Network Working Group
Request for Comments: 2675
Obsoletes: 2147
Category: Standards Track

D. Borman
Berkeley Software Design
S. Deering
Cisco
R. Hinden
Nokia

August 1999

IPv6 Jumbograms

Status of this Memo

Copyright Notice

Abstract

A "jumbogram" is an IPv6 packet containing a payload longer than 65,535 octets. This document describes the IPv6 Jumbo Payload option, which provides the means of specifying such large payload lengths. It also describes the changes needed to TCP and UDP to make use of jumbograms.

Jumbograms are relevant only to IPv6 nodes that may be attached to links with a link MTU greater than 65,575 octets, and need not be implemented or understood by IPv6 nodes that do not support attachment to links with such large MTUs.

1. Introduction

jumbo (jum'bO),

n., pl. -bos, adj.
-n.
1. a person, animal, or thing very large of its kind.
-adj.
2. very large: the jumbo box of cereal.

[1800-10; orig. uncert.; popularized as the name of a large elephant purchased and exhibited by P.T. Barnum in 1882]

-- www.infoplease.com

The IPv6 header [IPv6] has a 16-bit Payload Length field and, therefore, supports payloads up to 65,535 octets long. This document specifies an IPv6 hop-by-hop option, called the Jumbo Payload option, that carries a 32-bit length field in order to allow transmission of IPv6 packets with payloads between 65,536 and 4,294,967,295 octets in length. Packets with such long payloads are referred to as "jumbograms".

The Jumbo Payload option is relevant only for IPv6 nodes that may be attached to links with a link MTU greater than 65,575 octets (that is, 65,535 + 40, where 40 octets is the size of the IPv6 header). The Jumbo Payload option need not be implemented or understood by IPv6 nodes that do not support attachment to links with MTU greater than 65,575.

On links with configurable MTUs, the MTU must not be configured to a value greater than 65,575 octets if there are nodes attached to that link that do not support the Jumbo Payload option and it can not be guaranteed that the Jumbo Payload option will not be sent to those nodes.

The UDP header [UDP] has a 16-bit Length field which prevents it from making use of jumbograms, and though the TCP header [TCP] does not have a Length field, both the TCP MSS option and the TCP Urgent field are constrained to 16 bits. This document specifies some simple enhancements to TCP and UDP to enable them to make use of jumbograms. An implementation of TCP or UDP on an IPv6 node that supports the Jumbo Payload option must include the enhancements specified here.

Note: The 16 bit checksum used by UDP and TCP becomes less accurate as the length of the data being checksummed is increased. Application designers may want to take this into consideration.

1.1 Document History

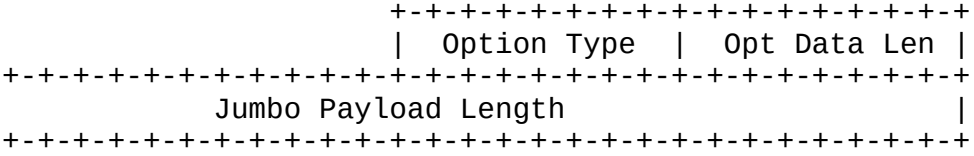
This document merges and updates material that was previously published in two separate documents:

- The specification of the Jumbo Payload option previously appeared as part of the IPv6 specification in RFC 1883. RFC 1883 has been superseded by RFC 2460, which no longer includes specification of the Jumbo Payload option.

- The specification of TCP and UDP enhancements to support jumbograms previously appeared as RFC 2147. RFC 2147 is obsoleted by this document.

2. Format of the Jumbo Payload Option

The Jumbo Payload option is carried in an IPv6 Hop-by-Hop Options header, immediately following the IPv6 header. This option has an alignment requirement of 4n + 2. (See [IPv6, Section 4.2] for discussion of option alignment.) The option has the following format:



Option Type	8-bit value C2 (hexadecimal).
Opt Data Len	8-bit value 4.
Jumbo Payload Length	32-bit unsigned integer. Length of the IPv6 packet in octets, excluding the IPv6 header but including the Hop-by-Hop Options header and any other extension headers present. Must be greater than 65,535.

3. Usage of the Jumbo Payload Option

The Payload Length field in the IPv6 header must be set to zero in every packet that carries the Jumbo Payload option.

If a node that understands the Jumbo Payload option receives a packet whose IPv6 header carries a Payload Length of zero and a Next Header value of zero (meaning that a Hop-by-Hop Options header follows), and whose link-layer framing indicates the presence of octets beyond the IPv6 header, the node must proceed to process the Hop-by-Hop Options header in order to determine the actual length of the payload from the Jumbo Payload option.

The Jumbo Payload option must not be used in a packet that carries a Fragment header.

Higher-layer protocols that use the IPv6 Payload Length field to compute the value of the Upper-Layer Packet Length field in the checksum pseudo-header described in [IPv6, Section 8.1] must instead use the Jumbo Payload Length field for that computation, for packets that carry the Jumbo Payload option.

Nodes that understand the Jumbo Payload option are required to detect a number of possible format errors, and if the erroneous packet was not destined to a multicast address, report the error by sending an ICMP Parameter Problem message [ICMPv6] to the packet's source. The following list of errors specifies the values to be used in the Code and Pointer fields of the Parameter Problem message:

error: IPv6 Payload Length = 0 and
IPv6 Next Header = Hop-by-Hop Options and
Jumbo Payload option not present

Code: 0
Pointer: high-order octet of the IPv6 Payload Length

error: IPv6 Payload Length != 0 and
Jumbo Payload option present

Code: 0
Pointer: Option Type field of the Jumbo Payload option

error: Jumbo Payload option present and
Jumbo Payload Length < 65,536

Code: 0
Pointer: high-order octet of the Jumbo Payload Length

error: Jumbo Payload option present and
Fragment header present

Code: 0
Pointer: high-order octet of the Fragment header.

