

**Comparison of IPv4 and IPv6 Header Overhead**  
**draft-eddy-ipv6-overhead-00**

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 9, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document provides an analysis and comparison of IPv4 and IPv6 header sizes under various circumstances. The goal of this document is to provide hard evidence for use in frequent discussions regarding transition, where header overhead comes up as an issue used to portray IPv6 depolyment as untenable. The results show that for links that are not fully utilized, the IPv6 overhead would only add a few percent to their load over what IPv4 implies, while for congested links, we note that header compression techniques for IPv6 are at least as effective as those for IPv4. This demonstrates that the

header overhead argument against IPv6 is misinformed.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Methodology . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Case Studies . . . . .	<a href="#">7</a>
<a href="#">3.1</a>	Case 0: Boundary Conditions . . . . .	<a href="#">7</a>
<a href="#">3.2</a>	Case 1: Normal Range of Behavior . . . . .	<a href="#">8</a>
<a href="#">3.3</a>	Case 2: Fragmentation . . . . .	<a href="#">9</a>
<a href="#">3.4</a>	Case 3: Jumbograms . . . . .	<a href="#">10</a>
<a href="#">3.5</a>	Case 4: Mobility . . . . .	<a href="#">11</a>
<a href="#">4.</a>	Summary of Findings . . . . .	<a href="#">13</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">14</a>
<a href="#">6.</a>	Acknowledgements . . . . .	<a href="#">15</a>
<a href="#">7.</a>	Informative References . . . . .	<a href="#">15</a>
	Author's Address . . . . .	<a href="#">16</a>
<a href="#">A.</a>	Useful Numeric Values . . . . .	<a href="#">17</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">19</a>

Eddy

Expires November 9, 2006

[Page 2]

## **1. Introduction**

While IPv4 has achieved widespread use and acclaim, its intended successor, IPv6, is still facing some hurdles in large-scale deployment. In both aeronautical networking and space networks a move towards network-centric operations and away from application-specific point-to-point links is occurring. In multiple groups that are attempting to define aeronautical or space networking architectures, the use of Internet protocols is well-accepted, but there is considerable uncertainty on whether to use IPv4, IPv6, or a dual-stack.

It is the technical opinion of many that IPv6 is favorable, due to some of its features (mobility and security are particularly important for network-centric operations). We have also seen IPv4 brought forward as a favored choice by others based on the logic that it has lower "overhead" than IPv6, and that aeronautical and space links may often be rather constrained. While it is undeniable that the IPv6 base header is larger than IPv4's, these arguments tend to be "hand-waves" that are not based on well-quantified data.

In this document, we systematically consider IPv4 and IPv6 header overhead in several case studies. Our goal is to provide stronger material for use in discussions regarding the comparison between IPv4 and IPv6 header sizes, and their effect on link loading. Companion documents debunk the related myth that IPv6 is not yet mature enough for production use [[Eiv06](#)], and compare the feature sets of IPv6 and IPv4 [[Eis06](#)].

Before proceeding into this study, the reader should make an assessment of the potential relevance of this issue. For networks that are only lightly loaded, where there is very little congestion or delay, what effect will increasing the size of packets by only a few percent have? Contrast this with chronically oversubscribed links, perhaps due to low physical layer capacities, where there might be substantial queuing, and consider what effect adding several dozen bytes to each packet has. Clearly this issue may or may not be important under different assumptions. In the cases where it is important, it is likely that header compression techniques are in use, and so the real trade study should be between the effectiveness of IPv4 and IPv6 header compression mechanisms. Current methods are able to compress IPv6 headers at least as well, if not better, than IPv4 headers [[RFC3095](#)].

The problem of header overhead in both IPv4 and IPv6 has long been recognized and a large amount of work has gone into header compression techniques. The IETF Robust Header Compression Working Group, for instance, has delivered particularly effective solutions

Eddy

Expires November 9, 2006

[Page 3]

for both versions of IP [[RFC3843](#)] as well as additional protocols layered over IP. It is our primary advice that for under-utilized links, header overhead is not a significant consideration, while for low-speed or congested links, current header compression technologies are fully sufficient and make this topic moot. The remainder of this document assumes no header compression technique is in use, and then presents quantized evidence that might be compared to the utilization levels of user LANs or provider backbones to see how insignificant the difference in overhead between IP versions is.

[Section 2](#) describes the basic terminology and procedure used in this study and introduces the scenarios considered in the case studies in [Section 3](#). A brief summarization of the results is given in [Section 4](#). [Appendix A](#) contains some numeric quantities that are used in the analysis.

