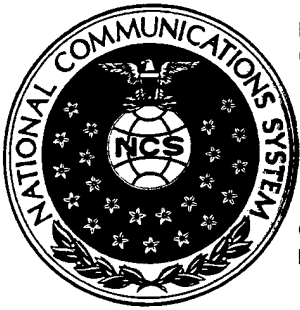


NCS TIB 97-1



NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 97-1

INTERNET PROTOCOL NEXT GENERATION (IPv6)

CLEARED
FOR OPEN PUBLICATION

JANUARY 1997 MAR 28 1997

19970516 017

RECTORATE FOR FREEDOM OF INFORMATION
AND SECURITY REVIEW (OASD-PA)
DEPARTMENT OF DEFENSE

OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM
701 SOUTH COURT HOUSE ROAD
ARLINGTON, VA 22204-2198

DTIC QUALITY INSPECTED 3

97-5-1029

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1997	3. REPORT TYPE AND DATES COVERED Final Report	
4. TITLE AND SUBTITLE Internet Protocol Next Generation (IPv6)			5. FUNDING NUMBERS DCA100-96-C-0038	
6. AUTHOR(S) Dale Barr				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Communication Technologies, Inc. (COMTek) 503 Carlisle Drive, Suite 200 Herndon, Virginia 20170			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Communications System Office of Technology and Standards Division 701 South Court House Road Arlington, Virginia 22204-2198			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NCS TIB 97-1	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Internet Protocol (IP) is the most widely-used method for transporting data within and between communications networks. It is as useful for the growing field of intranets (networks internal to an enterprise or organization and not connected to the outside world-e.g., a network used for classified processing) as it is for the geographically distributed, highly heterogeneous Internet. IP combines the functions of internode linking with those of links between physical networks to provide communications paths between nodes on different networks. IP provides a connectionless, unreliable, best-efforts packet delivery system. The concept of "best-efforts" delivery means that packets will not be discarded arbitrarily, without good cause. The cause of erratic packet delivery under a "best-efforts" commitment is normally exhaustion of resources or failure of a lower-level link or physical system. IP is called connectionless because it resembles the Postal Service or Western Union more than it does the telephone system. When a node using IP wishes to send a message to another such node, it simply sends the packet, properly addressed, analogous to mailing a letter or sending a telegram. In IP's connectionless design, every packet is treated completely independently from all others.				
14. SUBJECT TERMS Internet Protocol (IP) Internet Control Message Protocol (ICMP)			15. NUMBER OF PAGES 60	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASS	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASS	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASS	20. LIMITATION OF ABSTRACT UNLIMITED	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

NCS TIB 97-1



NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 97-1

**INTERNET PROTOCOL NEXT
GENERATION (IPv6)**

JANUARY 1997

**OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM
701 SOUTH COURT HOUSE ROAD
ARLINGTON, VA 22204-2198**

DTIC QUALITY INSPECTED 3

NCS TECHNICAL INFORMATION BULLETIN 97-1

INTERNET PROTOCOL NEXT GENERATION (IPv6)

JANUARY 1997

PROJECT OFFICER

APPROVED FOR PUBLICATION:



DALE BARR
Electronics Engineer
Technology and Standards
Division



DENNIS BODSON
Chief, Technology
and Standards Division

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. These systems constitute an important part of the overall infrastructure upon which the functioning of Government and National economies rely. This report addresses the Internet Protocol which is the most widely-used method for transporting data within and between communications networks. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
Attn: N6
701 S. Court House Road
Arlington, VA 22204-2198

The creation of this document
was performed under contract to the

Office of the Manager
National Communications System



Contract # DCA100-96-C-0038

by

Communication Technologies, Inc. (COMTek)
503 Carlisle Drive, Suite 200
Herndon, Virginia 20170

(703) 318-7212
(703) 318-7214 fax

INTERNET PROTOCOL NEXT GENERATION (IPv6)

A TUTORIAL FOR IT MANAGERS

Abstract

This document provides a tutorial on the Internet Protocol version 6, also referred to as Internet Protocol next generation or as IPng and now usually designated IPv6. The intended audience is the manager familiar with information technology in general, but not necessarily a specialist in data communications. The tutorial covers the most important characteristics of IP in general and specifically of IPv6 and addresses the issues considered important to managers of computer and data communications systems and services. The level of technical detail presented is more limited than that which would be significant to those who must implement applications and systems software to comply with and utilize the new standards. Although some comparison between IPv6 and its predecessor, IPv4, is inevitable, a discussion of the differences and transition issues is not covered in detail in this document.

Table of Contents

1. What Is IP?	1
1.1 <u>The IP Packet</u>	2
1.1.1 The IP Packet Header	3
1.1.2 The IP Packet Payload	5
1.1.3 ICMP (Internet Control Message Protocol)	6
1.2 <u>IP Routing</u>	6
2. What is IPv6?	9
2.1 <u>Definition, Background, History</u>	9
2.1.1 Growth—Address Space Exhaustion	10
2.1.2 Router Table Explosion	11
2.1.3 Other Protocol Constraints (Fragmentation, Control, Checksums, etc.)	12
2.1.3.1 Fragmentation Inefficiency	12
2.1.3.2 Control	13
2.1.3.3 Checksums	13
2.2 <u>The New Structure</u>	13
2.2.1 Addressing	13
2.2.1.1 Unicast Addresses	16
2.2.1.2 Multicast Addresses	16
2.2.1.3 Anycast Addresses	18
2.2.1.4 Address Assignment Concepts	19
2.2.2 Routing	20
2.2.3 Fragmentation	21
2.2.4 Checksums	22
2.3 <u>Autoconfiguration</u>	22
2.4 <u>Security</u>	23
2.4.1 Security Associations	25
2.4.2 Security Algorithms	26
2.5 <u>Quality of Service and Real-time Support</u>	27
2.6 <u>Programming Interface</u>	29
3. What Does the Telecommunications Manager Need to Know and Do?	30
3.1 <u>Identification of Software and Hardware Vendors</u>	30
3.2 <u>Personnel Training</u>	30
3.3 <u>User Awareness</u>	31
3.4 <u>Benefits of Transition to IPv6</u>	31
3.5 <u>Transition Planning</u>	32
3.5.1 From the "Simple Internet Transition Mechanisms" Internet Document	32
3.5.2 Interoperability	34
3.5.2.1 Dual-Stack Configuration	34
3.5.2.2 Tunnels	34
3.5.2.3 Domain Name Service	35
3.5.3 Managerial Actions	35
3.6 <u>The Bottom Line</u>	36

Internet Protocol Next Generation (IPv6) Tutorial

4. References	39
5. Appendices	40
5.1 <u>IPv6 Packet Header Format</u>	40
5.2 <u>Feature Comparison of IPv4 and IPv6</u>	41
5.3 <u>Glossary</u>	43
5.4 <u>Standards Information</u>	46

Figures

Figure 1. Comparison of ISO OSI to IP	2
Figure 2. The IP Packet (or Datagram)	3
Figure 3. IP Packet Payload Contents	5
Figure 4. Address classes in IPv4 (prior to CIDR)	7
Figure 5. Routing Hierarchy and Default Routing	7
Figure 6. IPng Standards Development Timeline	10
Figure 7. Packet fragmentation	12
Figure 8. Addressing for a multi-homed host	16
Figure 9. Multicast address format	17
Figure 10. Provider-based address format	19
Figure 11. MTU discovery and fragmentation under IPv6	21
Figure 12. Authentication Extension Header	24
Figure 13. Encryption Extension Header	25
Figure 14. Authentication computation	26
Figure 15. Comparison of ordinary and hierarchical encoding	28
Figure 16. IPv6 Tunneling Through IPv4 Routers	35
Figure 17. Transition process management	36
Figure 18. The IP transition bottom line	37

1. What Is IP?

IP is the Internet Protocol: a Connectionless, Unreliable, Best-efforts Delivery System
--

The Internet Protocol (IP) is the most widely-used method for transporting data within and between communications networks. It is as useful for the growing field of *intranets* (networks internal to an enterprise or organization and not connected to the outside world—e.g., a network used for classified processing) as it is for the geographically distributed, highly heterogeneous Internet. IP combines the functions of internode linking with those of links between physical networks to provide communications paths between nodes on different networks.

In more detail, IP provides a connectionless, unreliable, best-efforts packet delivery system. One should not be put off by the "unreliable" description; ensuring reliability is the responsibility of higher-level users of IP, such as TCP (Transmission Control Protocol). Unreliable delivery means that packets may be lost, delayed, duplicated, delivered non-consecutively (in an order other than that in which they were sent), or damaged in transmission.

The concept of "best-efforts" delivery means that packets will not be discarded arbitrarily, without good cause. The cause of erratic packet delivery under a "best-efforts" commitment is normally exhaustion of resources or failure of a lower-level link or physical system. In a highly-reliable physical system such as an Ethernet LAN, the "best-efforts" approach of IP may be entirely adequate for transmission of large volumes of information; as an example, Sun's widely-used Network File System (NFS) depends on little more than IP in such an environment and performs very reliably. In the geographically-distributed and highly diverse Internet, which is subject to many vagaries of operation, IP delivery is insufficient and must be augmented by a higher-level protocol (known as TCP, or Transmission Control Protocol) to provide satisfactory performance.

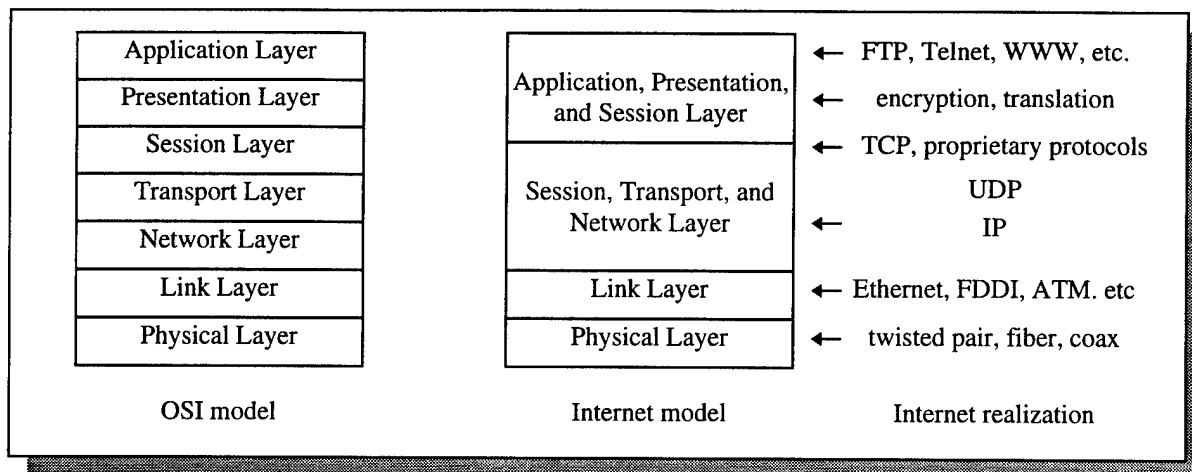
IP is called connectionless because it resembles the Postal Service or Western Union more than it does the telephone system. When a node using IP wishes to send a message to another such node, it simply sends the packet, properly addressed, analogous to mailing a letter or sending a telegram. (In fact, another name for an IP packet is a datagram.) The telephone system, on the other hand, creates a *connection* between two users which is maintained for the duration of the information exchange. Unlike the Postal Service, however, the services of IP can be used to create a connection-oriented operation mode, but this is the job of higher-level protocols and applications (such as TCP, File Transfer Protocol [FTP], Telnet, and others). In IP's connectionless design, every packet is treated completely independently from all others.

As the foregoing implies, IP is relied upon by higher-level protocols to create more complex services. The two principal protocols that make use of IP are UDP (user datagram protocol) and TCP (transmission control protocol). UDP is also unreliable and connectionless, like IP, but it introduces a further point of delivery, known as a *port*. A port is associated with a particular program or application that listens for information sent to that particular location. TCP also uses the port construct, but also moves beyond the concept of mere datagram (packet) delivery to provide a reliable connection between programming entities at network nodes, using packet

numbering and tracking to ensure delivery of all messages, or at least to account for their loss in the case where a connection is broken (e.g., a host crash). TCP accounts for the vast majority of traffic using IP in the Internet (but UDP, which is used for NFS, certainly dominates traffic on most LANs where the Network File System is in use) and often the two protocols are referred to together as TCP/IP. The ability of TCP to provide a reliable connection between two entities makes it the primary vehicle to supply services to applications, including Web browsers, Telnet, FTP (file transfer protocol), email, WAIS (Wide Area Information Service), gopher, Parallel Virtual Machine (PVM), and proprietary applications, such as the client-server database/inquiry systems from Oracle and Sybase.

Those familiar with the International Standards Organization Open Systems Interconnection (ISO OSI) 7-layer model may be concerned with the role of IP in that model. Since OSI was intended to define relationships in a network whose structural model is somewhat different from the Internet, the correspondence of the services provided by IP to the OSI concept is not quite exact. In particular, IP provides services which lie in both the network and transport layers of the OSI model. The higher-level UDP protocol overlaps the transport layer as well as the session layer (the latter because UDP uses ports), while the TCP protocol resides partly in the transport layer and partly in the session layer.

Figure 1. Comparison of ISO OSI to IP



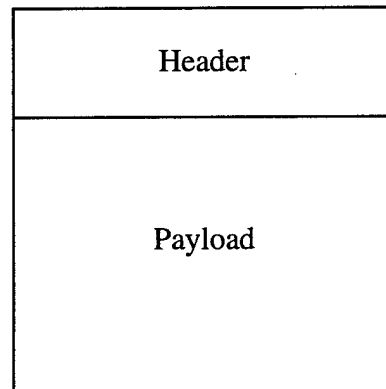
In this diagram, the above-mentioned overlaps are shown by a mingling of layers and functions. For example, the Internet world does not formally define a presentation layer as such, but in fact, presentation functions are present. However, the World Wide Web uses file suffixes (e.g., .txt, .html) to specify a file format (Multipurpose Internet Mail Extension, or MIME), which in turn determines how the contents will be viewed (or *presented*) to the end-user; although MIME types are interpreted by running an application function, the actual result is indistinguishable from a presentation-layer operation. Similarly, the behavior of TCP, in maintaining a connection, as well as in providing reliable end-to-end transmission, offers a session-layer kind of function in the ISO sense, as well as a transport function as ISO defines it.