A node that does not understand the Jumbo Payload option is expected to respond to erroneously-received jumbograms as follows, according to the IPv6 specification:

error: IPv6 Payload Length = 0 and
IPv6 Next Header = Hop-by-Hop Options

Code: 0
Pointer: high-order octet of the IPv6 Payload Length

error: IPv6 Payload Length != 0 and
Jumbo Payload option present

Code: 2
Pointer: Option Type field of the Jumbo Payload option

4. UDP Jumbograms

The 16-bit Length field of the UDP header limits the total length of a UDP packet (that is, a UDP header plus data) to no greater than 65,535 octets. This document specifies the following modification of UDP to relax that limit: UDP packets longer than 65,535 octets may be sent by setting the UDP Length field to zero, and letting the receiver derive the actual UDP packet length from the IPv6 payload length. (Note that, prior to this modification, zero was not a legal value for the UDP Length field, because the UDP packet length includes the UDP header and therefore has a minimum value of 8.)

The specific requirements for sending a UDP jumbogram are as follows:

When sending a UDP packet, if and only if the length of the UDP header plus UDP data is greater than 65,535, set the Length field in the UDP header to zero.

The IPv6 packet carrying such a large UDP packet will necessarily include a Jumbo Payload option in a Hop-by-Hop Options header; set the Jumbo Payload Length field of that option to be the actual length of the UDP header plus data, plus the length of all IPv6 extension headers present between the IPv6 header and the UDP header.

For generating the UDP checksum, use the actual length of the UDP header plus data, NOT zero, in the checksum pseudo-header [IPv6, Section 8.1].

The specific requirements for receiving a UDP jumbogram are as follows:

When receiving a UDP packet, if and only if the Length field in the UDP header is zero, calculate the actual length of the UDP header plus data from the IPv6 Jumbo Payload Length field minus the length of all extension headers present between the IPv6 header and the UDP header.

In the unexpected case that the UDP Length field is zero but no Jumbo Payload option is present (i.e., the IPv6 packet is not a jumbogram), use the Payload Length field in the IPv6 header, in place of the Jumbo Payload Length field, in the above calculation.

For verifying the received UDP checksum, use the calculated length of the UDP header plus data, NOT zero, in the checksum pseudo-header.

5. TCP Jumbograms

Because there is no length field in the TCP header, there is nothing limiting the length of an individual TCP packet. However, the MSS value that is negotiated at the beginning of the connection limits the largest TCP packet that can be sent, and the Urgent Pointer cannot reference data beyond 65,535 bytes.

5.1 TCP MSS

When determining what MSS value to send, if the MTU of the directly attached interface minus 60 [IPv6, Section 8.3] is greater than or equal to 65,535, then set the MSS value to 65,535.

When an MSS value of 65,535 is received, it is to be treated as infinity. The actual MSS is determined by subtracting 60 from the value learned by performing Path MTU Discovery [MTU-DISC] over the path to the TCP peer.

5.2 TCP Urgent Pointer

The Urgent Pointer problem could be fixed by adding a TCP Urgent Pointer Option. However, since it is unlikely that applications using jumbograms will also use Urgent Pointers, a less intrusive change similar to the MSS change will suffice.

When a TCP packet is to be sent with an Urgent Pointer (i.e., the URG bit set), first calculate the offset from the Sequence Number to the Urgent Pointer. If the offset is less than 65,535, fill in the Urgent field and continue with the normal TCP processing. If the offset is greater than 65,535, and the offset is greater than or equal to the length of the TCP data, fill in the Urgent Pointer with 65,535 and continue with the normal TCP processing. Otherwise, the TCP packet must be split into two pieces. The first piece contains data up to, but not including the data pointed to by the Urgent Pointer, and the Urgent field is set to 65,535 to indicate that the Urgent Pointer is beyond the end of this packet. The second piece can then be sent with the Urgent field set normally.

Note: The first piece does not have to include all of the data up to the Urgent Pointer. It can be shorter, just as long as it ends within 65,534 bytes of the Urgent Pointer, so that the offset to the Urgent Pointer in the second piece will be less than 65,535 bytes.

For TCP input processing, when a TCP packet is received with the URG bit set and an Urgent field of 65,535, the Urgent Pointer is calculated using an offset equal to the length of the TCP data, rather than the offset in the Urgent field.

It should also be noted that though the TCP window is only 16-bits, larger windows can be used through use of the TCP Window Scale option [TCP-EXT].

6. Security Considerations

The Jumbo Payload option and TCP/UDP jumbograms do not introduce any known new security concerns.

7. Authors' Addresses

David A. Borman
Berkeley Software Design, Inc.
4719 Weston Hills Drive
Eagan, MN 55123
USA

Phone: +1 612 405 8194
EMail: dab@bsdi.com

Stephen E. Deering
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Phone: +1 408 527 8213
EMail: deering@cisco.com

Robert M. Hinden
Nokia
313 Fairchild Drive
Mountain View, CA 94043
USA

Phone: +1 650 625 2004
EMail: hinden@iprg.nokia.com

8. References

[ICMPv6] Conta, A. and S. Deering, "ICMP for the Internet Protocol Version 6 (IPv6)", RFC 2463, December 1998.

[IPv6] Deering, S. and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification", RFC 2460, December 1998.

[MTU-DISC] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP Version 6", RFC 1981, August 1986.

[TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[TCP-EXT] Jacobson, V., Braden, R. and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[UDP] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

9. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.