Eddy

Expires November 9, 2006

[Page 4]

## 2. Methodology

For the purposes of this study, we will consider the "overhead" to consist solely of the IP-layer headers, options, and extension headers. To be specific:

- o We define the IP-layer Service Data Unit (SDU) as whatever buffer of bytes is passed down from a transport protocol (or other layer above IP), to be sent over the network using IP. The IP-layer Protocol Data Unit (PDU) that is constructed then consists of the SDU combined with IP Protocol Control Information (PCI). This follows the nomenclature used in the ISO/OSI reference model.
- o We assume that only the PCI portion of the IP-layer PDU varies between IPv4 and IPv6. Thus the difference in overhead can be determined strictly from the size of the PCI. There are certain applications that may carry IP addresses inside their SDUs (e.g. FTP or IKE), but our assumption is that this has only a relatively small affect overall (in FTP, the file transfered is usually much longer than the length of an IPv6 address, and in IKE, certificates and keying material are similarly much larger than an IPv6 address).
- o A few components of the PCI may be identical between IPv4 and IPv6. For instance, the IPsec Authentication Header has the same size and form when used with IPv4 as it does with IPv6. In this study, we will only consider PCI components that differ between the two versions of IP.

Measuring the PCI as a raw number of bytes is useful, but it is also informative to view the lengths of the IPv4 PCI and IPv6 PCI as percentages of the total PDU. In the case of relatively small SDUs, a large PCI size is significant, but in the case of larger SDUs (as used in bulk file transfer), a difference of even several dozen bytes in the PCI size will have little effect on the percentage of the PDU consumed by the PCI.

Since fragmentation of datagrams is treated differently in the IPv4 and IPv6 protocols, we will consider what happens when PDUs known to be larger than the Path MTU (PMTU) are handled by each version of IP, in addition to PDUs that fit within the PMTU.

Even with these considerations, this is a somewhat naive comparison since the overhead implied by the selection of a network layer protocol also consists of several factors in addition to the per-packet headers. This includes the size of updates in routing protocols (e.g. OSPF and BGP), the address fields in queries and responses to address resolution protocols (e.g. DNS and ARP/ND), and



Eddy

Expires November 9, 2006

[Page 5]

the configuration protocol packets (e.g. DHCP), to name a few additional contributions. Compared to the per-packet overhead, however, these should be negligible contributions.

We define a number of specific cases for which we compute IP-layer overhead:

Case 0: (a) Consider a 0-byte SDU (i.e. an empty payload). (b) Then consider a 65,515-byte SDU. Assume the PMTU in both of these cases is greater than the PDU size.

Case 1: Consider an N-byte SDU (for  $N > 0$ ), when the PMTU is assumed to be greater than the PDU size.

Case 2: Consider an N-byte SDU, when the PMTU is assumed to be less than N, but greater than  $(N + (PCI * 2)) / 2$  (i.e. the SDU can fit within two IP packet fragments). This demonstrates a simple case of fragmentation, where only two fragments are needed.

Case 3: Consider an N-byte SDU, where  $N > 65,535$ , and the PMTU is large enough to support the IPv6 PCI and SDU. This allows for consideration of jumbogram support in IPv6, which IPv4 lacks.

Case 4: Consider an N-byte SDU sent between a mobile node (MN) and a corresponding node (CN), where the PMTU is greater than the PDU size, and where the MN and CN support the standard mobility features of IPv4 or IPv6 (including the route optimization feature which is a part of IPv6, but not IPv4).

Eddy

Expires November 9, 2006

[Page 6]

### **3. Case Studies**

In this section, each of the specific cases listed previously is further described and analyzed.

#### **3.1 Case 0: Boundary Conditions**

One way the reader might view the two subcases presented here of minimum and maximum-sized packets, is as demonstration of the futility of even attempting to gauge header overhead without any knowledge of what the offered transport/application load is. For small packets, any network header at all seems extraordinarily inefficient since in either IP version the node IP addresses alone may be larger than the data. For large packets, the reverse is true.

To start, it should be noted that the first part of this case is somewhat unrealistic. In reality, a 0-byte SDU would never be passed to the IP layer, since to even address a particular application or service requires a transport header, and even IP-layer signalling occurs using ICMP datagrams as SDUs, so there is no practical use for a 0-byte SDU. This case simply illustrates a hard limit on the smallest size an IP PDU can take, and the worst relative amount of the PDU that the PCI can consume.