1.1 The IP Packet

The IP Packet, or Datagram, is the fundamental unit of transmission

IP centers around the concept of a *packet*. A packet is also known as a *datagram*, although that term is also used in the context of a number of different protocols at different levels of communications architecture; the term datagram specifically implies that the protocol operates in connectionless mode, as described above.. Pursuing the Postal Service analogy used previously, an IP packet is comparable to a letter, although it is different in some important ways. An IP datagram consists of a *header* and a *payload*. The header is analogous to the envelope handled by the Postal Service, while the payload is analogous to the contents of that envelope. Note that the post office is not permitted to open envelopes or peek inside; its entire job can be done by looking only at the outside of the envelope. Similarly, IP can do its job based on the contents of the header alone (ordinarily, that is, with some exceptions for special handling cases, just as the post office may open envelopes to check for contraband or for information when an envelope is undeliverable as addressed and provided with no return address, causing it to end up at the dead letter office).

Figure 2. The IP Packet (or Datagram)



1.1.1 The IP Packet Header

The IP Packet Header provides addressing and control

The IP packet header begins with a version number in the first four bits. This is inflexible and will be maintained for all versions of IP, whatever else in the packet header changes. For IPv6, the value is, of course, 6, which distinguishes it from the value of 4 which appeared in IPv4 packet headers. This value determines how an Internet node, which may have the capability of processing more than one generation of IP formats (something that will be absolutely necessary

for a lengthy transition period to the next generation) will handle the bits that follow in the rest of the header and in the packet itself. The header also contains a *length* field, which describes the number of bytes (octets) in the IP packet. (The previous version of IP specified a total length, while the next generation IP specifies the payload length; the references do not make clear why this change occurred, but it may be that the fixed-length header of IPv6—the header length of IPv4 was variable—allows one to save a couple of instructions in the packet composition stage and improve efficiency.) An IP packet header contains a destination address and a source address. The source address designates the originating node's interface to the network, and the destination address specifies an intended recipient interface or possibly multiple recipients (multicasting). The source address plays a role in returning control information (using ICMP as discussed below) about packet transmission and network behavior to the originating location and also allows the destination node to reply to messages. Note that an address identifies an interface, and not a node itself; a node may have more than one interface to the net, each of which must have a different address. (See Figure 8. Addressing for a multi-homed host.) This can affect the efficiency of transmission, since the path to a node via one interface may be shorter than via another.

The IP packet header contains a counter which is used to limit the life of the packet on the network. We have all heard stories of the Postal Service delivering letters thirty years after they were mailed, because the letter was somehow caught in the system, such as by slipping behind a counter or being stuck in equipment supposed to be empty. If this were to happen with IP packets, however, they would consume resources of the system indefinitely, and it could be even worse than that if packets were to be duplicated repetitively and not be delivered—the entire Internet could be brought to a halt by a blizzard of reproducing but undeliverable packets. Thus, the counter is kept and decremented each time a packet moves through a routing step (a "hop"). If the counter ever reaches zero, the packet is discarded. By keeping the initial value small (and it is limited by the modest number of bits given to it), the possibility of the Internet experiencing a "Sorcerer's Apprentice" melt-down is prevented. The original design and intention for the counter was that it would represent an actual elapsed time limit, but in practice (and as recognized in the next generation of IP), it really counts down the number of times the packet is forwarded, which is much less effort to compute, an important criterion in the demanding environment of communications performance. (Some designers criticized the small size of the field, asserting that it limited the number of routing steps, but it was pointed out that any network connection requiring more than 255 routing steps would be hopelessly slow in performance, and the limit was retained.)

Another element of the IP packet header, but one not yet in significant use, is a designation for type of service. The developers of the protocol expected that this field could designate priority and other forms of special handling, but the Internet world has not generally made use of this capability. The intention to use the Web for real-time operations, such as multi-media, seems likely to bring about a change, probably as part of the transition to IPv6.

There are a number of other fields in the IP packet header, especially in the more complex IPv4 header, but their purpose is either superseded in IPv6 or else is too technically detailed to be of interest at the level of this tutorial. We may summarize the important points of the IP header in the following list:

- ♦ Version number
- ♦ Size
- ♦ Source and destination addresses
- ♦ Counter controlling lifetime of packet
- ♦ Service type
- ♦ Etc. (a group of less-important components)

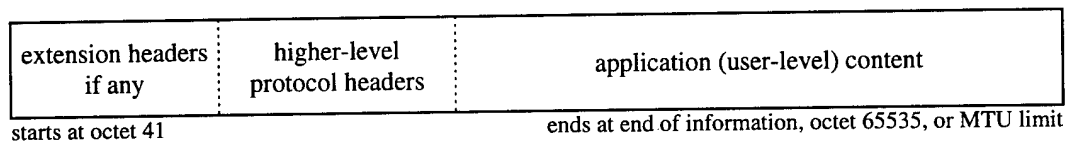
Note that these are not necessarily the names used in the IP definition documents, and that some of the names differ between IPv4 and IPv6. The terms used here are descriptive and are intended to convey the maximum amount of information for the audience of this tutorial.

1.1.2 The IP Packet Payload

The IP Packet Payload carries information and error/control protocols

What goes in the payload section of an IP packet? Virtually anything, up to the maximum size limit of 2^{16} (65536) bytes. IP itself does not examine the contents of the payload section in general—except that it may be used to contain further protocol information of a nature considered essentially part of the Internet Protocol itself, specifically Internet Control Message Protocol (ICMP). Usually, though, the contents of an IP packet's payload area will be information encapsulated in a higher-level protocol, such as UDP (User Datagram Protocol) or TCP, and the details of these are really beyond the scope of an IP tutorial, except to say that these allow the implementation of services actually employed by end-users. IP does check for the presence of such protocol contents, which are specified in the same way as ICMP, in order to determine what higher-level software is to be given the payload for further processing. One additional form of payload is routing protocol information, which is necessary for the IP software to be aware of means for conveying packets to their destination. The several types of routing protocol are discussed in more advanced literature.

Figure 3. IP Packet Payload Contents



1.1.3 ICMP (Internet Control Message Protocol)

ICMP is the control and management protocol for IP

ICMP provides the IP user (generally a higher-level protocol) with a means of learning about and dealing with errors and problems which occur in packet transmission and routing. Although

ICMP messages are carried in the payload area of the IP packet, ICMP is an essential part of the Internet Protocol itself.

ICMP provides message types which cover such faults as unreachable destinations, time limit exceeded (i.e., number of routing steps too large) and parameter problems. ICMP also allows a sender to query if a destination is in operation (echo request/reply), redirect a routing, control the volume of information sent by another sender (source quench), and request time-stamps for determining routing delays. Some of these classes of ICMP messages cover many additional types of information, such as, in the case of an unreachable destination, the information as to what caused the problem (network unreachable, host down, unknown host or network, etc.) This level of detail is of more interest to programmers than to managers; it suffices to point out that ICMP messages provide sufficient information to resolve most network problems and afford effective management of network systems.

An important point about ICMP error messages is that they provide detail not only about the IP packet that caused a problem, but also convey enough information to return error messages to higher level protocols and applications. This is necessary because the IP level often does not have enough knowledge to decide how to proceed in case an error is encountered; that awareness must reside at a higher level in the system.

1.2 IP Routing

Routing of IP packets is critical to performance, requiring planning and careful design

Consider postal ZIP codes. They are organized in a geographic hierarchy: The first digit specifies a region of the country, the next two digits a metropolitan area or sectional sorting center, the next two digits a local postal delivery district, and the last four an individual box cluster, side of a single block of one street, or single large customer, such as a corporation. (And there are even eleven-digit codes defined, which specify the individual delivery box or house and will soon be in use.) At each stage, it is clear where the next step in routing a letter is to take it.

The same has not been true of IP. Routing a packet is a difficult proposition. An IP address as originally designed specifies only a network and a node's interface on the network. (See Figure 4 below.) Some networks may have their own subnetworks, but those are not visible in the address and are handled within the network (via a locally-known subnet mask—a set of bits that identifies which part of an address specifies a subnetwork and which a host interface), rather than at the internetwork (Internet) level. Looking at an IP address provides no clue as to the path a packet destined for that address must take. An originating host with a packet to send will know that the packet not destined for the local network must be given to a router with access to the Internet (a gateway), but the router must figure out how to proceed from that point.

Figure 4. Address classes in IPv4 (prior to CIDR)

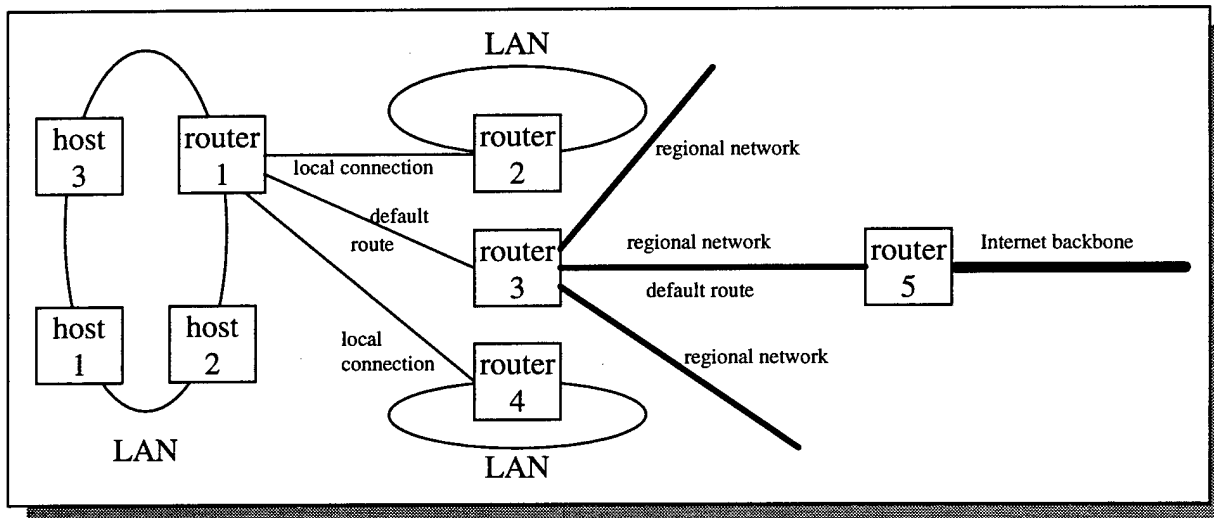
	8 bits	8 bits	8 bits	8 bits
--	--------	--------	--------	--------

Class A address	Network number	Interface number
Class B address	Network number	Interface number
Class C address	Network number	Interface number

To do this, the routers must keep information on where to send packets, based on the network address. It is not necessary to keep information on individual hosts; delivery to a router on the host's network is sufficient, as that router will know how to transfer the packet to the destination host, using any subnetworks if present. The information which associates networks with routers is kept in the *routing tables* at each router.

It would seem that every router must know about every network in the Internet in order to do routing, but this is not quite the case. It is possible for a *default route* to be defined, which will cause all packets not local to the originating network or to a network defined in the tables to be sent to a default router. It is this that establishes a start towards a hierarchy like that for the postal ZIP code structure; local routers need only know of a default router which has more global knowledge of routes. And that router may know of another default router with even wider information.

Figure 5. Routing Hierarchy and Default Routing



In this diagram, traffic within a single LAN never reaches a router; it is recognized at the host software level. Routers 2 and 4 need only send all non-LAN traffic to Router 1. Router 1 needs tables only for the hosts on the LANs served by Routers 2 and 4 and uses a default route to send all other traffic to Router 3. Router 3 must know about all subdomains in the regional networks to which it is connected (including the LANs served by Routers 1, 2, and 4). Router 3 then uses a default routing for all other traffic, sending it to Router 5.

But this limited hierarchy must have a top level. Routers attached to the backbone (very high performance long-haul circuits) of the Internet (e.g., Router 5 in Figure 5) really must have routing tables that cover every network in the world. Keeping these tables up to date is a prodigious job for network backbone administration, while the lookup process for such large routing tables is in conflict with the need for the backbone routers to provide the highest possible performance, consistent with their position on high-performance networks processing the largest volume of packets. As a stopgap measure, CIDR was introduced within the past couple of years to stave off problems with router table size and address space exhaustion; the following definition is from the Frequently Asked Questions file on CIDR (available via the Internet):

CIDR stands for Classless Inter-Domain Routing and is documented in RFCs 1517, 1518, 1519, and 1520. CIDR is an effective method to stem the tide of IP address allocation as well as routing table overflow. Without CIDR having been implemented in 1994 & 1995, the Internet would not be functioning today.

Basically, CIDR eliminates the concept of class A, B, and C networks and replaces this with a generalized "IP prefix". CIDR can be used to perform route aggregation in which a single route can cover the address space of several "old-style" network numbers and thus replace a lot of old routes. This lessens the local administrative burden of updating external routing, saves routing table space in all backbone routers and reduces route flapping (rapid changes in routes), and thus CPU load, in all backbone routers. CIDR will also allow delegation of pieces of what used to be called "network numbers" to customers, and therefore make it possible to utilize the available address space more efficiently.

CIDR provides a temporary solution to present design problems, but its introduction and success have inspired some of the design of the next generation of the Internet Protocol.

