Global Information Assurance Certification Paper

INTERNET PROTOCOL FRAGMENTATION PROCESSES, VERSION 4 VERSUS VERSION 6 & Path MTU Discovery

Donna Fortin
November 21, 2000

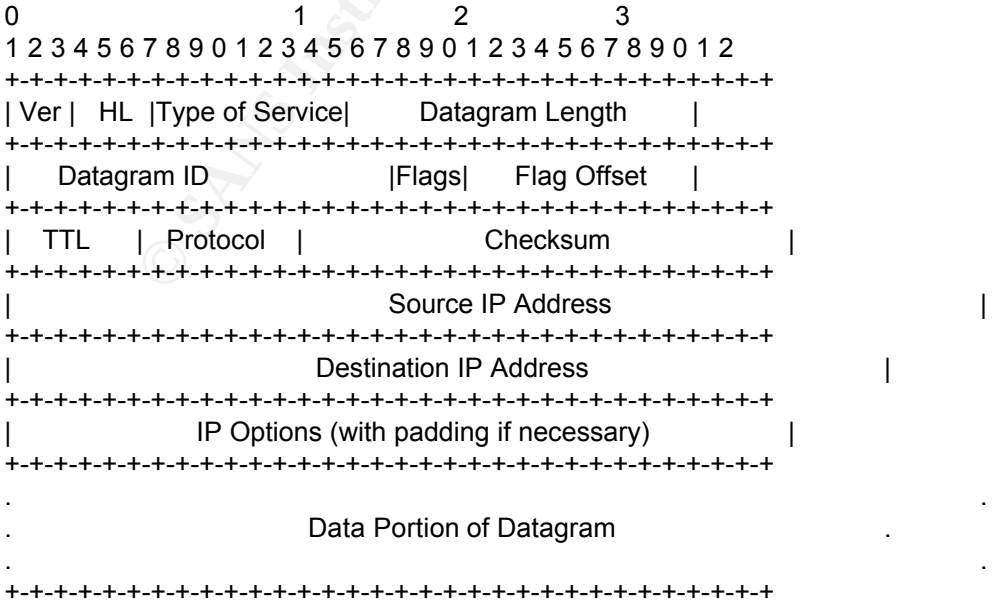
Internet Protocol Background / General Discussion

The first packet switch network was a four-node packet switching network known as Advanced Research Projects Agency Network (ARPANET). It went into operation in 1969. In 1974 a new suite of internet protocols known as TCP/IP was proposed by Vinton G. Cerf and Robert E. Kahn. The ARPANET remained the backbone of a growing evolution of academic and commercial research networks. By 1994, when the Internet was officially changed from a research testbed to a commercial service network it was made up of millions of interconnected computers. Today IPv4 is utilized throughout the internet.

The IP protocol utilizes encapsulation to deliver data. A simplistic definition of IP protocol follows. IP operates by accepting data from the next higher protocol, either TCP or UDP, creating a datagram, routing it through the network, and delivering it to the recipient host and then pertinent application. IP uses the subnet mask and IP routing tables to deliver the datagram to the next router or host on the path to the destination. The subnet mask helps determine whether or not the source node is on the same LAN as the destination. The routing table designates how the IP packet is routed when the destination node is not on the same LAN as the source node. All routers and hosts connected to a network and the internet have a routing table which defines the nodes within range of it. By routing to the next hop designated within each router or host routing table the datagram is transferred from source to destination.

IPv4 Fragmentation,

Figure 1 illustrates the fields required for an IPv4 datagram packet. The fields within the IPv4 header that directly support fragmentation of IPv4 packets include header length, length of datagram, identification, flags, and fragment offset. The header length is necessary since use of the option fields will create a longer header when utilized. The IPv4 header can be 20 bytes to 60 bytes. The length of datagram is important for network traffic. If during the routing of the datagram packet a network maximum frame size prevents the pass through of the existing packet it must be fragmented. The identification field consists of a 16-bit number that designates fragmented datagram packets as initially coming from the same packet.