If it were possible to send a 0-byte PDU, then the overheads for IPv4 and IPv6 would be:

IPv4 PCI (IPv4 Base Header): 20 bytes or 100% of the PDU

IPv6 PCI (IPv6 Base Header): 40 bytes or 100% of the PDU

IPv4 and IPv6 both have the same worst-case bound on the percentage of the PDU that their headers can take up, although the base IPv6 header is twice as large as the base IPv4 header. Since in reality, many types of link pad small frames out to reach some minimum length, small packets (those of only a few bytes in the SDU), may still be under the link's minimum and require padding. This means that even measuring the IP header overhead for small packets may be pointless, since even if the IP header were compressed to 0-bytes, the link would apply padding to the frame. For instance, Ethernet frames have a minimum frame size of 64 bytes, and Gigabit Ethernet's minimum is even larger at 520 bytes. In the case of standard Ethernet, with 18 bytes of frame header/trailer, IPv6 SDUs of up to 6 bytes and IPv4 SDUs of 26 bytes have the same real overhead.

At the other end of the spectrum, the largest possible IPv4 packet is 65,535 bytes, due to the 16-bit IPv4 total length field. IPv6 has a similar 16-bit field, but its semantics differ, in that it encodes

Eddy

Expires November 9, 2006

[Page 7]

the payload length. This means that a single IPv6 packet may be larger than an IPv4 one by the size of the IPv6 header plus the size of the IPv4 header (60 bytes using only base headers). Given a PDU of 65,515 bytes (the maximum possible in IPv4):

IPv4 PCI (IPv4 Base Header): 20 bytes or 0.03% of the PDU

IPv6 PCI (IPv6 Base Header): 40 bytes or 0.06% of the PDU

Neither the IPv4 nor IPv6 headers make up any significant portion at all of the PDU in the case of maximally-sized standard packets. The case of jumbograms considered later in Case 3, shows that IPv6 is actually capable of being even more efficient than this, but at this scale it is almost absurd to even consider.

### **3.2 Case 1: Normal Range of Behavior**

This case considers an IP packet that can be sent end-to-end with no fragmentation. This is by far the most prevalent case under normal operating conditions, and so the figures presented for this case should have the most bearing on a practical determination on the significance of the header overhead issue.

IPv4 PCI (IPv4 Base Header): 20 bytes or  $(20/(N+20)) * 100\%$  of PDU

IPv6 PCI (IPv6 Base Header): 40 bytes or  $(40/(N+40)) * 100\%$  of PDU

Since the PCI is simple (a single IP header) in the case of each protocol version, the impact of the PCI on the PDU is easy to understand. For instance, on a 48-byte SDU (the minimum IPv4 MTU [[RFC0791](#)], minus the IPv4 base header size), the IPv4 and IPv6 PCIs represent 29.41% and 58.82% of the PDU, respectively. For a 1240-byte SDU (the minimum IPv6 MTU, minus the IPv6 base header size), the IPv4 and IPv6 PCIs represent 1.58% and 3.13% of the PDU, respectively.

While it is true that the IPv4 overhead is half that of the IPv6 overhead in this case, with a reasonable-sized SDU, as would be used by applications such as bulk-transfer, both protocol's headers are of relatively little consequence, only making up a few percent of the PDU. Note that when comparing the IPv4 base header size to the IPv4 minimum MTU (29.41%) and the IPv6 base header to the IPv6 minimum MTU (3.13%), the results of the requirement to support larger MTUs on IPv6 links are seen to allow for keeping a much tighter handle on the header overhead than was possible with IPv4's requirement.

As an example of interactive traffic that sends small packets, consider the SSH protocol when the user types a character at a time

Eddy

Expires November 9, 2006

[Page 8]

into a remote shell. A single typed character causes a 48-byte block of application data to be passed to TCP. TCP puts a 20-byte base header onto this, and then (usually) a 12 byte Timestamp Option, so an 80-byte SDU ( $48+20+12$ ) is passed to IP, regardless of whether IPv4 or IPv6 is being used. With base IP-layer headers then, IPv4 PCI takes up exactly one fifth of the PDU, while the IPv6 PCI takes up one third of the PDU.

We can also consider very large MTUs, such as those that might be used on a fiber-optic network bus in an aircraft or space vehicle. The FDDI MTU of 4352 bytes allows a 4332-byte SDU with IPv4 and a 4312-byte SDU with IPv6 base headers. Respectively, the PCIs are 0.46% and 0.92% of the PDU. OC-3 packet over SONET interfaces default to a similarly sized MTU of 4470 bytes, giving PCIs that may be as little as 0.45% and 0.89% of the PDU. In the case of such links, the header overhead argument against IPv6 seems largely irrelevant.