2. What is IPv6?

"IPng is a new version of IP which is designed to be an evolutionary step from IPv4. It is a natural increment to IPv4. It can be installed as a normal software upgrade in Internet devices and is interoperable with the current IPv4. Its deployment strategy is designed to not have any flag days or other dependencies. IPng is designed to run well on high performance networks (e.g. ATM) and at the same time is still efficient for low bandwidth networks (e.g. wireless). In addition, it provides a platform for new Internet functionality that will be required in the near future." (*From the IP next generation home page on the Internet—<http://playground.sun.com/pub/ipng/html/ipng-main.html>*)

(A "flag day" is an expression referring to a specific date in a transition when all operations must simultaneously shut down and convert to a new operating mode. The disruption caused by such an event is considerable, and, in the case of a world-wide Internet, something that absolutely must be avoided because it would simply be impossible to co-ordinate.)

2.1 Definition, Background, History

The IPv6 definition is the result of many months of effort by the leading experts in networks and communications
--

IPv6 is the designation for the new version of the IP standard. It was previously known as IPng, standing for IP next generation. There were several reasons for the development of a new standard in the face of a current working and widely used system. Most important was the exhaustion of the 32-bit address space of IPv4. Although 32 bits provides a possible 4 billion addresses or so, the existence of reserved address ranges, structure of network addresses within those 32 bits, and other considerations (such as the practical fact that only a fraction of the addresses—see Huitema's discussion—can be allocated effectively) led to a forecasted crisis in address availability for the early 2000s, if not the late 1990s. There were other problems of a more technical nature whose impact would fall primarily on performance, without creating the problem of total inaccessibility for those who needed new addresses, but which needed to be dealt with very soon if the Internet was to continue to expand without slowing to an unusable crawl.

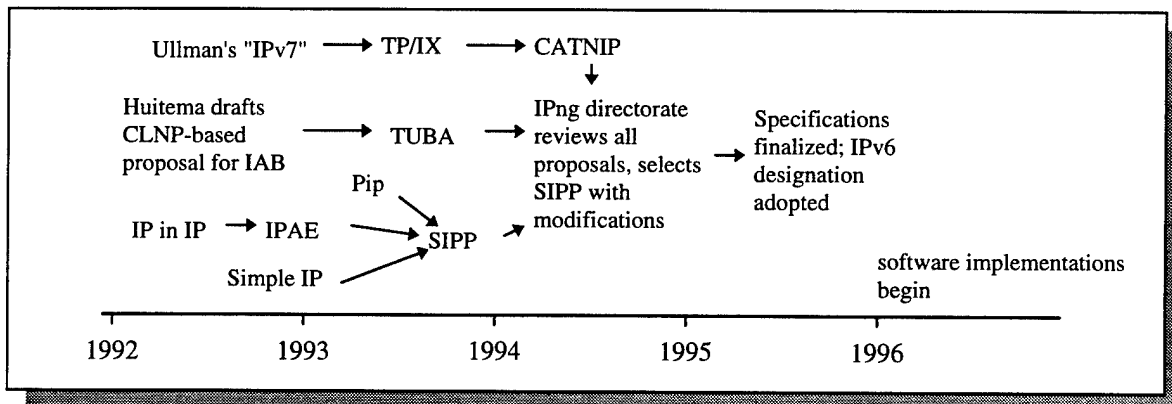
The development of the new standard for IP began in the summer of 1992 following the first congress of the Internet Society. The Internet Activities Board (IAB) looked at a new standard drawn up by Christian Huitema based on the Connection-Less Network Protocol (CLNP) of the International Standards Organization (ISO). Premature publicity and politics, along with the grizzled age and lack of market success of CLNP, led to a considerable uproar over whether such a drastic step as definition of a new IP standard should be undertaken without much wider discussion. The contretemps led to a reorganization of the various bodies contributing to the decision-making process under the Internet Engineering Task Force (IETF), along with the

creation of a number of teams competing to create a better approach for the new standard. A directorate was also created to evaluate the various proposals.

From 1992 to 1994, the competing proposals were reviewed, extended, and defined more precisely. They included such ideas as TUBA (TCP and UDP over Bigger Addresses—the CLNP-based concept), TP/IX (which comprised changes to TCP as well as to IP) that later became CATNIP (Common Architecture for the Internet—a common packet format for IP and CLNP, but which failed to become fully ready in time for a decision on IPng to be made), and SIP (Simple IP), which later merged with another idea to become Simple IP Plus, or SIPP.

The IPng directorate reviewed these proposals in the summer of 1994 and selected SIPP, with some modifications, as the winning proposal. Since the term "IPv5" had already been used for an experimental real-time streaming protocol, the new designation for IPng was IPv6. Another year would pass before the final version of the specification would be agreed upon and published.

Figure 6. IPng Standards Development Timeline



The details of the existing situation that led to the desire for and creation of the new standard are explained in the following sections.

2.1.1 Growth—Address Space Exhaustion

The IP address space, designed for a limited-size Internet, is running out of room for new networks

The designers of the original Internet Protocol lived in the 1978 world of mainframe computers and terminals, a world in which only a few thousand computers and a few dozen networks existed. The primary applications for network data transmission were exchange of bulk data (payroll records, seismic logs, telemetry) and transactions (airlines reservations, point-of-sale). They did not foresee that in less than two decades, computers would cost less than used cars and fit onto desktops, nor that an Internet would connect tens of thousands of enterprises, government

entities, and educational institutions, with electronic mail, software distribution, and multi-media information being used in millions of homes and offices. (Note that the telephone industry is experiencing a similar problem with the proliferation of cellular phones, fax machines, and computer modems, which has led to telephone numbers and area codes that look strange by the standards of twenty years ago—and difficulty in making sure that existing hardware can communicate with the new numbers. Even the Postal Service has gone from five-digit ZIP codes when originally introduced in 1963 to nine digits, with eleven digits already designed and planned for introduction soon.)

Part of the effort of defining a new standard included an attempt to identify the future size of the Internet, assuming that growth would continue, not only in the number of individuals using it, but also in the number of computers per individual (many Americans already have multiple computers at home and at work, and integration of chips into other devices, such as automobiles and home appliances, has become widespread in the US). The working assumption of a hundred computers per human being was given a safety margin, and the final estimate to be used in developing an addressing specification was 1 quadrillion (10^{15}) computers connected by 1 trillion (10^{12}) networks. The designers took into account the inefficiencies in address assignment and came to the conclusion that new addresses should be somewhere between 57 and 68 bits in size.

Given the above, it appeared that 64 bits (8 octets or bytes) would be sufficient to handle addressing for the foreseeable future. Part of the process that evolved SIPP into IPv6 changed the 64-bit addresses to 128-bit addresses. Under Huitema's most pessimistic assumptions about address assignment efficiencies, 128-bit addresses will provide for hundreds of computers per square meter of the earth's surface. The problem of address space exhaustion should be solved by IPv6 for the foreseeable future.

2.1.2 Router Table Explosion

Routing requires tables which have grown unmanageably large
--

As was discussed above, in section 1.2 (IP Routing), the need to keep track of routing paths can be finessed for individual networks through the use of default routings which automatically send packets intended for remote destinations to a more knowledgeable router. But some routers, specifically those serving the long-haul, high-performance backbone networks of the Internet are required to know everything about the Internet's structure. As networks continue to be added to the Internet, the tables describing it grow by the scores daily. This phenomenon, known as router table explosion, was one of the three possible ways the Internet could collapse (the other two being related to the shortage of available addresses, as discussed previously). Under IPv4, a technique known as Classless Interdomain Routing (CIDR) is being used to cope with the router table explosion resulting from the enormous expansion of the past three years. (The World Wide Web barely existed in 1993; in 1996, there are close to 100 million Web pages. The desire to provide Web service has led to the creation of many more IP addresses than would have been the case if most users had continued to be primarily consumers of information through the Internet.)

The goal of CIDR is to replace the IPv4 concept of network classes (IP addresses were assigned, as shown above in Figure 4, to classes A, B, and C, respectively, based on the anticipated size of the network, allowing a very few enormous networks and many relatively small ones—which has led to address space exhaustion for certain classes) with a routing structure based on creating a hierarchy of routers and nets within the Internet. (Recall the discussion of ZIP codes in section 1.2 and remember that there was no more than a rudimentary hierarchy created by default routings.) Unfortunately, because the groupings used by CIDR were not based on geography—and could not be, given the nature of how Internet service providers expand their business without regard to geographical or political boundaries—the hierarchical structure of CIDR still encountered limits. But it provided a design concept for IPv6, and IPv6 does not suffer from the original constraint of having network classes in the first place, which CIDR had to overcome.

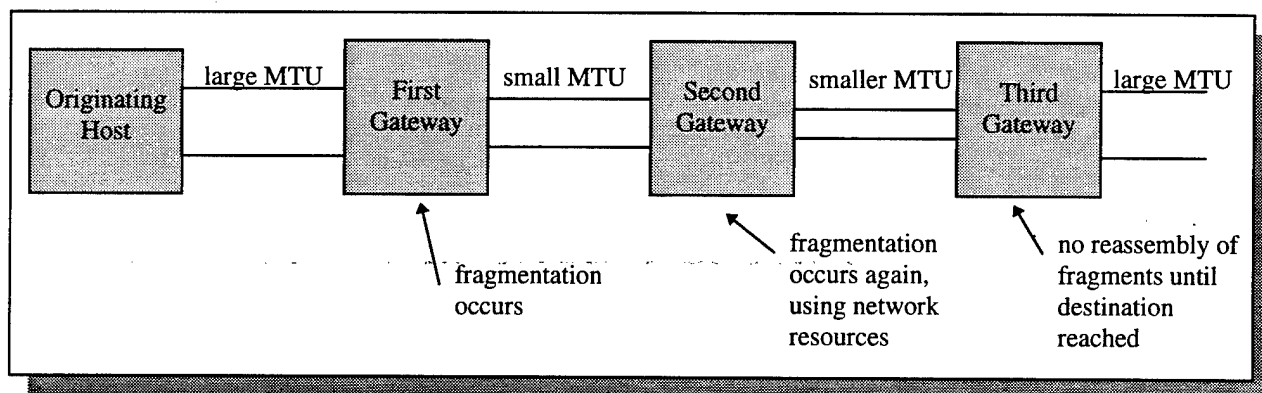
2.1.3 Other Protocol Constraints (*Fragmentation, Control, Checksums, etc.*)

2.1.3.1 Fragmentation Inefficiency

Fragmentation-on-demand in the present IP has proved to be inefficient

The limit on packet size is called Maximum Transfer Unit (MTU) and may vary as a packet travels through the Internet, encountering different physical transmission links and software restrictions. (For example, the Ethernet LAN protocol limits packet size to 1518 octets, including its own header and CRC.) The present system fragments packets on a demand basis; that is, whenever a packet encounters a gateway whose MTU is smaller than the packet size, fragmentation occurs. Thus it is possible for fragmentation to occur more than once during a packet's trip through the Internet, while re-assembly does not occur until the fragments arrive at the final destination. This leads to considerable inefficiency in transmission, as the demand on the router systems that must fragment the packets as they travel takes compute resources away from the primary task of the routers, which is, after all, to route packets, not tinker with them.

Figure 7. Packet fragmentation



2.1.3.2 Control

Some control features of the present IP are no longer used or needed

The ICMP specification as implemented in the existing Internet Protocol depends on carrying control and error messages in IP packets. A number of types of messages are defined, but some of those in the original specification have proved to be of limited value; implementations have diverged from the standard by providing support for the ICMP types in an inconsistent manner. This has an impact on network performance at each routing step in a heterogeneous environment, which is certainly characteristic of the global Internet itself. Also, as Huitema points out, support for options requires code to be present in critical sections of the routing algorithms, leading to a desire for software implementors to dispense with options in order to improve performance, which accounts for some of the divergence from the standard.

2.1.3.3 Checksums

The requirement for checksums in current IP impacts performance but provides little benefit

The present Internet Protocol calls for the carriage of checksums within IP packets, and therefore their computation at each routing step (because the header changes at each step due to modification of the time-to-live field, which requires a new checksum). This computation must be done for every packet at every routing step, and the total expenditure of computing resources on checksumming in an Internet carrying trillions of packets a day is considerable. Some vendors have disabled this feature in order to gain a performance advantage over the competition, which makes the value of the feature questionable in any path that includes a router from such a manufacturer. Hence, the subject of checksums is one of the areas providing opportunities for improvement in the future.

2.2 The New Structure

2.2.1 Addressing

IPv6 increases address space by factors of quadrillions times quadrillions

The change in IPv6 most visible to the end-user and requiring action by systems administrators and managers is the expansion of the size of IP addresses from 32 bits to 128 bits. Concomitant with this change is the elimination of the address class structure that was found in the previous version of IP. A critical fact to be observed is that the present 32-bit IP addresses may be accommodated in IPv6 as a special case of IPv6 addressing. This means that re-numbering of existing sites can be postponed for a considerable period (until the local address space is no

longer adequate), and that therefore, some of the transition to IPv6 addressing can be done almost without disruption to parts of the Internet visible to the end-user.

IPv6 addressing requires new syntax to express the 128-bit addresses; the previous system of four decimal numbers separated by dots (".") was designed to represent 32-bit numbers only. The new syntax uses the colon (":") to separate groups of four hexadecimal digits. Although it might appear that eight such groups would be necessary, the designers expect that significant portions of new addresses will be zeros, at least at first, and they allow one group of consecutive zero double-octets to be represented by a double colon ("::"). The numbers preceding a double colon are assumed to be at the left-hand end of the 128 bits, and the numbers following at the right-hand end. (Clearly, only one such string of zeros may be so represented, to avoid ambiguity.) There are a few other forms of shorthand representation, such as for IPv4 addresses contained in IPv6 format. But most users should not have to deal with any of these representations, just as they do not commonly use the dotted-decimal notation of IPv4 now; most Internet addresses are represented symbolically (e.g., `sampson.cacr.caltech.edu`), rather than numerically, and that will continue to be the case for all but those involved with direct maintenance and configuration of network software. (There is a subtle problem which has not been solved yet; in the syntax of Uniform Resource Locators [URLs] used in World Wide Web applications, the colon is already in use to separate an IP address from a port number used by the higher-level TCP protocol. This usage syntactically overloads the colon, leading to ambiguity in a URL specification which employs an IPv6 numeric address specification; resolution of this difficulty is still being discussed.)