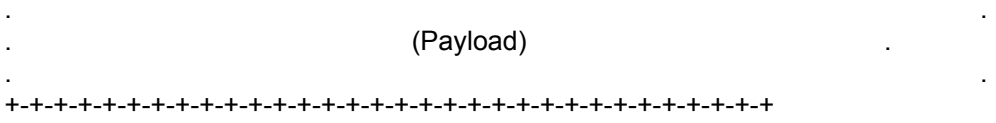


Figure 1 IPv4 Datagram Packet Header Format

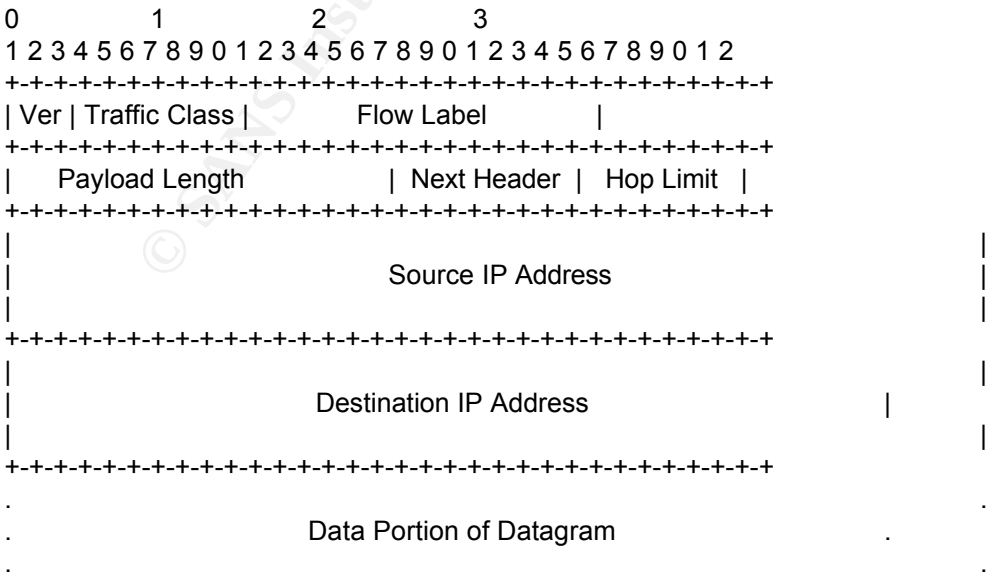
Flags and fragment offset fields work together to describe the specific location of the fragment within the original packet. One bit within the flag field indicates whether or not the packet may be fragmented. The fragment offset is a 13-bit field that displays what is known as the fragment block. The original datagram packet is broken into 8-bit blocks known as fragment blocks. The 13-bit field displays this fragment block value from 0 to 8191, which directly corresponds to a 0 to 65,528-bit offset from the original datagram packet. Another bit within the flag field shows if this is the last fragment of the original datagram or more come after this particular block.

With the present options available to IP version 4 the internet has proven very reliable for data transfer up through the 1980s and 1990s yet IPv4 has become less efficient as the internet has continued to expand into the new century. IPv4 was not designed to manage the extent of nodes it supports today. As the personal computer market has expanded in the past decade so has the requirement for a larger internetworking solution for TCP/IP. Over the past 5 years there has been efforts to create a next generation TCP/IP to incorporate requirements of today's internet. IPv6 offers many more distinctive attributes.

IPv6 Fragmentation / Enhancements for Packet Transfer

The actual operation process of IPv6 is the same as IPv4. In order to phase IPv6 into operation IPv6 can be utilized within an IPv4 environment. In order to anticipate an expanding internet IPv6 has included simplification of the IP header in order to realize increased operability from the internet. The fragmentation handling process with IPv6 has been simplified to create less packet processing required in transit. A brief description of IPv6 extension headers is required here for clarification on IPv6 fragmentation policy.

