

o the IPv4 length of the probe packet matches bits 0-15 of the Nonce field, and

o the IPv4 identification of the probe packet matches bits 16-31 of the Nonce field, and

o the IPv6 flow label value matches bits 32-51 of the Nonce field

Following a successful probe, but before advancing SEGSIZE to N, the ITE SHOULD enter a brief verification phase during which it sends additional probe segments to detect asymmetric multipath MTU.
restrictions and/or route fluctuations. Thereafter, the ITE SHOULD re-probe periodically to confirm that packets with up to SEGSIZE byte segments are still reaching the ETE.

After probing the path to discover a new SEGSIZE, the ITE may elect to set or clear the DF bit in subsequent non-probe packets (see: Section 3.4.2). For example, the ITE may elect to clear the DF bit to maintain an optimal packet delivery ratio across temporal link MTU restrictions (e.g., due to dynamic rerouting of flows, etc.) while it may elect to set the DF bit to avoid all IPv4 fragmentation in the network.

ITEs that elect to clear the DF bit in non-probe packets SHOULD engage in "active probing" to periodically confirm SEGSIZE "frequently enough" such that cyclical misassociations and possible data corruptions at the ETE do not occur [I-D.heffner-frag-harmful] if a flow begins to fragment. ITEs that elect to set the DF bit in non-probe packets SHOULD carefully consider any ICMPv4 "fragmentation needed" messages that arrive (see: Section 3.4.4) but are advised that packet delivery ratios may suffer when the flow transmission rate is high and/or the path round trip time is large.

### 3.4.4. Processing Errors

ITEs may receive ICMPv4 "fragmentation needed" error messages from middleboxes inside a tunnel, but are advised to consider them as "soft errors". Implementers are advised to consult [RFC1191][RFC2923][RFC4821] for operational recommendations on processing ICMPv4 "fragmentation needed" messages.

ITEs may receive encapsulated ICMPv6 "packet too big" messages [RFC1981] from an ETE at the far end of a tunnel (see: Section 3.5.2). The ITE SHOULd cache the MTU value encoded in the "packet too big" message as the new MTU for the flow, and relay the ICMPv6 message back to the original source.

ITEs may receive encapsulated ICMPv6 "parameter problem" messages with code "reassembly/Checksum error" [RFC4443] from an ETE at the far end of the tunnel (see: Section 3.5.2). This may indicate an isolated packet splicing error at the ETE, or packet loss due to temporal network conditions such as congestion, MTU restrictions, link errors, signal intermittence, etc. If the ITE receives persistent reassembly/Checksum errors from an ETE, it SHOULD take adaptive measures, e.g., reduce the SEGSIZE for the flow, rate-limit the packets it sends into the tunnel, etc. Since each reassembly/Checksum error corresponds to a dropped packet, the ITE SHOULD relay the messages back to the original source (subject to rate limiting).
3.5. Egress Tunnel Endpoint Specification

The following subsections specify mechanisms implemented by the ETE:

3.5.1. Decapsulation and Reassembly

The IPv4 length, ULPID, SEGID and A fields in the IPv4 packets in a chain (along with the IPv6 flow label [RFC3697]) provide sufficient information for the ETE to reassemble an original ULP with protection for packet reordering in the network. ETEs MUST configure per-flow reassembly buffers of at least 1280 bytes and SHOULD configure reassembly buffers of 9180 bytes or larger to accommodate the nominal tunnel MTU (see: Section 3.2). Note that these reassembly buffers occur at the Sub-IP layer and are thus distinct from the IPv4 and IPv6 reassembly caches.

ETEs use per-flow reassembly buffers to concatenate the segments received in packet chains for a particular ULPID in increasing SEGID order (i.e., SEGID 0, followed by SEGID 1, etc.) even if the packets were re-ordered by the network. When all segments for a particular ULPID have been concatenated into the reassembly buffer, the ETE uses 2’s complement Fletcher-32 to verify the checksum if one was included (see: Section 3.4.1). The ETE discards the Sub-IP layer encapsulation headers and trailing checksum, and delivers correctly-reassembled ULPs to the IP layer (i.e., IPv6). It discards incomplete ULPs and ULPs with incorrect checksums, and sends an appropriate error message as specified in Section 3.5.2.

3.5.2. Sending Errors

If the ETE receives a packet chain that would overflow the reassembly buffer, it discards the chain and sends an ICMPv6 "packet too big" message [RFC1981] back to the IPv6 source via the reverse tunnel back to the ITE. The ETE includes in the message body up to 1280 bytes beginning with the upper layer packet headers (IPv6 and above) and the contents of the reassembly buffer beyond the upper layer packet headers; it encodes the size of the reassembly buffer in the MTU value.