### **3.3 Case 2: Fragmentation**

IPv4 and IPv6 take completely different approaches to fragmentation. In IPv4, fragmentation of a datagram can occur at any point in the network where a particular link's MTU is too small to accomodate the entire packet (assuming the Don't Fragment bit is not set); i.e. in IPv4, fragmentation is a router-level function. In IPv6, a router never fragments a packet. If an IPv6 packet is larger than a link MTU, then an ICMPv6 Packet Too Big message is sent to the packet's source, and the packet is dropped. Fragmentation has been removed from a router's responsibilities in IPv6, and is strictly an end-host's task to perform, as needed. For this reason, the header overhead during a fragmentation scenario in IPv4 and IPv6 has to be specially compared, since it is static across the path in IPv6, but differs after the router that performs it in IPv4. We assume that from a prior failure or PMTU probing, the IPv6 end-host already knows that fragmentation is required here, so that a failed transmission does not factor into the overhead computed.

First consider a simple case that only requires a packet to be segmented into two fragments:

IPv4 PCI, before fragmenting router (IPv4 Base Header): 20 bytes  
or  $(20/(N+20)) * 100\%$  of PDU

IPv4 PCI, after fragmenting router (2x IPv4 Base Headers): 40  
bytes or  $(40/(N+40)) * 100\%$  of PDU



Eddy

Expires November 9, 2006

[Page 9]

IPv6 PCI, (2x IPv6 Base Headers and IPv6 Fragment Headers): 96 bytes or  $(96/(N+96)) * 100\%$  of PDU

As an example, consider the case of a PMTU of 1280 bytes (the IPv6 minimum), and an SDU of 1400 bytes, where the MTU of the first-hop link is greater than 1420 bytes plus link headers (e.g. an Ethernet). An IPv4 host will send a single 1420 byte PDU on the first link, with PCI overhead of 1.41%. This IPv4 packet then becomes fragmented, at some point, into two IPv4 packets, one of which is of size 1280 bytes (1260 bytes of the original SDU and 20 of IPv4 header) and another of size 160 (140 bytes of the original SDU and 20 of IPv4 header). The first new packet has overhead of 1.56%, and the second has overhead of 12.5%. Together, the combined PCI is 2.78% of the combined PDU sizes.

Correspondingly, in IPv6, when it is not performing PMTU Discovery, a host is likely to pro-actively fragment its packets to be within the 1280 byte guideline (or even lower to accomodate some tunneling). So in a simple setting, the 1400-byte SDU might be fragmented into a 1280-byte and a 216-byte pair of packets (carrying 1232 and 168 bytes of the original SDU each). These would have overhead of 3.75% and 22.22% each, and a combined overhead of 6.42%. This overhead applies across the entire end-to-end path.

In general, transport protocols will either not send large SDUs to IP (they will segment data at the transport layer), or they will perform some form of PMTU Discovery (e.g. [\[RFC1191\]](#)) in order to send the largest segments possible without causing fragmentation. In transport protocol implementations that support IPv6, when reliable operation is required of the transport, a resegmentation and retransmission occurs in response to ICMPv6 Packet Too Big messages. If the PMTU is not known, and the transport sends a segment that is too large, which is then retransmitted as either multiple IPv6 fragments or resegmented transport data, then the overhead is actually different than what we report here. The entire PDU sent in the failed attempt can be considered overhead, as well as the ICMPv6 message in the reverse direction. If the SDU is resegmented by the transport, then extra transport overhead is imposed.

### [3.4](#) Case 3: Jumbograms

One feature that is part of IPv6, but has no analogue in IPv4 is jumbogram support [\[RFC2675\]](#). This has proven useful mainly in super-computing contexts, but to our knowledge has not been deployed significantly outside this domain. It may be relevant to satellite networks that interconnect supercomputers. A base IPv4 or IPv6 header includes a 16-bit field for the payload length, but IPv6 has the Jumbo Payload option, which allows this field to be overridden

Eddy

Expires November 9, 2006

[Page 10]

with a 32-bit field. This permits two IPv6 hosts with sufficient knowledge of each other's and the network's capabilities to send single packets of larger than 65,535 bytes.