The IPv6 address space is truly vast; 2^{128} is nearly 10^{39} , a number so big we don't really have a convenient name for it—call it a million quadrillion quadrillion. This is many more than the number of nanoseconds since the universe began (somewhere around 10^{26}), or the number of inches to the farthest quasar (about 10^{27}); it's about the number of protons in 100 million metric tons (one quadrillion liters—enough for several hundred thousand Zeppelins) of hydrogen. Even if addressing assignments are truly inefficient, it seems unlikely that the IPv6 address space will be exhausted in the near future. With so much space to work with, the designers have been able to partition the address space in a number of ways to provide services and address assignments that were not available previously.

IPv6 has discarded the previous concept of network classes. This feature of IPv4 was already in trouble as address space became exhausted, leading to the stopgap solution of CIDR. But it has added a number of additional groupings and address types, some of which were at the research stage when the need for a new protocol version became apparent. These new types will require training for network administration personnel, but will provide benefits in more efficient utilization of the Internet and in structure of local domains.

The large IPv6 address space has been extensively partitioned on the basis of values in the first eight bits (ten in a couple of cases). The following table describes the various restrictions on addresses by prefix:

Table 1. IPv6 Address Space Allocation

Allocation	Prefix (binary)	Fraction of Address Space
Reserved (includes IPv4 compatibility addresses, which are zero in the first 96 bits)	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Reserved for IPX Allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Unassigned	001	1/8
Provider-Based Unicast Address (i.e., most addresses)	010	1/8
Unassigned	011	1/8
Reserved for Neutral-Interconnect-Based Unicast Addresses (geographic-based)	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link Local Use Addresses	1111 1110 10	1/1024
Site Local Use Addresses	1111 1110 11	1/1024
Multicast Addresses (both permanent and transient)	1111 1111	1/256

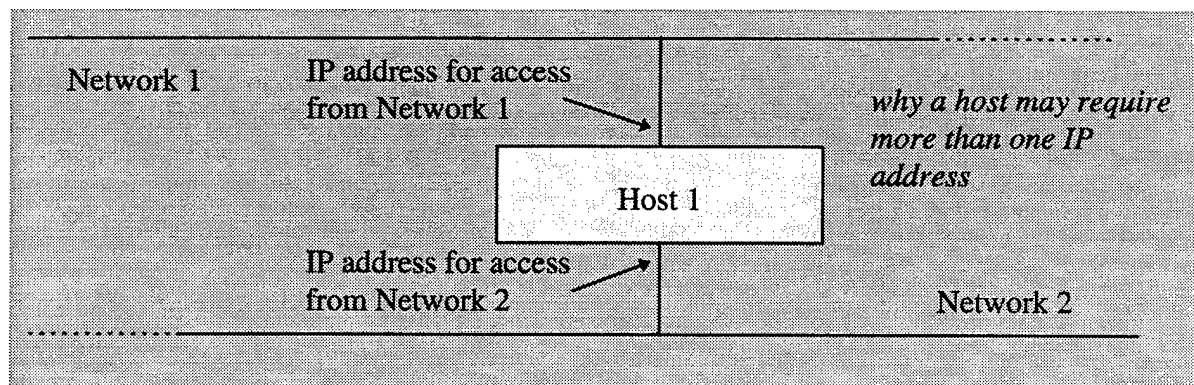
Even though the address space for a class may be as little as $\frac{1}{1024}$ of the total address space, the IPv6 address space is so huge that that allocation still provides for close to a thousand quadrillion quadrillion addresses. No subgroup is likely to be exhausted in the near future, and many reserved groups of addresses remain.

2.2.1.1 Unicast Addresses

A unicast address identifies a host's interface to the network

A *unicast address* is the basic type of address used to send a message to a single destination. (Sometimes terminology makes the simple sound complex.) These addresses were once known as point-to-point addresses. A unicast address is unique (within its scope of validity, since some may be local to a site and can be re-used at other sites). A particular host may have more than one unicast address, however, since a host may be attached to more than one network through different LANs and gateways. (In other words, a unicast address identifies, not a host itself, but that host's interface to the network, which is a very important distinction, although one often blurred in the minds of end-users.) Unicast addresses form the largest number and most familiar category of IP addresses, and there is little changed in IPv6 from previous versions of the Internet Protocol in terms of how such addresses are used. Most addresses visible to and used by end-users will be unicast addresses.

Figure 8. Addressing for a multi-homed host



2.2.1.2 Multicast Addresses

Multicasting provides a logical, rather than physical form of broadcasting

The *multicast address* provides a capability to replace the original concept of packet broadcasting. Some early designs to support the multicasting paradigm were developed in the early 90s, but they were not widespread. The design of IPv6 makes sure that all nodes which support the new protocol will have the capability.

Whereas broadcasting was a simple facility for sending a packet to all hosts on a particular network through a defined network broadcast address, multicasting provides considerably more sophistication. In a broadcast environment, there is no selection of which hosts on a network receive a packet; they *all* do. Under a multicasting discipline, it is possible to choose which

hosts will receive packets sent to a multicast address. (Consider the difference between a letter carrier delivering a piece of junk mail to every address on his route vs. a magazine delivering a copy only to subscribers.) One might characterize multicasting as a form of logical broadcasting, where the recipients are not necessarily defined entirely by a single physical connection.

To be sure, there is a price to be paid for this more powerful tool, and it comes about in address assignments. In the old broadcasting method, there was only one address to be used for the destination of a broadcast packet, and that was the network id concatenated with the reserved address of all 1 bits. (There was also a local broadcast address of 32 bits of all 1's, intended to be used during system setup procedures before a host had finished determining its IP configuration, but the protocol required that a proper net broadcast address be used after that initialization period was completed.) Often, the network hardware (e.g., Ethernet) had built-in capability that was used when a local broadcast address was recognized; this efficiency can still be used in the multicast environment when a multicast group is defined to be equivalent to an old-format broadcast network pool of recipients.

Multicasting requires a more complex form of address management and assignment. For multicasting under the old version of IP, a new protocol was introduced, Internet Group Management Protocol (IGMP). Since not every router or host had software to support IGMP, the use of multicasting was not universally available. IPv6 incorporates the multicasting concept into all nodes which support the protocol, and all nodes therefore are able to participate in the address management and assignment steps required to use the multicasting facility. In IPv6, the functions of IGMP are incorporated into the basic ICMP control protocol. The multicast address looks like this:

Figure 9. Multicast address format

8 bits	4 bits	4 bits	112 bits
0xFF	flags	scope	group ID

The hex value of FF in the first 8 bits identifies this as a multicast address. The other values are:

- flags 0 for a permanently assigned group (worldwide); 1 for a transient multicast group; other bits are undefined
- scope various values to limit transmission of packets to nodes, links, sites, or organizations, so that network congestion can be prevented
- group ID a unique identifying value (temporary or permanent) identifying the participants in a multicast

Group IDs may be permanently assigned on a worldwide basis; these will have a flag value with 0 in the lowest bit. Or they may be allocated on request for transient multicast operations (such

as a video distribution). Certain group IDs have reserved meanings: 0 may never be used, 1 is all nodes, 2 is all routers, and 0x10000 refers to dynamic host configuration servers. All of these reserved values should be limited to node or link scope. Every IPv6 node is expected to be able to process these reserved values. Other group IDs have been given meanings, although they are not part of the IPv6 protocol definition; e.g., group ID 43 refers to all NTP (Network Time Protocol) servers; the scope field determines which NTP servers will actually see a multicast packet with this group. Note that the concept of scope requires routers to be aware of the addresses belonging to scopes such as site or organization. Also, link scope with a group ID of 1 is equivalent to the old broadcast concept and can make use of the Ethernet or other link hardware broadcast facility for efficiency; routers (and even hosts) will be expected to have this efficiency built into their software.

ICMP messages are used to request transient group IDs, determine group membership, and terminate participation in a multicast group. Transient group IDs are assigned by random number generators to minimize conflicts, while a collision-detection algorithm is used to ensure that a new group ID is not in use. Further details are beyond the scope of this document.

Multicasting continues to be a topic for research. Fine-tuning of transmitting and receiving to subsets of a multicast group is being studied, as are concepts of routing control. The design of IPv6 ensures that any advances in the state of the art for multicasting will be done in an upward-compatible manner, so that current implementations will continue to work (but may not benefit from any performance gains that new techniques provide).

2.2.1.3 Anycast Addresses

Anycasting increases efficiency by using the Internet's built-in redundancy

Anycasting and anycast addresses are new concepts in IPv6. Researchers had begun to explore methods for anycasting under IPv4, but there was no obvious way to implement this feature within the limitations of that protocol. The new capabilities of IPv6 allowed the inclusion of this capability.

The purpose of anycasting is to take advantage of the natural redundancy that must be built into the Internet and its subnetworks. Because a failure in any single point of service cannot be allowed to bring down the Internet or disconnect a portion of it from the rest, there are multiple routing and service points for all major capabilities of the Internet. For example, domain name service is provided by at least two nodes in all cases, and all but the smallest organizations will have more than one path to the Internet backbone. An anycast address may be used to specify that a request made to that address can be serviced by any *one* of a multiplicity of routers or nodes.

For example, when a domain name service request is made, an anycast address could be used to specify any name server. Unlike a multicast or broadcast, however, only one name server (the nearest) should receive the request and respond to it. Another application of anycasting would be

"fuzzy" routing, in which a packet may be addressed to or via any router of a particular subnetwork.

An anycast address is indistinguishable from a unicast address, although it seems likely that some special pattern, such as a subnet ID followed by all zeros may be used for anycast addresses. At the time the IPv6 protocol was being defined, anycasting was very much a research topic. Implementations of IPv6 which will provide full anycasting support are still being defined and designed, so it is not known yet how this capability will work in practice. It is expected to provide additional efficiency in programming solutions and in obtaining better network performance, although IPv6 utilization can proceed without depending on it at first.

2.2.1.4 Address Assignment Concepts

IPv6 address assignment more efficiently conforms to the hierarchical structure of the Internet

The previous IP depended initially on a division of addresses into network classes, with network identification matching octet boundaries. As difficulties developed with this concept, additional addressing mechanisms, such as subnets were introduced. When this reached its limits, a concept called CIDR (Classless Inter-Domain Routing) was developed to provide ways to route packets without depending on network class. The experience with CIDR, which attempts to impose a hierarchy on IPv4 addresses that somewhat correspond with the topology of the Internet (as was discussed earlier), is being used as the basis for development of address assignment under IPv6.

Unicast address assignment is the primary problem; multicast and anycast addressing are either handled by another means or are subordinate to the unicast assignment method. The principal choice of address assignment methodology in IPv6, which provides at least some mapping between the addresses and the topology, as is essential to make coping with growth possible, is by service provider. Generally, a service provider's network will be a topologically cohesive subnet, which means that a single routing table entry can be used for *all* nodes served by that provider. The format of provider-based addresses is provisionally being given by the following:

Figure 10. Provider-based address format

3	m	n	o	p	125-(m+n+o+p)
010	registry	provider	subscriber	subnet	interface

The size in bits of the fields denoted by m, n, o, and p are still being determined, although the current plan has m=5 and the others multiples of eight, as described subsequently. The value of 11000 in the registry field denotes the familiar North American InterNIC in Reston, VA. 01000

denotes RIPE NCC in Europe (Réseaux IP Européens), and 10100 is Asia's APNIC. Each of these registries will issue provider IDs or may have ranges of provider IDs to be allocated by national subregistries; the current plan is for 16 bits to be used for provider IDs with the next 8 bits zero. Subscriber IDs are issued by providers and are planned for 24 bits, with another 8 zero bits allowed for future expansion. Providers may issue subscriber IDs on a basis which reflects their own internal network structure. This leaves 64 bits to specify the subnet structure of the individual subscriber and the node's interface; these will probably be partitioned into 16 and 48 bits, respectively, but a subscriber may even have more than one level of hierarchy itself. This allows for approximately 32 registries (about 8 times as many as currently exist), over 65,000 providers per registry (the eight bits left for expansion just may be needed), almost 17 million subscribers per provider (phone companies have more subscribers than this now, so the expansion bits may well be used eventually), over 65,000 subnets per subscriber (surely enough), and nearly 17 million interfaces per subnet (definitely enough).

There are a couple of glitches in this rosy scenario: First, the necessary redundancy (alluded to in the discussion on anycasting in section 2.2.1.3) means a service provider of any size will have more than one connection from its subnet to the Internet; this means that multiple routing table entries are going to be required. Second, service provider addressing has the potential to lead to commercial or governmental monopolization, as one of the designers pointed out. To allow for an alternative approach to addressing and routing on a truly geographic basis, a set of *neutral*, or geographic-based, addresses is also furnished in the addressing scheme. How the neutral addresses will be allocated and under what circumstances remains to be defined.

Finally, the addressing scheme allows for some special formats. We have seen that IPv4 addresses are accommodated as special IPv6 addresses with the first 96 bits zero. The address of all zeros is used as an origin address (but *never* as a destination address) by a node that is in the initialization process and has not determined its own correct IP address yet, while the address of 1 (i.e., 127 0 bits followed by a 1) is the *loopback* address used for testing; a packet with the loopback address is never sent onto the network at all but is turned around within the IP software on the node itself.

The remaining special address formats are for site-local and link-local use. These are intended to be employed by sites which wish to use IPv6 technology but need not be connected to the global Internet (e.g., sites engaged in classified processing). Their names indicate their scope; they are guaranteed unique only within that scope. Further, a packet with a link-local address is never to be transmitted by a router at all; it stays within a particular LAN. These two formats are distinguished by specific 10-bit address prefixes as shown in the table above.