IPv4 utilizes standard header fields that offer the consistent options described above. All routers and hosts along the path as well as the destination host still must process all the fields within the IPv4 header whether or not they are utilized. IPv6 utilizes a slightly different model for its datagram header and IP processing than IPv4; it has fewer fields and is a standard size. Figure 2 depicts the fields required for an IPv6 datagram packet.



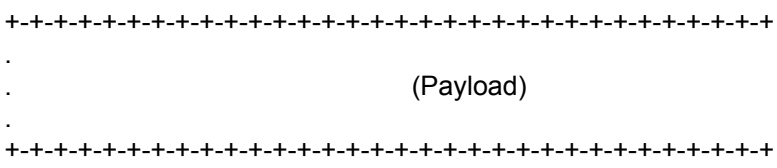


Figure 2 IPv6 Datagram Packet Header Format

IPv6 utilizes “extension” headers for each datagram packet that requires special options. The “Next Header” field delineates the next header. When options are required the next header would be an extension header. All extension headers reside within the data portion of the datagram and are not considered part of the header itself. Figure 3 illustrates the extension header concept and Figure 4 shows the standard extension header format. The IPv6 specification (RFC 2460) recommends that the next headers be placed in a specific order. The required order is IPv6 header, hop-by-hop extension header, destination options header, routing header, fragment header, authentication header, encapsulation security payload header, destination options header, and finally upper layer protocol header. Thus, when there are no additional IP options required the next header designated would be the higher protocol such as TCP or UDP.

IPv6 has two categories of extension headers. One category requires processing by every node between the source and destination (“hop by hop” type extension headers) while the other category (all remaining extension header types) requires processing by the destination host only. Figure 3 shows the difference between these two type headers by showing node perception along the IPv6 datagram path.