If the ETE receives at least one segment, but one or more segments are lost and/or checksum verification fails, it SHOULD send an ICMPv6 "parameter problem" message with code "reassembly/checksum error" [RFC4443] back to the IPv6 source via the reverse tunnel back to the ITE. The ETE includes in the message body up to 1280 bytes beginning with the upper layer packet headers (IPv6 and above) and contents of the reassembly buffer beyond the upper layer packet headers, and sets the pointer to either the beginning of the first missing segment or the beginning of the 4 byte checksum field (if no segments were
After sending the error, the ITE discards the packet-in-error, i.e., it does not deliver the packet as an ULP to the IP layer.

### 3.5.3. Sending Probe Replies

If the ETE receives a segment used for probing (i.e., an IPv4 packet in the chain with the ‘P’ flag set), it sends a Node Information Query reply (NI Reply) \[RFC4620\] message back to the ITE. The ETE MUST construct the NI Reply as follows:

- the Type, Code, Qtype and Flags fields set as specified for a NOOP reply in \([\text{RFC4620}, \text{Section 6.1}]\), and
- the IPv4 length of the probe packet encoded in bits 0-15 of the Nonce field, and
- the IPv4 identification of the probe packet encoded in bits 16-31 of the Nonce field, and
- the IPv6 flow label value encoded in bits 32-51 of the Nonce field

If the IPv4 packet containing the probe segment encodes the value 63 in the SEGID field, the ETE discards the segment; otherwise, it includes the segment as part of the normal reassembly procedure described above.

### 3.5.4. Active Reassembly Buffer Management

The ETE MUST actively manage reassembly buffers and discard as early as possible any reassemblies that are not likely to complete due to, e.g., loss of one or more packets in the chain, gross reordering of packets in the network, etc. In particular, the ETE must discard partial reassemblies before the 8-bit ULPID encoded by the ITE wraps. The ETE therefore must augment the classical timer-driven reassembly buffer management strategy with an event-driven strategy.

### 4. IANA Considerations

The IANA is instructed to assign a code type for "reassembly/checksum error" under the ICMPv6 Parameter Problem message type in the "ICMPv6 Type Numbers" registry.
5. Security Considerations

The nonce values in NI Reply messages from ETEs provide spoofing protection against off-path attackers.

6. Acknowledgments

Earlier versions of this work attempted to cite all who made significant contributions over the years, but the list was far too extensive to capture. This work therefore acknowledges the mindshare of many contributors. There are still others to acknowledge whose vital contributions could never be expressed in tangible terms.

7. Appendix A: Additional Considerations

ITEs can use the probing mechanism described in Section 3.3 as a general-purpose method for eliciting acknowledgements from an ETE if improved reliability at the expense of additional overhead is desired.

The equal size restriction for non-final segments and non-overlapping restriction for all segments in packet chains provides a significant simplification for reassembly algorithms [RFC0815].

Use of the link adaptation mechanisms specified in this document may lead to an overall increase in short chains of small packets in the Internet. Network administrators are advised to follow the recommendations in [RFC3150] to minimize packet loss and packet reordering. Also, overly-long packet chains should be avoided if possible due to interactions with Active Queue Management (AQM) in the network.

Since link-layer CRC-32 checks normally occur on each segment in the path, most errors detected during ULP reassembly are due to packet splices and/or errors in the data path between the NIC hardware and the reassembly buffer. The Fletcher-32 checksum algorithm has been shown to provide an effective edge-to-edge error detection capability for such errors [STONE]. The Fletcher-32 checksum is also dissimilar from both CRC-32 and the Internet checksum used by many upper layer protocols, thereby decreasing the likelihood of undetected errors.

Some upper layer packetization protocols (e.g., NFS) may generate fixed payload sizes and rely on the network layer to deliver the payloads either as whole IP packets or as chains of IP fragments. Since NFS performance (and the performance of other upper layer packetization protocols) is sensitive to packet handling overhead,
implementations should periodically attempt to increase the SEGSIZE through probing even if initial probe attempts fail.

8. Appendix B: Changes

(Note to RFC Editor - please remove this section before publishing as an RFC.)

Changes since -04:

- Rearranged sections for clarity.
- Removed setting of IPv4 "Reserved Fragmentation", since ITE/ETE capabilities can be discovered during the initial tunnel negotiation.

Changes since -03:

- Clarified that mechanisms cover IPv6-in-(foo)-in-IPv4; not just IPv6-in-IPv4.
- New terminology for ITE/ETE
- Clarifications to layering model
- Replaced RA with NI Reply as probe response
- Reduced SEGID to 6 bits and increased ULPID to 8 bits
- IPv6 flow label RFC cited

Changes since -01, -02:

- Updated references

Changes since -00:

- Defined new coding of segmentation/reassembly info in the IPv4 Identification field
- Changed "tunneling mechanism" to "tunnel endpoint"
- Clarified text on trailing checksums
- General document cleanup; removed "additional considerations" that no longer apply
9. References

9.1. Normative References


9.2. Informative References


Author’s Address

Fred L. Templin (editor)
Boeing Phantom Works
P.O. Box 3707
Seattle, WA  98124
USA

Email: fred.l.templin@boeing.com