2.2.2 Routing

IPv6 routing is based on the proven CIDR concept

Routing is a highly technical subject. The discussion of address allocation indicates that IPv6 contains methods for making routing of packets considerably more efficient than they have been under the current IP definition. Hierarchical structures of addresses will limit the router table

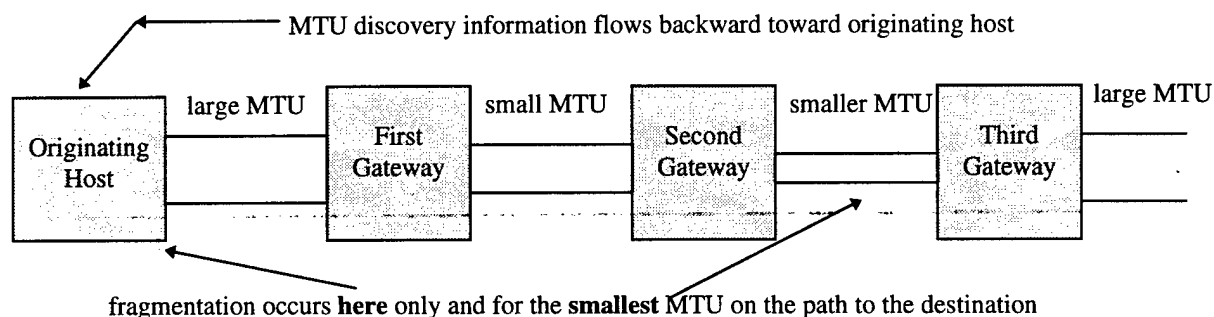
explosion problem, while new forms of addresses, such as anycasting and multicasting will make use of network capacity in a more efficient manner. The actual techniques involved in route selection within domains (OSPF, RIP) and between domains (EGP, BGP, IDRP, CIDR) are well beyond the scope of this document. They are largely invisible both to managers of network sites and to the end-users of network facilities because their implementation resides within the purview of the network software providers themselves, but the system administrators and network programmers at a site will need to know some of the details of these services, as local configuration tables must reflect paths to the Internet. However, it can be said that the success of CIDR in preventing the collapse of the Internet, starting in 1994-5, has been the basis for the development of the routing capabilities under IPv6, which are founded on the concept of provider-based address assignment as discussed previously. Building on a method which has been proven in practice leads the IPv6 designers to believe that the new system will be efficient and reliable when it is introduced.

2.2.3 Fragmentation

IPv6 requires fragmentation only once—at the originating host, not at intermediate routers

IPv6 deals with the problem of varying physical network packet sizes in a new way: IPv6 provides an *MTU discovery* mechanism, rather than performing fragmentation at the point in a packet's trip through a network where it encounters an MTU smaller than its own size. This means that any packets which encounter an MTU too small to let them through are *discarded*, with an appropriate notification (through ICMP) back to the sender. Since the sender is able to determine the total path MTU and is therefore responsible for doing so, the sender does all fragmentation initially, and all packets are processed through the network without delay at routers, there being no overhead of intermediate fragmentation (and no need for gateway software to provide fragmentation services). The IPv6 specification does provide for fragmentation information in extension headers, but that information appears as payload to routers along the path; it is only of interest at the source and destination nodes.

Figure 11. MTU discovery and fragmentation under IPv6



2.2.4 Checksums

Checksums are eliminated from IPv6 and some error control is relegated to other protocols

Checksums have been eliminated from the IPv6 packet. This seems risky at first glance, but the task force, on closer examination, found that the types of errors best prevented by checksums of IP headers were extremely infrequent, while the errors most likely to occur were faults within routers themselves, where the checksums were being computed. Thus, an error could occur and still have a correct checksum. Most other kinds of errors in the new header would result either in packet loss (to be dealt with by the higher-level protocol) or a delay in processing at worst. Thus there seemed to be no need to maintain checksums at the level of the IP header.

2.3 Autoconfiguration

**IPv6 reduces the burden on system administrators
by allowing networks to be configured automatically**

One of the most important new capabilities provided by IPv6 is that of automatically configuring the network addresses of equipment newly added to a network. The concept of "plug and play" has been popular in the personal computer world since the introduction by Apple Corporation of the Macintosh, which required negligible effort to set up a networking operation, and which has since been adopted by Intel under Windows 95. Since the Internet developed among mainframe computers and workstations, where highly-trained technical staff were available to develop configuration designs and files, the first protocols simply did not include autoconfiguration in their vision of the capabilities required.

Two consequences of the Internet explosion are the purchase of Internet-capable equipment by non-computer professionals (what Huitema refers to as "the dentist's office") and the sheer volume of systems requiring attachment to the network (which Huitema calls "the thousand computers on the loading dock"). In the former case, the end-user does not know the internals of the Internet Protocol needed to configure his system; in the latter, there is simply not sufficient staff to accomplish the configuration steps manually in reasonable time.

Some basic capabilities have already been in use for dealing with these problems. They include the Address Resolution Protocol (ARP) for determining a physical address from an IP address, the Reverse Address Resolution Protocol (RARP) for determining a node's own IP address during initialization, the Bootstrap Protocol (BOOTP) for a similar purpose, and the Dynamic Host Configuration Protocol (DHCP) to provide a basic allocation and release of IP addresses on an as-needed basis. However, these protocols are in addition to the Internet Protocol itself, and therefore, their availability and use are not consistent across the Internet, similar to the situation described in sections 0 and 2.2.1.3 on anycasting and multicasting capability.

IPv6 resolves this by making autoconfiguration capability part of the basic protocol. The addressing scheme, by including link-local addresses and multicasting addresses, provides the tools for allowing a node to configure itself.

Physical autoconfiguration. The simplest example is that of a node on a pure LAN, not connected to the Internet (which might be the case for the dentist mentioned above). Most LANs today use an addressing scheme descended from the Ethernet design known as IEEE-802 addresses. In an IEEE-802 LAN, each interface has a unique 48-bit number associated with it; for Ethernet, the Xerox Corporation ensures that each Ethernet interface number is unique in the world (48 bits provides more than a quarter of a quadrillion addresses, so we will not run out of these any time soon). The IEEE-802 address is then used as the lower 48 bits of a 128-bit IPv6 link-local address (note that 48 bits cannot be used with a 32-bit IPv4 address) to form an address that is guaranteed unique within the LAN. The address will begin with the prefix 0xFE80, which denotes link-local addresses in the IPv6 addressing scheme. (If the LAN does not conform to IEEE-802, some other method may be used to generate a 48-bit unique address, such as a computer's serial number; even a random number may be chosen—for a LAN with a thousand nodes, the odds are hundreds of billions to one against two nodes accidentally selecting the same 48-bit number at random.)

Stateless autoconfiguration. For LANs which connect to the larger Internet, a more sophisticated system must be used. IEEE-802 addresses still can play a role in defining network addresses without resorting to servers. By using multicasting and addresses for the all-nodes and all-routers in the local context, a node can negotiate an address for itself by agreement with all participants in the local context, even in the presence of routers connecting several LANs. This is known as *stateless* autoconfiguration. The details involve a special ICMP message known as *router advertisement*.

Full autoconfiguration with servers and states. Unfortunately, stateless autoconfiguration has its limits; it uses address spaces inefficiently and does not easily allow for hierarchies of subnetworks (because the IEEE-802 address of 48 bits uses up too much of the addressing available for subnets). The final alternative is *stateful* autoconfiguration, which makes use of servers to define and resolve addresses. (This is also the method to use if one is concerned with security; one does *not* want to use a pure plug-and-play approach when there is the remotest possibility that an intruder could plug his machine into the net and become "just another user" with access to the system.) Various mechanisms, including those above, along with others, such as neighbor discovery, path discovery, etc. allow configurations and routing tables to be kept current in this complex environment.

2.4 Security

IPv6 provides new capabilities for increased security of data transmission on the Internet

In the present design of the Internet, security considerations have been relegated to upper-level protocols and applications. The original design of IP did not contemplate methods for providing secure communication. (It may be noted that the ISO OSI definition originally placed security at

the presentation layer, which is well above the network/transport layers where the functions of IP reside, remembering that IP does not precisely fit into the OSI model. Later developments led to security propagating to many other layers of OSI implementations, often because individual manufacturers simply provided it there.) One goal of the IPv6 design team was to provide additional security mechanisms at the IP level of communication.

There are two mechanisms introduced in IPv6 to provide security: the Authentication Header and the Encapsulating Security Header. Both of these are extension headers beyond the basic packet header. The first is deemed generally exportable, because it does not address issues of confidentiality (i.e., the concealment of information), while the second may be restricted to use within certain countries only, depending on the method of encryption.

The Authentication Header provides an important mechanism that has been lacking in the existing Internet, that of ensuring that a packet did indeed come from the specific origin given in the packet's source address and also was not tampered with during transmission. In the summer of 1996, an Internet Service Provider (ISP) in New York City (Panix) was disabled for a lengthy period by the transmission of a flood of forged packets requesting services (e.g., requests to the web server, etc.) whose origin could not be determined; the packets had originating addresses of many sites on the net but those addresses were simply inserted as replacements for the true origin. Despite weeks of effort, the perpetrator of this attack has yet to be identified. A similar attack occurred shortly before Christmas of 1996, with 200 packets a second virtually disabling all service at WebCom in Santa Cruz, CA, resulting in commercial losses to users of this ISP who had hoped to be able to enhance their holiday sales via the Web. Use of the Authentication Header in IPv6 would render a site immune to this sort of damage, one of the few known inherently unfixable vulnerabilities of IPv4 with regard to security against integrity violation. (Network managers have been requested to modify their software to prevent packets with incorrect, or "spoofed," addresses from leaving their sites for the global Internet as a stopgap measure, but there is no guarantee that any or all will take such action, leaving Internet sites which provide public services vulnerable to this kind of attack into the indefinite future.)

Figure 12. Authentication Extension Header

8 bits	8 bits	16 bits
next header	length	reserved
security parameter index (32 bits)		
authentication data (as many 32-bit words as required by the algorithm chosen by the security association)		

The Encapsulation Header is a mechanism to provide for encrypted information transmission. The IPv6 specification does not identify what encryption technique is to be used, only how data are to be identified as having been encrypted. DES (data encryption standard) may be the default case, but users are free to employ any proprietary system of their choice within what is usually termed a *security association*. The important points are that authentication/integrity service

prevents third parties from *introducing* spurious information into the Internet, while encryption prevents third parties from *extracting* information from the Internet that they are not entitled to have (and also provides an additional means for preventing the insertion of counterfeit packets into a communication link).

Figure 13. Encryption Extension Header

8 bits	8 bits	16 bits
next header	length	reserved
security parameter index (32 bits)		
encrypted payload (the entire payload) possibly preceded by parameters dependent on the algorithm chosen (e.g., a public key, a DES-CBC initialization vector, etc.) and possibly followed by padding to a 4- or 8-octet boundary		

2.4.1 Security Associations

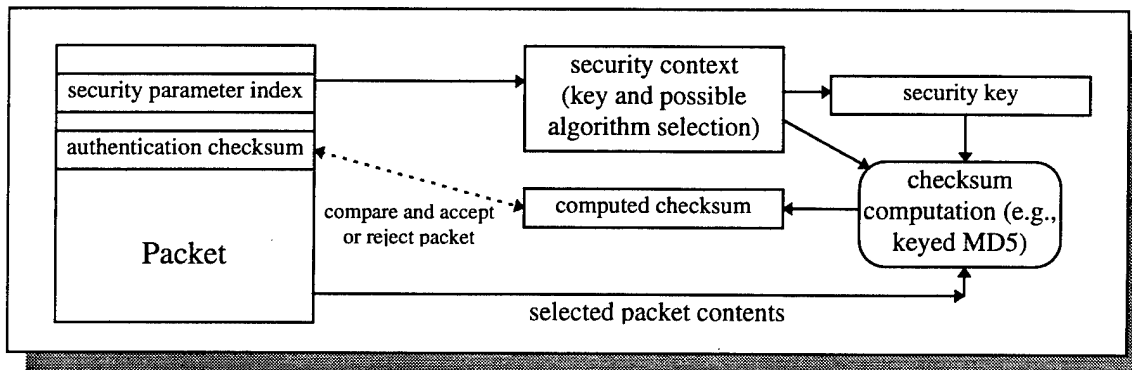
The administration of security using the features of IPv6 necessarily involves extra-technological entities. These have been termed security associations by the protocol developers. A security association is an administrative grouping of site managers who have agreed to use common procedures to control security elements such as encryption keys, encryption algorithms, and identification methods. A security association is analogous to such government groupings as DISCO for personnel security (in which various parts of the DoD agree on a common method of investigation and to accept the results of the BI) or GSA (which certifies the physical security of safes for classified information), or, in the commercial sector, a group of contractors organizing a proposal who agree on ways to protect confidential information from competitors while being able to share it among themselves despite being from different companies.

Once a security association has been created among a group of sites, there must be mechanisms for secure exchange of information among the members of the association. The use of technological means for distribution of keys has been explored theoretically by Diffie and Hellman, and a proposal called Photuris (from the generic name for some fireflies, insects which use electromagnetic signals) is being considered. For many defense-related applications, however, the traditional methods of registered, cleared couriers will probably be employed for some time. The details of the Photuris proposal are quite intricate and well beyond the scope of this tutorial.

Since a site may belong to more than one security association, the authentication and encryption headers contain a field known as the *security index parameter*. This 32-bit value indicates a context in which other security parameters, such as an authentication checksum, are to be interpreted. An authentication checksum may, for example, be based on the contents of the

packet and on a secret key value shared among the members of the association, with different index parameters indicating a different secret key. A member of the association will use this key and the packet contents to compute the checksum by an agreed-upon algorithm; if and only if the computed value matches the transmitted value, the packet is accepted as valid. A similar practice is used to select appropriate keys for decryption of secure contents by deriving the correct key for an association from the index parameter. Figure 14 illustrates the components of the computation for authentication of a packet. Note that only part of the packet is used for the authentication calculation; this is because certain parts of it change as it transits the net (e.g., the hop count is decremented, hop-by-hop routing options alter the destination field at each step, and so on). The selection and application of a key for decryption is similar, except that there is no accept/reject comparison step; instead, the decrypted contents are passed to the higher-level protocol (such as UDP or TCP) for further processing (or handled further by IP in case of encapsulation for certain kinds of tunnels and firewalls).