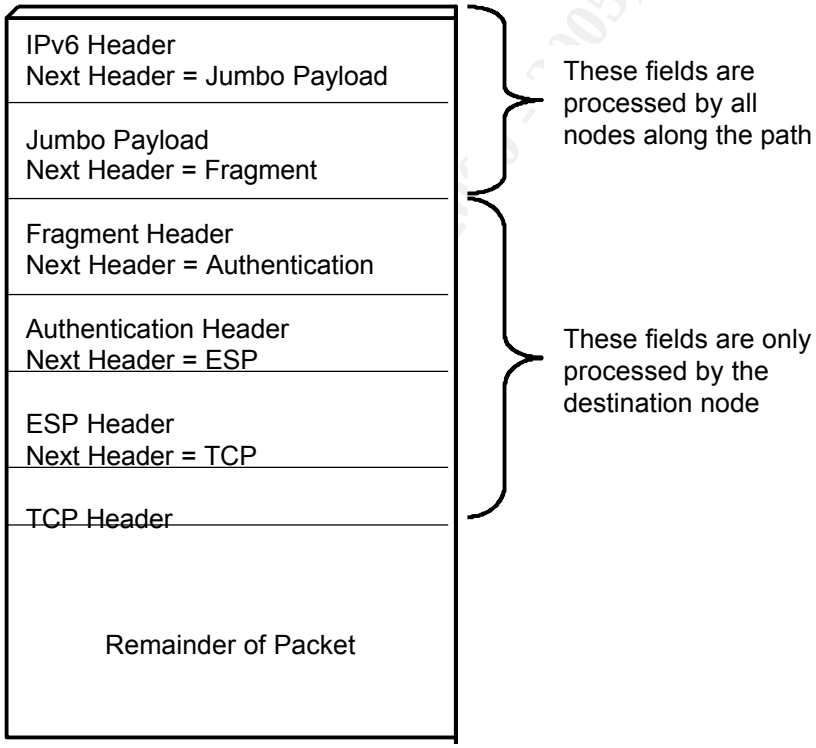


Figure 3-3 Extension Header concept holding 4 extension headers

0 1 2 3

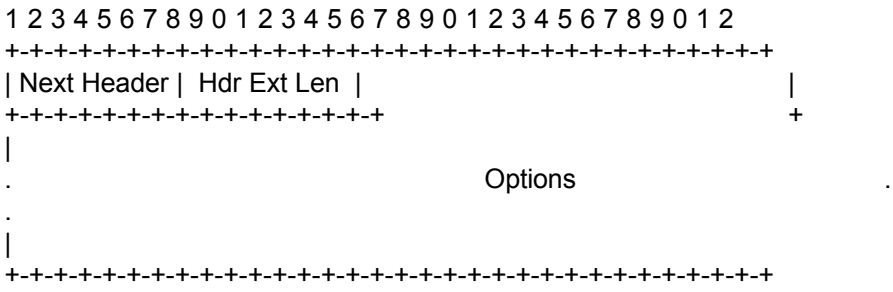


Figure 4 Standard hop-by-hop / destination Extension Header Format

Thus, the concept of datagram packet fragmentation for IPv6 is managed fundamentally different than for IPv4. IPv4 packets can be fragmented at any intermediate node along the path that does not allow packets as large as the original packet size. On the other hand, IPv6 allows fragmentation only at the originating or source node. With IPv6 the fragmentation option is utilized as an extension header with its own format.

Figure 5 portrays the fragmentation header format. The next header field (8 bits) is the format of the subsequent header field. The next field (8 bits) of the fragmentation header is reserved for future use. The fragment offset (13 bits) value tells the destination numbered in 8-bit segments where this portion fits within the fragmentable portion of the packets. The fragmentable portion includes only the payload and extension headers that are to be processed when the packet has arrived at its final destination. The next field is another field reserved (2 bits) for future use. The next field is the M flag which when the value is 1 indicates another fragment is forthcoming, a zero indicates this is the last fragment. The final identification field holds a 32-bit identifier that is intended to uniquely identify any packet sent recently.

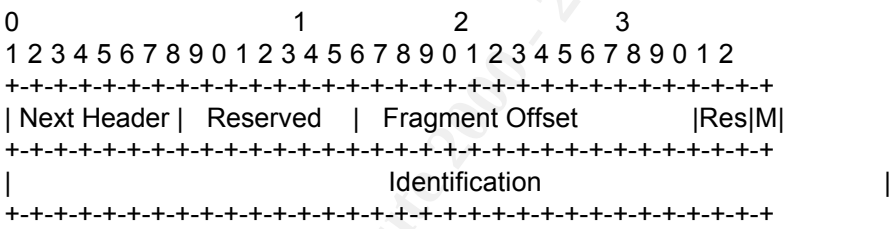


Figure 5 Fragmentation Header Format

Path MTU Discovery IPv4 / IPv6

Since fragmentation within IPv4 can potentially hamper throughput, and fragmentation in IPv6 requires path mapping, path Maximum Transmission Unit (MTU) discovery is imperative for proper operation of IP networks. Simply put, the path MTU discovery is a process where the originating node sends a packet to the destination. Along the path it checks the largest allowable MTU size of each node and delivers this information back to the originating node prior to releasing any datagram packets to the destination. The packets now sent from source to destination will be no larger than the intermediate nodes with the lowest capability of MTU size discovered through the MTU discovery process.