The jumbogram feature of IPv6 allows larger SDUs than are possible in IPv4, and thus can reduce both network layer and transport (or higher) layer overhead. Sending a jumbogram requires coordinated support from several protocol layers (at least the link, network, and transport), and is only possible when as transport/application actually has greater than 65,515 bytes of data to send. In the case where this is possible in IPv6, compared to what would be required in IPv4 for data totaling  $N$  ( $> 65,515$ ) bytes:

IPv4 PCI ( $H = \text{ceil}(N/(65,535-20))$  IPv4 Base Headers):  $H*20$  bytes or  $(H*20/(N+H*20) * 100)\%$  of PDU

IPv6 PCI, (IPv6 Base Header, IPv6 Hop-by-Hop Option, and IPv6 Jumbo Payload Option): 48 bytes or  $(48/(N+48) * 100)\%$  of PDU

As a quick example, consider  $N=200,000$ . Using IPv4, for maximum efficiency, the transport could put this into 4 IPv4 SDUs (assuming it either knew the link could support frames of this size, or did not care about fragmentation for some reason). This would give an overall header overhead of 0.03% of the PDU. If a single IPv6 jumbogram is used, the overhead is 0.02% of the PDU. In terms of header overhead, this is not an area where there is neither an issue nor any real difference between IPv4 and IPv6. Jumbograms allow a transport to operate more efficiently by not segmenting and reassembling data itself (usually involving slow memory copies), and possibly allow the link to be more efficient, but they do not significantly affect network layer protocol overhead.

### **3.5 Case 4: Mobility**

Due to the way Mobile IPv4 operates (in its most efficient mode), using triangle routing, packets will cross part of their path within a tunnel, and then another part regularly, with no tunnel. Thus, the IPv4 PCI size changes depending on where in the network it is measured when Mobile IPv4 is used. On the other hand, we will assume a Mobile IPv6 scenario where Route Optimization (RO) is supported, such that packets go directly to their destination without tunneling. This is a feature of IPv6 that has no analogue in IPv4. In a Mobile IPv6 with RO setting, though, different PCI components get placed on a packet depending on whether a mobile node (MN) is using a Care-of Address as a "from" address in outgoing packets, whether the Care-of Address is being used as a "to" address by a corresponding node (CN), or whether Care-of Addresses are used in both directions (between two MNs, both away from their "home" networks). The reader should refer

Eddy

Expires November 9, 2006

[Page 11]

to a more detailed Mobile IP reference if these differences seem confusing (e.g. [[RFC3775](#)]).

Mobile IPv4 and Mobile IPv6 (including NEMO) can also both operate in a bidirectional tunnel mode, in which a direct path between MN and CN, or vice-versa, is never taken, but packets in both directions are always redirected through the home network. Since this case is even less desirable than a triangle route, to simplify analysis, we do not consider it here.

IPv4 PCI, inside tunnel (2x IPv4 Base Headers): 40 bytes or  $(40/(N+40) * 100)\%$  of PDU

IPv4 PCI, outside tunnel (IPv4 Base Header): 20 bytes or  $(20/(N+20) * 100)\%$  of PDU

IPv6 PCI, from CN to MN (IPv6 Base Header and Mobile IPv6 Type 2 Routing Header): 64 bytes or  $(64/(N+64) * 100)\%$  of PDU

IPv6 PCI, from MN to CN (IPv6 Base Header, IPv6 Destination Option, Mobile IPv6 Home Address Option, and padding): 64 bytes or  $(64/(N+64) * 100)\%$  of PDU

IPv6 PCI, between two MNs away from home (IPv6 Base Header, IPv6 Destination Option, Mobile IPv6 Home Address Option and Mobile IPv6 Type 2 Routing Header, and padding): 88 bytes or  $(88/(N+88) * 100)\%$  of PDU

In Mobile IPv4, the standard means of tunneling involves using two IPv4 base headers, although other methods are available, some of which can be more efficient. In some of the Mobile IPv6 with R0 cases, padding has to be added to the PCI to meet the IPv6 requirement of ending on a 64-bit boundary. Even when the IPv4 PCI is at its largest, during the tunneled portion of its journey through the network, it is still smaller than any of the IPv6 PCI configurations when a mobile node is using a Care-of Address. In the worst case of IPv6 overhead when both nodes are mobile and in foreign networks, the 88 bytes of PCI would make up 6.87% of a 1280-byte PDU.