Figure 14. Authentication computation



2.4.2 Security Algorithms

The IPv6 standard specifies certain security algorithms by default. For authentication, this is *keyed MD5* (message digest 5), designed by Rivest. For encryption, the algorithm is the *DES-CBC* (data encryption standard, cipher block chaining) standard. The IPv6 standards developers felt that it was better to provide at least one choice as part of the standard to ensure that the features would be usable.

But these are only the default cases. Individual security associations are free not only to select and share keys but also encryption algorithms. If the defaults of keyed MD5 and DES-CBC are not considered satisfactory, a security association is able to select any other method it desires to use. The security-parameter-index was given 32 bits to allow it to be large enough to specify almost any reasonable choice of context. The interpretation of this parameter itself is left to the security associations. That is, the use of the term *index* for this value should not imply that only small integers are to be represented. Instead, it may be used for an associative lookup of key and algorithm information or even a set of bit field representing various choices and capabilities. The standard has been designed with both reasonable default choices and open-ended expansion

room, precisely because the details of the implementation of security in data communications is still a topic for extensive research and development.

2.5 Quality of Service and Real-time Support

IPv6 provides capabilities to determine quality of service, including support for real-time and multi-media information flows

The IPv6 definition introduces the concept of a *flow*, which is a sequence of packets sent by a source to a destination (which may be either unicast or multicast) requiring special handling by intermediate routers. A flow is identified by the value in the 24-bit flow label field of the IP packet header *and* by the origin of the packets; i.e., a host originating a flow need only establish uniqueness of the flow label to itself, not across the Internet, unlike the concept of a transient multicast address. (A value of zero in the flow label field indicates that a packet is not part of a flow.) The definition of a flow is a broad one, which does not limit what information may constitute a flow, nor what special handling, if any, by routers is contemplated. In addition to the flow label field, the IP packet header also provides a priority field, whose functions include distinguishing between congestion-controlled traffic and non-congestion-controlled (e.g., real-time) traffic. Together, the capabilities that these fields can provide are usually referred to as *quality-of-service* facilities.

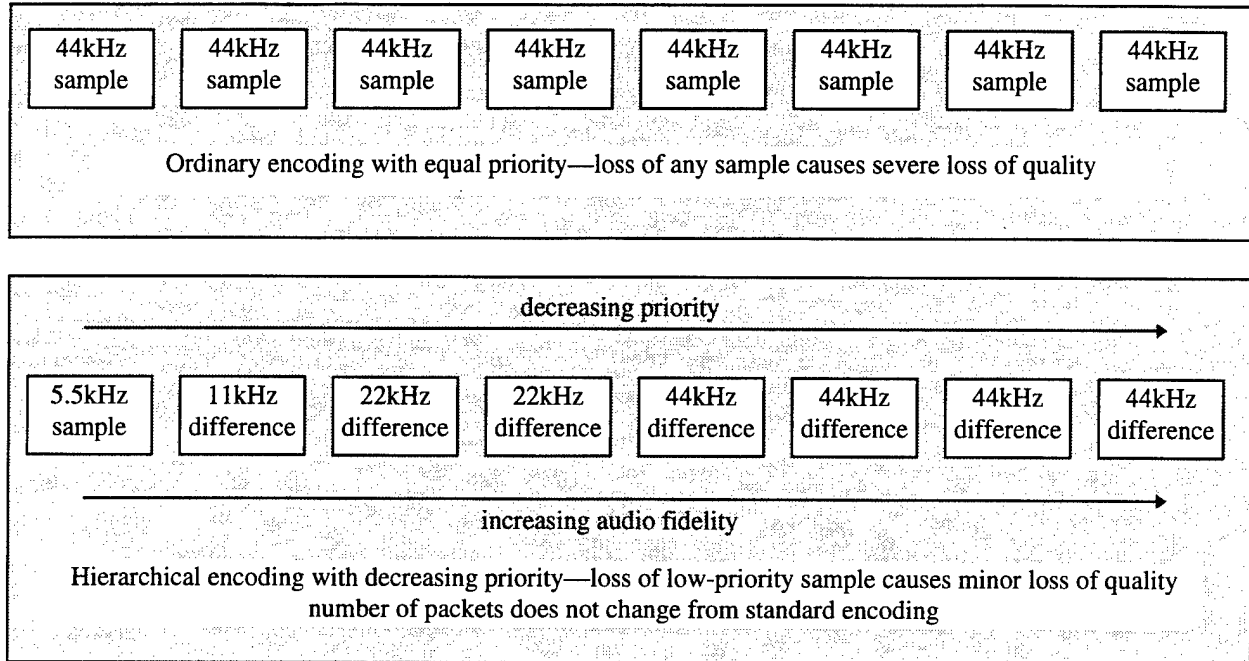
One of the most likely areas in which aggregate flows and quality-of-service assurances will be desired is multimedia, where interactivity can be lost and sound and image characteristics degrade quickly if packets must be retransmitted to any significant extent. High-rate telemetry is also vulnerable to loss of irreplaceable information if retransmission requirements mean that input buffers cannot be emptied before they overflow.

In order to request the performance level required to support a real-time or multi-media flow, a Resource Reservation Protocol known as RSVP was introduced to the Internet Protocol suite even before the definition of IPv6 reached completion. This protocol will be supported by IPv6, but research is now beginning to indicate that some other approaches to resource allocation, such as fair queuing, in which resources are given to a multiplicity of queues on an equal basis (rather than on the more customary first-come, first-served basis, which can allow one source to claim a large share of the common resource), will provide better overall utilization than will reservation, which inevitably leaves some of the reserved resources unused.

Research has also revealed some other techniques that can take advantage of the quality-of-service facilities of IPv6 through *hierarchical coding*. An example given by Huitema is of sound (e.g., speech) encoding, in which sampling done at a high rate of 44 kHz is processed into the same number of packets, but which carry the information in a different form: There is a base packet containing the minimum information for the speech to be intelligible, followed by packets giving the differences between the base and the higher sampling rates of 11 kHz, 22 kHz, and 44 kHz, providing successively higher levels of fidelity. The base packet is given the highest priority, with decreasing priority given to the higher sampling rates. Thus, some level of information is almost sure to get through on a timely basis, whereas intelligibility would suffer

much more from loss of packets if all of the 44 kHz samples were transmitted at equal priority. This comparison is illustrated in Figure 15.

Figure 15. Comparison of ordinary and hierarchical encoding



(Something similar is done in analog transmission of FM stereo signals, in which the center band carries the sum of the left and right channels, while a higher band carries their difference; an addition/subtraction process in the receiver allows L and R to be reconstructed from L+R and L-R. Loss of the higher band, which is more likely to occur in fringe reception areas, still leaves a monaural signal carrying the intelligible information.) Thus it can be seen that the flow and priority fields can work together to provide a means for effective handling of multi-media information. (And an analogous method can be employed to transmit video signals, using both a time- and a space-sampling and differencing technique; some of this has been demonstrated in real-time by the Sarnoff Laboratories of SRI International in Princeton, NJ.)

Finally, it is worth mentioning that the concept of flows should not be confused with that of virtual circuits. Huitema goes to some length to make clear that flows are not *necessarily* given any different handling from unlabeled datagrams. Huitema also takes the opportunity in his discussion of real-time support and flows to consider the relationship between IP and ATM links. Although this is somewhat beyond the scope of this tutorial, his concerns are that ATM has chosen some parameters (e.g., packet size) and characteristics (the virtual circuit) that are at odds with the packet size and packet-driven philosophy of IP. These may lead to some interesting cost phenomena, as ATM virtual circuits are established and broken with great frequency. Huitema suggests that other technologies (Fast Ethernet, Gigabit Ethernet, fiber, and even direct operation of SONET links) may compete with ATM for providing the bandwidth required for real-time

support using IPv6; in any case, he believes that market competition will determine the ultimate choices to be made.

2.6 Programming Interface

Programmers have new tools and interfaces for IPv6 but can still support IPv4
--

Programmers who will need to write new applications or modify existing ones that require the services of IPv6 will need to use new interfaces. The definitions of packets contained in the socket libraries that are furnished as part of the C/UNIX programming environment will need to reflect the changes from IPv4 to IPv6, such as packet size, field definitions, and additional program procedure calls. The number of changes required is surprisingly small and are described in only a couple of pages of Huiterna's book. The designers of IPv6 took care to allow for the need to maintain both IPv4 and IPv6 connections by application programs and higher level protocols, and there are address conversion routines, all of which will be provided by vendors of compiler environments who intend to be IPv6-compliant.

3. What Does the Telecommunications Manager Need to Know and Do?

3.1 Identification of Software and Hardware Vendors

Many vendors already support IPv6 implementations

IPv6 implementations are being developed for many different host operating systems and routers. These include 4.4-lite BSD, BSDI/OS, Digital UNIX, DOS/Windows, HP-UX, Linux, NetBSD, Novell, Solaris 2, Streams, and BayNetworks, Cisco, Telebit, Penril and Ipsilon routers. This list comprises the most widely used UNIX workstation vendors (Hewlett-Packard, Sun Microsystems, and Digital Equipment), the principal networking vendors (Novell, Bay, Cisco, 3Com), and other sources, including the freeware Linux clone of UNIX. Most of these vendors plan to provide phased implementation of the various features, such as multicasting, and may also phase in support for different link protocols (Ethernet, FDDI, frame relay, etc.).

By attending various industry conferences, subscribing to journals, and searching the Internet, it should be possible to identify a wide variety of further offerings in the area of communications software. Generally, the hardware vendors, some of whom are mentioned above, will provide information on where software may be obtained, if they do not provide it themselves. As IPv6 spreads in acceptance, virtually every software system *will* be obliged to provide support for it. A manager with an installed equipment base, however, will probably want to pursue offerings from the existing vendors with a presence in his shop, as that will minimize retraining and perhaps afford financial advantages, since upgrades are generally less costly than entirely new software.

3.2 Personnel Training

Many sources will offer personnel training courses

Early opportunities for personnel involved with network maintenance, configuration, and management to become familiar with IPv6 are already available. They are primarily found in sessions, seminars, and continuing education meetings at various conferences on data communications technology. These began as far back as early 1995.

As IPv6 software from telecommunications vendors reaches the market, those vendors will be impelled to provide training in its use, both to support the purchasers and to provide a readier market for the products. Further, the independent education vendors, such as James Martin Associates, will surely develop courses in which networking personnel may be enrolled.

Courses are likely to come in at least two flavors: One for those familiar with IP already, and one for those starting from ground zero. In addition, the courses will probably focus separately on the two areas of transition of existing networks to the new regime and that of taking advantage of the new capabilities (autoconfiguration, real-time, anycasting, multicasting) of IPv6.

For example, the University of New Hampshire provides tutorials and short courses in conjunction with its data communications research group, the InterOperability Lab. They offer a number of papers on topics in computing and data communications, including IPv6, in their On-Line Educational Program. The last is even available without charge.

3.3 User Awareness

Users must be kept informed of the effects of the transition to IPv6

In order for a successful transition to IPv6 to take place, the full co-operation, rather than resistance, of the end-user community is essential. This is characteristic of most problems in technology transfer. Fortunately, the design of IPv6 has kept in mind the need to minimize the impact of the new protocol on end-user operations. Most end-users make use of symbolic host names, rather than numeric addresses, and these symbolic names will not change. Users will occasionally see numeric addresses reported back by software (e.g., `nslookup`, `traceroute`) but they will not have to manipulate such addresses or remember them for the most part.

As is usual in technology transfer, the alert manager will plan for advance information briefings long before the date of any transition events. These briefings will emphasize the minimal impact of the next generation of IP, describe the benefits, and explain the transition plan. The actual transition will be most effectively initiated with a group of early adopters, usually an internal R&D or similar advanced development group used to experimenting with new technology. These participants will assist management both by resolving the problems that inevitably occur in the introduction of a complex of new software and by serving as an additional resource in the promotion and instruction of the larger body of users when the IPv6 implementation is phased in.

3.4 Benefits of Transition to IPv6

The IPv6 transition has real benefits with modest cost and little disruption

IPv6 provides the information systems manager with a significant number of benefits in return for making the transition to the new technology. Although the most important gain from IPv6 is certainly that of not being left behind as the Internet evolves, which would otherwise mean being unable to access IPv6-only sites, experiencing performance impacts from old technology conversion, and other deficits, there are also positive gains to the manager from making the transition to IPv6. Most important, as has been pointed out previously, the introduction of IPv6 does not significantly alter the environment experienced by end-users, and the introduction can be done without a "flag day" or site-wide shutdown.

The immediate benefits to an individual site come from increased productivity through the new facilities of IPv6. The autoconfiguration capability makes it possible for the network management staff to add equipment and adjust network topologies with a great deal less effort on their part, not to mention reducing the disruption of ongoing end-user work. The use of new

features, such as anycast addresses, will improve network utilization, increasing the productivity of all activities that make use of the Internet, while the streamlined IPv6 packet header will have a similar effect by increasing the performance of router software.

Economic productivity is also increased through the provision of new features, such as security and quality-of-service capabilities. The former guards against the loss of resources, reducing the amount of time that must be spent in manual recovery and in detecting intrusion. The latter makes possible the use of the Internet for many new applications, which increase the output of personnel.

To be sure, the history of technology should remind the manager that new technology seldom results in lowered expenditures, except in organizations that have ceased to grow at all. New technology means faster growth, higher productivity, and new capabilities in return for those increased expenditures. An IT manager in an organization with a future must be prepared to spend more money when a new technology leads to new possibilities for applications, but the manager *will* get more results for the outlays than was the case for the old technology.