For IPv4 path MTU discovery was implemented through RFC 1063 and later obseleted by RFC 1191. In order to utilize MTU discovery within IPv4 he *don't fragment* bit in the "flag" field must be set. The size of the packet sent is the assumed MTU, which equals the known MTU of the first hop. During relay of the packet if fragmentation is required at any node due to the size of the packet, the packet is discarded and an ICMP stating the destination is unreachable due to fragmentation required but DF (don't fragment) field set. The source sends another packet of smaller size. The path MTU discovery process is continued until the packet is

delivered without an ICMP reply. In short, path MTU discovery for IPv4 increases throughput of traffic between hosts on different LANs. Note that if the DF flag is not set then the packets will be fragmented along the path unbeknownst to the source host.
RFC 1981 specifically suggests that "IPv6 nodes SHOULD implement Path MTU Discovery in order to discover and take advantage of paths with PMTU greater than the IPv6 minimum link MTU." Since IPv6 does not allow fragmentation along the path, path MTU discovery is implemented by sending packets of varying size for the sole purpose of discovering the largest possible packet for a particular path between a source and destination. Note that when the source host has not implemented path MTU discovery, packets sent out that are too large and illicit an ICMPv6 reply will be rectified at the source node by its sending smaller packets. The difference here with MTU discovery is that the source node will not send these packets out only to map the packet's path to destinations.

The Computer Science Department of the Tel Aviv University has suggested an addition to the IPv6 protocol to alleviate some of the shortcomings of the PMTU discovery algorithm. This addition to the IPv6 protocol would be the use of a new hop by hop extension headers known as a Detection Option and a destination extension header known as Indication Option. Both the Detection Option and Indication Option headers consist of three fields, option type, option data and affirmative PMTU.

For an initial detection of PMTU by a source node a Detection Option header would be sent within an IPv6 packet of minimum MTU size (576 bytes). The affirmative PMTU field within the Detection Option header would take on the value of the first hop link MTU value. Since it is a hop by hop extension header it will be processed at each node along the path. Each subsequent node along the path would compare the affirmative PMTU field with the next hop link and replace the affirmative PMTU field with the lower of the two values.

Once at the destination the affirmative PMTU field would hold the maximum size packet able to transverse the particular path. This value would be then be stored statically within the destination node. The destination node would then send an Indication Option header with the same data back to the source node. Since the Indication Option header is a destination extension header only the source and destination nodes would process the data within its fields.

Conclusion

Internet Protocol is the backbone of the world Internet. Because it is impossible to standardize capabilities of all the hosts and routers around the world packet fragmentation policies are necessary within IP standards to maintain structure. IPv4 and IPv6 were compared here to show their different strategies. Yet, both of these fragmentation policies expose hosts to potential service attacks through ICMP replies. It is important to investigate possibilities such as Option headers within the IPv6 protocol to subjugate the deficiencies of existing fragmentation policies in order to begin limiting Information Assurance attacks on our operational networks.

Works Cited

Feit, S., TCP/IP Signature Edition, McGraw-Hill, 1998.

Deering, S., Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification," Request for Comments 2460, December 1998, URL: http://www.ietf.cnri.reston.va.us/rfc/rfc2460.txt?number=2460 (17 Nov 2000).

Loshin, P., IPv6 Clearly Explained, Morgan Kaufmann, 1999.

Mogul, K., Kent, C., Partridge, C., McCloghrie, K., "IP MTU Discovery Options," Request for Comments 1063, July 1988, URL: http://www.ietf.cnri.reston.va.us/rfc/rfc1063.txt?number=1063 (20 Nov. 2000).

Mogul, K., Deering, S., "Path MTU Discovery," Request for Comments 1191, November 1990, URL: http://www.ietf.cnri.reston.va.us/rfc/rfc1191.txt?number=1191 (18 Nov. 2000).

McCann, J., Deering, S., Mogul, K., "Path MTU Discovery for IP version 6," Request for Comments 1981, August 1996, URL: http://cellworks.washington.edu/sage/1997/06/rfc1981.txt (18 Nov. 2000).

Kreidenko, V., Burtman, D., "Protocols & Computer Networks Project Path MTU Discovery for IP version 6," URL: http://www.rad.com/networks/1997/mtudisc/project.html (13 Nov. 2000).

Last Updated: July 25th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event