#### **4. Summary of Findings**

In general, header overhead at the IP layer is not a relevant concern in most normal networks in use today for two reasons. The first is that the bandwidth of many currently-used links (in LANs, and many provider backbones) seems to be at least adequate, if not plentiful. For very constrained links, where bandwidth is tight and demand may be high (as in many types of wireless links), then header overhead becomes an issue. But the second reason why IP header overhead is not usually a valid concern, is that in these specific links, header compression techniques can usually be applied to significantly reduce IP overhead.

This document computed the size of IPv4 and IPv6 headers relative to the size of the SDU from a transport/application protocol over a number of different scenarios and configurations. Typically, IPv6 involved more overhead, but only differed from IPv4 within several percent of the SDU's size, despite the base IPv6 header being twice as big as the IPv4 header.

The impact of increasing the capabilities of a protocol at the expense of blowing up the header overhead is not a new consideration. One documented example is the analysis of the TCPng proposal in [RFC 1705](#) [RFC1705]. The IPng design effort that resulted in IPv6 considered this and concluded with a 40-byte IPv6 header, a doubling of IPv4's 20-byte header. Most of the case studies we have presented in this document show that in reality, this difference in header size is less significant than it seems at first.





## **5. Security Considerations**

This informational document only contains a comparison of header sizes. There are no security considerations.

## 6. Acknowledgements

Work on this document was performed at NASA's Glenn Research Center, in support of the NASA Space Communications Architecture Working Group (SCAWG), and the FAA/Eurocontrol Future Communications Study (FCS). Will Ivancic and Joe Ishac of NASA contributed useful comments on this document, as well as Terry Bell, Bob Dimond, and Dave Stewart of Verizon.

## 7. Informative References

- [EIs06] Eddy, W. and J. Ishac, "Comparison of IPv6 and IPv4 Features", [draft-eddy-ipv6-ipv4-comparison-00](#) Internet-Draft (work in progress), May 2006.
- [EIV06] Eddy, W. and W. Ivancic, "Assessment of IPv6 Maturity", [draft-eddy-ipv6-maturity-00](#) Internet-Draft (work in progress), May 2006.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", [RFC 1191](#), November 1990.
- [RFC1705] Carlson, R. and D. Ficarella, "Six Virtual Inches to the Left: The Problem with IPng", [RFC 1705](#), October 1994.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", [RFC 2675](#), August 1999.
- [RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", [RFC 3095](#), July 2001.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", [RFC 3775](#), June 2004.
- [RFC3843] Jonsson, L-E. and G. Pelletier, "RObust Header Compression (ROHC): A Compression Profile for IP", [RFC 3843](#), June 2004.

Eddy

Expires November 9, 2006

[Page 15]

Author's Address

Wesley M. Eddy  
Verizon Federal Network Systems  
21000 Brookpark Rd, MS 54-5  
Cleveland, OH 44135

Phone: 216-433-6682

Email: [weddy@grc.nasa.gov](mailto:weddy@grc.nasa.gov)

## [Appendix A](#). Useful Numeric Values

This appendix compiles a list of various PCI components and their sizing information, as well as some MTUs, limits, and link or Subnetwork Defined Convergence Protocol (SNDCCP) requirements that are used in the case studies.

PCI Component	Size	Reference
IPv4 Base Header	20 bytes	[ <a href="#">RFC0791</a> ]
IPv6 Base Header	40 bytes	[ <a href="#">RFC2460</a> ]
IPv6 Hop-by-Hop Options Header	2 bytes	[ <a href="#">RFC2460</a> ]
IPv6 Destination Options Header	2 bytes	[ <a href="#">RFC2460</a> ]
IPv6 Fragment Header	8 bytes	[ <a href="#">RFC2460</a> ]
IPv6 Jumbo Payload Option	6 bytes	[ <a href="#">RFC2675</a> ]
Mobile IPv6 Type 2 Routing Header	24 bytes	[ <a href="#">RFC3775</a> ]
Mobile IPv6 Home Address Option	18 bytes	[ <a href="#">RFC3775</a> ]



Size	Importance
64 bytes	minimum transmitted Ethernet frame
68 bytes	IPv4 (or SDCP) minimum MTU
520 bytes	minimum transmitted Gigabit Ethernet frame
576 bytes	X.25 MTU and minimum packet that IPv4 hosts are required to handle
1280 bytes	IPv6 (or SDCP) minimum MTU
1500 bytes	Ethernet IP MTU
4352 bytes	FDDI IP MTU
4470 bytes	OC-3 SONET IP MTU
65535 bytes	IPv4 maximum total packet size, IPv6 maximum payload size





## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

## Acknowledgment

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

Eddy

Expires November 9, 2006

[Page 19]