3.5 Transition Planning

A sound transition plan to IPv6 is vital for success

The IPv6 design group put a great deal of effort into making the transition from IPv4 to IPv6 possible with minimal disruption of existing environments and practices. The details of the transition process are in RFC 1933, *Transition Mechanisms for IPv6 Hosts and Routers*, which is available on the Internet in various RFC indexes, such as:

- ◆ http://www.graphcomp.com/info/rfc/rfc_list.html
- ◆ <http://andrew2.andrew.cmu.edu/rfc/rfc-index.html>
- ◆ <http://www2.es.net/hypertext/rfcs.html>

3.5.1 From the "Simple Internet Transition Mechanisms" Internet Document

The following is a verbatim quotation of a section from the referenced document, which is available on the Internet (<http://playground.sun.com/pub/ipng/html/ipng-transition.html>):

The key transition objective is to allow IPv6 and IPv4 hosts to interoperate. A second objective is to allow IPv6 hosts and routers to be deployed in the Internet in a highly diffuse and incremental fashion, with few interdependencies. A third objective is that the transition should be as easy as possible for end-users, system administrators, and network operators to understand and carry out.

The Simple Internet Transition (SIT) is a set of protocol mechanisms implemented in hosts and routers, along with some operational guidelines for

addressing and deployment, designed to make transitioning the Internet to IPv6 work with as little disruption as possible.

SIT provides a number of features, including:

Incremental upgrade and deployment	Individual IPv4 hosts and routers may be upgraded to IPv6 one at a time without requiring any other hosts or routers to be upgraded at the same time. New IPv6 hosts and routers can be installed one by one.
Minimal upgrade dependencies	The only prerequisite to upgrading hosts to IPv6 is that the DNS server must first be upgraded to handle IPv6 address records. There are no prerequisites to upgrading routers.
Easy Addressing	When existing installed IPv4 hosts or routers are upgraded to IPv6, they may continue to use their existing address. They do not need to be assigned new addresses. Administrators do not need to draft new addressing plans.
Low start-up costs	Little or no preparation work is needed in order to upgrade existing IPv4 systems to IPv6, or to deploy new IPv6 systems.

The mechanisms employed by SIT include:

- An IPv6 addressing structure that embeds IPv4 addresses within IPv6 addresses, and encodes other information used by the transition mechanisms.
- A model of deployment where all hosts and routers upgraded to IPv6 in the early transition phase are "dual" capable (i.e., implement complete IPv4 and IPv6 protocol stacks).
- The technique of encapsulating IPv6 packets within IPv4 headers to carry them over segments of the end-to-end path where the routers have not yet been upgraded to IPv6.
- The header translation technique to allow the eventual introduction of routing topologies that route only IPv6 traffic, and the deployment of hosts that support only IPv6. Use of this technique is optional, and would be used in the later phase of transition if it is used at all.

SIT ensures that IPv6 hosts can interoperate with IPv4 hosts anywhere in the Internet up until the time when IPv4 addresses run out, and allows IPv6 and IPv4 hosts within a limited scope to interoperate indefinitely after that. This feature

protects the huge investment users have made in IPv4. SIT ensures that IPv6 does not render IPv4 obsolete. Hosts that need only a limited connectivity range (e.g., printers) need never be upgraded to IPv6.

The incremental upgrade features of SIT allow the host and router vendors to integrate IPv6 into their product lines at their own pace, and allows the end users and network operators to deploy IPng on their own schedules.

3.5.2 Interoperability

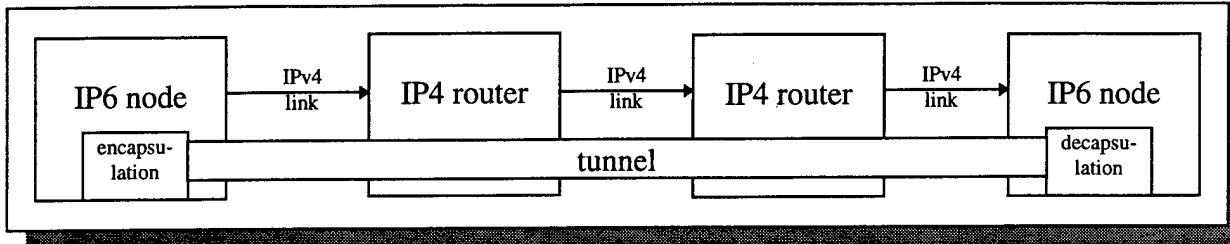
3.5.2.1 Dual-Stack Configuration

The interoperability of the protocols is a primary concern in implementing a transition. This was a high priority for the IPv6 designers, in fact. The requirement that node software supporting IPv6 also recognize and support IPv4 is not difficult; it only means continuing to support the code that is already in existence, since a simple branch on the value in the first octet (4 or 6) will allow the software to select the correct algorithms to process a packet between either the new or old implementation. Nodes with such software implementation are known as *dual-stack configurations*.

Similarly, the capability for an IPv6 node to create and dispatch a packet to an IPv4 destination is not hard; again, the presence of the old software needed to handle IPv4 packets will take care of this job and only requires recognition of the presence of an IPv4-type address. Since all IPv4 addresses are zero in the first 96 bits, this is quick and easy.

3.5.2.2 Tunnels

The only real difficulty is in arranging for communication between two IPv6 nodes which are connected only through an IPv4 routing path. Note that this implies that the IPv6 nodes must have IPv4 addresses (or otherwise the IPv4 router could not reach them at all). Given this, the answer is straightforward, if somewhat wasteful of resources: The IPv6 packets sent between the two IPv6 nodes are encapsulated in IPv4 packets as payload. (IPv4 has been extended to include a protocol recognition type for IPv6, just as if IPv6 were a higher-level protocol.) A special procedure, known as *tunneling*, is used to create an automated link of this sort. A tunnel, once configured, appears to be no different from an ordinary link between the two IPv6 nodes. (This has some special impacts on hop counting—the tunnel appears to be only one hop to the IPv6 packet, although there may be many IPv4 links inside the tunnel—and on MTU/fragmentation management.)

Figure 16. IPv6 Tunneling Through IPv4 Routers

Tunnels will be a managed resource in the transitional IPv6 world because they have implications for performance and resource management. For example, a tunnel's packets appear to come from a single source (the encapsulating node), whereas they actually represent a multiplicity of net users in the IPv6 regime beyond the tunnel end. This impairs their ability to compete for resources with the packets originating in the IPv4 hosts themselves, since the usual algorithm is to allocate resource share on the basis of equality by source, and the number of sources is concealed by tunneling. Also, since a tunnel appears to be a single hop to the IPv6 routing regime, without attention to this, routing may prefer an inefficient tunnel with many IPv4 hops inside it over a shorter IPv6-only path. Finally, since tunnels are objects that span multiple organizations, their establishment, management, and eventual elimination must necessarily engage the attention of network managers except in the case of an individual host's communication, where the tunnel is a temporary, automatically established and disposed-of, object.

3.5.2.3 Domain Name Service

The final component of ensuring interoperability between IPv6 and IPv4 is the domain name service. IP itself is concerned only with numeric identifiers (32 bits for IPv4, 128 bits for IPv6), but human users cannot readily manage such designations. The DNS (domain name service) servers must be upgraded to provide both IPv4 and IPv6 addresses in response to a request for the IP address corresponding to a symbolic host name. In terms of C language coding, this means that a new procedure call (`hostname2addr`) must be added to provide the IPv6 equivalent of today's `gethostbyname`. Sites which run their own DNS must upgrade their servers, as well as depending on the global Internet DNS sites (e.g., InterNIC) to make the transition to IPv6.

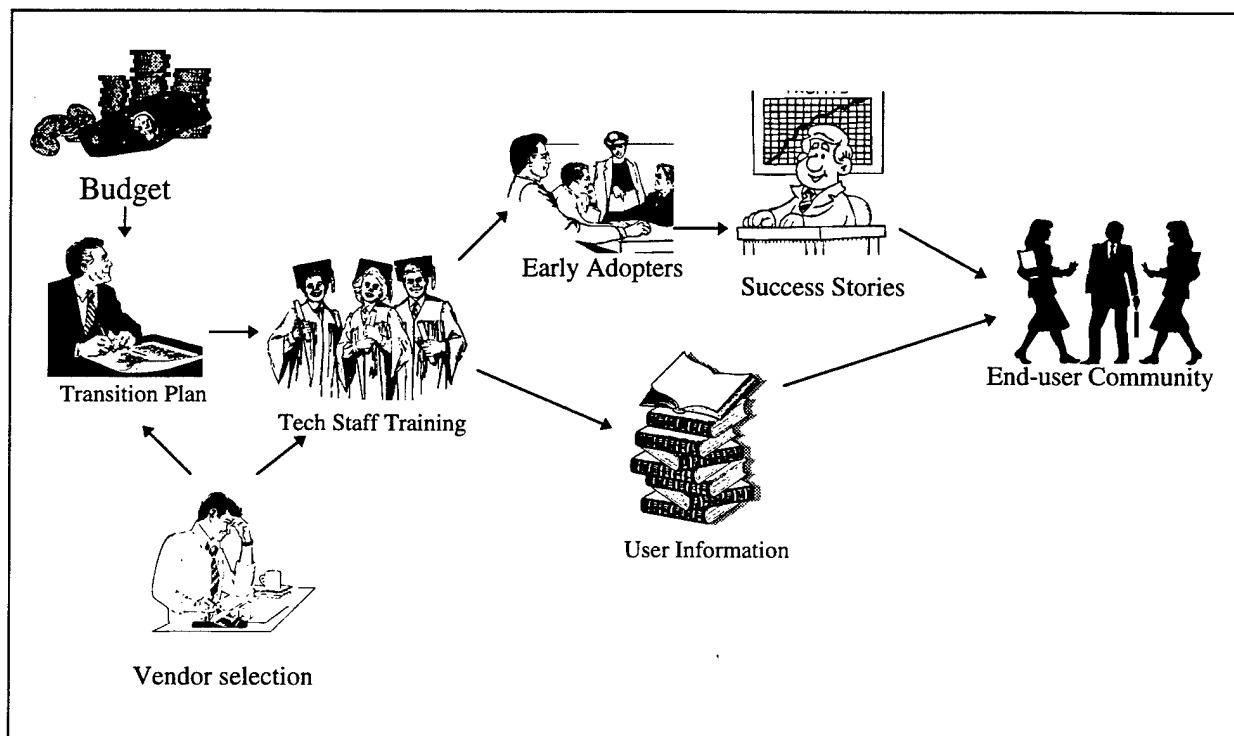
3.5.3 Managerial Actions

The telecommunications manager must take action to deal with the introduction and dissemination of IPv6. As described above, there are several components of any action plan. These are:

- Allocation of budgetary resources to implement the transition
- Identification of appropriate vendors for software to support IPv6

- Identification of vendors for personnel training services
- Identification of personnel to be trained and training them
- Dissemination of information to end-users on how the transition will occur and how they will be affected
- Creation of a detailed implementation plan with schedules for each of the above steps, concluding with the installation of the IPv6 software on nodes under the manager's control.

Figure 17. Transition process management



For systems with many subnetworks and sites, an overall plan will cover the execution of these steps for each individual location, to the extent that they need to be partitioned. Because the protocol itself is designed to interoperate with IPv4, transition to IPv6 can be done in a piecewise fashion. This permits the manager of a large operation, such as may be found in the US government or a Fortune 50 corporation, to minimize risk by engaging only smaller, less-critical areas of operation in the transition first. As has always been the case for new technology introduction, it is highly desirable to identify a set of *early adopters*—a group, such as the R&D department, that is willing to take risks, explore new methodologies, and explore the benefits of the new system. These individuals should then be used as missionaries to the more conservative groups within the organization (e.g., accounting, payroll) to persuade them to make the transition and to assist them in doing so without disruption.

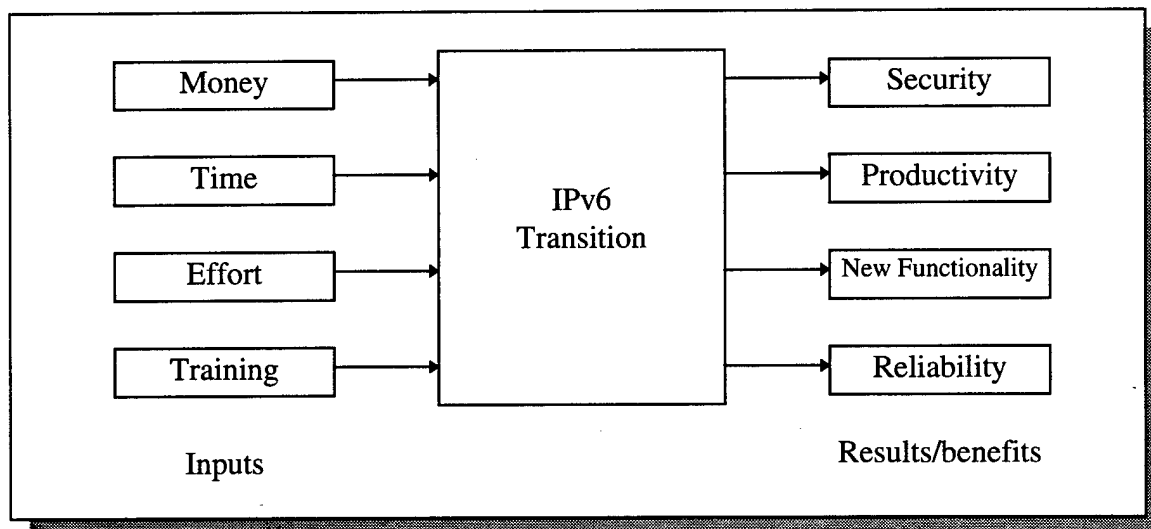
The steps shown as bullets above are illustrated in Figure 17, along with the role of the early adopters in proselytizing the user community and the influence of vendor selection on both the plan for the transition and also the arrangements for training the technical staff. Note that the technical staff communicates its new knowledge directly to the early-adopter group and also to the general user community through the preparation of documentation.

3.6 The Bottom Line

**The IPv6 transition will occur;
the only choice for management is *how* and *when*—
but there will be a solid payoff**

The bottom line regarding management of IPv6 is that there is no real choice about making the transition, other than to decide exactly when this step must be taken. As IPv6 spreads through the Internet, the remaining sites using IPv4 will become increasingly isolated. New network entities will eventually be required to be IPv6 only, making access to them difficult, and tunneling of IPv4 packets between IPv4 sites through the IPv6 environment will experience much poorer performance. Multimedia applications will increase in importance, and they will depend on the new facilities provided by IPv6 in handling real-time flows. Security considerations will also make continuing in the IPv4 environment increasingly questionable, as vulnerability to "address-spoofing" attacks which result in denial of service becomes an unacceptable risk, when IPv6 provides the necessary authentication defense, along with other integrity services.

Figure 18. The IP transition bottom line



Internet Protocol Next Generation (IPv6) Tutorial

The transition to IPv6 will cost money, take time, require replacement of software, and mean retraining of personnel. Unlike the year-2000 problem currently receiving much attention and functioning in much the same time frame, this investment will provide some **immediate benefits**, such as autoconfiguration, more efficient network usage, increased security, and support for real-time flows. IPv6 conversion will leave a site's current operations mostly unchanged, as far as end-users are concerned, just as replacing current software with software that handles four-digit years provides no immediate functional changes. In both cases, the alternative to conversion is for a site to *have no future* at some point, although for IPv6, the date is not quite as definite as 01 January 2000. And in that future with IPv6, there are very substantial additional long-term benefits from being able to proceed with development of networked applications without concern that the site will be blocked from participating in the development and expansion of the computing industry and the Internet.

4. References

Most of the references on IP and other protocols are found on the Internet itself. They may be retrieved using a World Wide Web browser or by FTP, in the case of RFCs. A few books are now being published to provide information on IPv6, but the field is still in flux, and printed volumes will quickly become out of date, while on-line information is being updated regularly.

1. Bradner, Scott O. and Mankin, Allison. *Internet Protocol Next Generation*. Addison-Wesley. (Information available at <http://aw.com/cp/bradner-mankin.html>)
2. Conner, Douglas E. *Internetworking With TCP/IP*, 2nd ed., vol. 1. Prentice-Hall, Upper Saddle River, NJ: 1991
3. Hinden, Robert M. "IP Next Generation Overview."
<http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html>
4. Huitema, Christian. *IPv6: The New Internet Protocol*. Prentice-Hall, Upper Saddle River, NJ: 1996
5. Lehtovirta, Juha. "Internetworking: Transition from IPv4 to IPv6."
<http://www.Tascomm.fi/~jlv/ngtrans/>
6. IPv6 Technology Project at the National Institute of Standards and Technology.
<http://snad.ncsl.nist.gov/itg/ipv6.html>
7. RFCs:
 - ♦ Deering, S. and Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, August 1996.
 - ♦ Hinden, Robert and. Deering, S, "IP Version 6 Addressing Architecture", RFC 1884, December 1995.
 - ♦ Gilligan, R and Nordmark, E., "Transition Mechanisms for IPv6 Hosts and Routers," RFC 1933, April 1996.
 - ♦ Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6), RFC 1970, August 1996.
 - ♦ Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 1971, August 1996.
 - ♦ McCann, J., Deering, S., and Mogul J. "Path MTU Discovery for IP version 6", RFC 1981, August 1996

5. Appendices

5.1 IPv6 Packet Header Format

The IPv6 packet header consists of ten 32-bit words (40 8-bit octets or bytes) as follows:

	← 32 bits →		
0	version	priority	flow label
1	payload length in octets (bytes)		next header
2	source address		
3			
4			
5			
6	destination address		
7			
8			
9			

The meaning of the various fields in the packet are:

Field	Size (bits)	Usage
version	4	always 6, denoting IP version 6
priority	4	priority of transmission (for control, congestion management, real-time)
flow label	24	flow identification (for associating packets with a flow to support real-time and multimedia operations)
payload length	16	size of payload in octets (bytes)—does not include header size
next header	8	identifies type of extension header, if any, at start of payload field (same values as for IPv4)
hop limit	8	maximum number of routing steps permitted (value decremented at each step and packet discarded if value reaches zero)
source address	128	address of originating host or router
destination address	128	address of destination host or router (may vary if route selection is used, since final destination may be in the routing extension in that case, and this specifies only the next hop)

5.2 Feature Comparison of IPv4 and IPv6

Feature	IPv4	IPv6
Address	32 bits (4 octets)	128 bits (16 octets)
Address space	over 10^9 possible addresses	over 10^{38} possible addresses
Packet header	variable size—time-consuming to handle	fixed size (40 octets)—more efficient
Special fields in header	many types, often not supported by vendors due to impact on performance	eliminated for efficiency or replaced by other features
Packet size	65536 octets maximum	- normal packet up to 65536 octets "jumbogram"—up to 4 billion octets for high-performance computing LANs
Address allocation	- by network classes A, B, C (large, medium, small nets) - CIDR (stopgap measure to deal with address space exhaustion, router table overgrowth) - local use limited to link only - expansion room used up	- IPv4 compatibility - hierarchical by registry, provider, subscriber, and subnet - hierarchical by geographic region - local use by link or site - over 70% of addresses reserved for future expansion
Address notation (numeric)	dotted decimal notation	hexadecimal with colons and shortcuts (abbreviations); IPv4 addresses a special case

Internet Protocol Next Generation (IPv6) Tutorial

Feature	IPv4	IPv6
Address types	• point-to-point • local broadcast (depends on physical link features); limited multicast • experimental anycast (not globally available)	• point-to-point (termed unicast) • multicast (sends to many interfaces at once) ◆ by link ◆ by site ◆ by organization ◆ by any grouping • anycast (reaches one of a group of interfaces)
Fragmentation	possible multiple step fragmentation, done by routers, impacting routing performance	done at most once, by host (not router), after MTU discovery over the path, improving router performance
Quality of service	defined but not generally used consistently	• flow labeling • priority • support for real-time data and multimedia distribution
Security	limited; no authentication or encryption at IP level (dependence on higher-level protocols; vulnerable to denial-of-service and address deception or "spoofing" attacks)	• authentication (validation of packet origin) • encryption (privacy of contents) requires administration of "security associations" to handle key distribution, etc.

Internet Protocol Next Generation (IPv6) Tutorial

Feature	IPv4	IPv6
Configuration management	• manual compilation of tables; even simple networks require investment of time to configure • support for local diskless workstation address resolution • heavy reliance on default routing paths	• automatic configuration of link-local addresses based on physical addresses (e.g. Ethernet) • stateless automatic configuration for simple networks • diskless workstation support • limited human administration, mostly for complex environments • neighbor discovery algorithm builds routing paths
Routing management	BGP-4 between subdomains • uses TCP—high overhead • designed for 32-bit addresses • single address family • full tables use large amounts of storage OSPF, RIP within subdomains	IDRP between subdomains • IP datagram-based—low overhead • accommodates 128-bit addresses • multiple address types • aggregated tables economize storage OSPF and RIP updated but similar

5.3 Glossary

anycast	addressing any single member (usually the nearest) of a defined group of interfaces
autonomous system	an IPv4 term for a subdomain or subnetwork under the control of a single entity (e.g., a corporation or other enterprise) which manages all configuration within the subdomain
BGP-4	Border Gateway Protocol (version 4)—for IPv4 routing between subdomains (autonomous systems)

Internet Protocol Next Generation (IPv6) Tutorial

broadcast	addressing all interfaces on a network; usually provided as a feature of the physical structure of a LAN, such as Ethernet
byte	usually 8 bits; an octet
configuration	assignment of addresses to interfaces in a network or subnet; the process of making the assignment
connection	a time-persistent relationship between two communicating sites; an interaction characterized by multiple communications activities of a related nature
datagram	packet ("datagram" is often used to emphasize the connectionless nature of a protocol)
field	a contiguous grouping of information, as in a packet header
flag day	an event when operation of a system or systems must be suspended to convert to a new mode of operation or software (an undesirable phenomenon)
fragmentation	breaking a large packet into smaller units for transit over a size-restricted transmission link (i.e., when a packet's size exceeds the MTU)
header	the first part of a packet describing its content and processing options
hop	a single routing step
host	a node whose purpose is computing, not the transfer of packets among other nodes
IDRP	Inter-Domain Routing Protocol—for IPv6 routing between subdomains
interface	a network connection; a single addressing point (a node may have more than one if it is connected to different networks, e.g., for availability assurance)
Internet	a global network connecting many subordinate networks belonging to entities of all sorts, including the government, military, commercial, educational, and non-profit institutions
Internet Protocol	a data communications protocol for transfer of information in packets by means of a best-efforts, connectionless, unreliable (non-error-correcting) scheme
intranet	a network internal to an organization that makes use of the Internet Protocol for transmission, rather than a proprietary manufacturer's protocol (such as Novell Netware®)
IP, IPv4, IPv6	Internet Protocol (v4 = version 4, v6 = version 6)

Internet Protocol Next Generation (IPv6) Tutorial

IPng	IP next generation—a term used to refer to IPv6 until the version 6 designation was official
jumbogram	a packet larger than the normal 65536-octet limit, having a maximum size of 4G octets, primarily expected to be used in supercomputing LAN environments (intranets) for maximum performance but unlikely to travel on the Internet itself
MTU	maximum transfer unit; limit on packet size able to transit a physical link
multicast	addressing of a group of interfaces; more versatile than broadcast
multi-homed	describing a host which is connected to more than one network and therefore has more than one IP address (all router nodes are multi-homed because the purpose of a router <i>is</i> to connect between networks)
node	a computing device attached to a network, such as the Internet; a host or router
octet	8 bits; same as a byte for most hardware
OSPF	Open Shortest Path First—a routing protocol within subdomains; efficient in use of resources but complex to implement, due to its use of state tables
packet	the basic unit of data transmission in IP
path	the linkage between a source and destination; a list of nodes traveled by a packet from origin to destination
protocol	a well-defined set of formats and commands to manage the transfer of information
RIP	Routing Information Protocol—a routing protocol within subdomains; easily implemented but not able to make efficient use of connections
router	a node whose primary activity is the transfer of packets received from a network to destinations on another network
routing	discovering and selecting an appropriate path for transmission of a packet or all packets; also, selecting the next node in a path
security association	an administrative organization of sites that have agreed on common procedures for sharing and distributing security keys for authentication and encryption, and also possibly algorithms for those tasks
security parameter index	a 32-bit value used to specify a context for the interpretation of security keys and, optionally, encryption algorithms (if the default choice is not desired)

Internet Protocol Next Generation (IPv6) Tutorial

subdomain	an organizational unit within the Internet smaller than a global domain, often under the control of an entity, such as a corporation or government agency, able to control address assignments and configuration within the unit
TCP	Transmission Control Protocol—a higher-level protocol using the services of IP to provide reliable connection-oriented services for applications (e.g., Telnet, World-Wide Web, electronic mail, etc.)
tunnel	a logical relationship established between two nodes which, through encapsulating packets as the payload of other packets, makes a multi-step route appear to be a single hop; used to pass IPv6 packets through nodes with only IPv4 capability, to provide specific routing, or to provide security
UDP	User Datagram Protocol—a higher-level protocol using the services of IP to provide an unreliable connectionless service for applications (e.g., Network File System); used instead of TCP where the underlying physical system, such as an Ethernet LAN, is highly reliable
unicast	addressing a single interface for packet transmission

5.4 Standards Information

The standards process by which IPv6 was developed is less formal than that which characterizes many of the standards of the computing industry. The formal standards bodies include the ITU (CCITT formerly), ANSI, and IEEE. The Internet Protocol, however, has been characterized more by informal consensus among experts than by a formal standards procedure with government sanction. The body involved in the IPv6 standard was the Internet Engineering Task Force (IETF). Being small, composed of technology experts, and non-governmental, the IETF was able to act rapidly and develop the standard in only two years (as shown in Figure 6 previously).

A convenient characteristic of Internet standards is that they are propagated by use of the Internet itself. From tradition of many years, the standards are known as RFCs (Request For Comment). Not all RFCs have been entirely serious, as a review of the file of some 2000 of them will show. But they are the primary mechanism for disseminating new proposals, providing reference for communications protocol definitions, and maintaining a historical record of the evolution of network technologies.

The RFCs are not distributed in printed format, as has been the case for standards of other types (such as those issued by ANSI). Instead, complete files of RFCs are maintained at publicly available Internet sites at various locations around the world. Some of these locations are given by the following URLs, which may be accessed by any WWW browser:

- ♦ http://www.graphcomp.com/info/rfc/rfc_list.html
- ♦ <http://andrew2.andrew.cmu.edu/rfc/rfc-index.html>
- ♦ <http://www2.es.net/hypertext/rfcs.html>

Internet Protocol Next Generation (IPv6) Tutorial

Although these indices are HTML documents, the RFCs themselves are maintained as plain ASCII text which can be read with even a text-only browser, such as lynx. Unfortunately, this means that diagrams must be rendered in a somewhat clumsy fashion using characters, but the meaning is generally clear.

Some of the RFCs for the development of the IPv6 standard are given in the following table:

Topic	RFC #s
CIDR	1517, 1518, 1519, 1520
TUBA	1347, 1526, 1561
Pip	1621, 1622
TP/IX	1475
CATNIP	1707
SIPP	1710
Discussion	1667-83, 1686-88, 1705, 1715
IETF decision	1719
Directors' recommendation	1752
IPv6 definition	1883
IPv6 address architecture	1881, 1884, 1887, 1897
ICMP	1885
Domain Name System for IPv6	1886
Transition issues for IPv6	1933
Security for IPv6	1825, 1826, 1827, 1828, 1829
Neighbor Discovery Algorithm	1970
Stateless Address Autoconfiguration	1971

In addition to these RFCs issued and published, there are a number of drafts that have yet to achieve the status of an RFC. Many of these may be reached as hyperlinks from the IPng specifications Web page at:

<http://playground.sun.com/pub/ipng/html/specs/specifications.html>