FOR FURTHER STUDY

Case et. al. [RFC 2570] presents an overview of SNMPv3, gives background and motivation, and discusses changes among the various versions. It also contains a summary of RFCs related to v3, and explains which v2 standards still apply. Many other RFCs discuss individual aspects of the protocol. For example, Wijnen et. al. [RFC 2575] presents the view-based access control model, and Case et. al. [RFC 2572] discusses message handling.

ISO [May 87a] and [May 87b] contain the standard for ASN.1 and specify the encoding. McCloghrie et. al. [RFCs 2578, 2579, 2580] define the language used for MIB modules and provide definitions of data types. Case et. al. [RFC 1907] defines version 2 of the MIB.

An older proposal for a network management protocol called HEMS can be found in Trewitt and Partridge [RFCs 1021, 1022, 1023, and 1024]. Davin, Case, Fedor, and Schoffstall [RFC 1028] specifies a predecessor to SNMP known as the Simple Gateway Monitoring Protocol (*SGMP*).

EXERCISES

30.1 Capture an SNMP packet with a network analyzer and decode the fields.

30.2 Read the standard to find out how ASN.1 encodes the first two numeric values from an object identifier in a single octet. Why does it do so?

30.3 Read the two standards and compare SNMPv2 to SNMPv3. Under what circumstances are the v2 security features valid? Invalid?

30.4 Suppose the MIB designers need to define a variable that corresponds to a two-dimensional array. How can ASN.1 notation accommodate references to such a variable?

30.5 What are the advantages and disadvantages of defining globally unique ASN.1 names for MIB variables?

30.6 Consult the standards and match each item in Figure 30.11 with a corresponding definition.

30.7 If you have SNMP client code available, try using it to read MIB variables in a local router. What is the advantage of allowing arbitrary managers to read variables in all routers? The disadvantage?

30.8 Read the MIB specification to find the definition of variable *ipRoutingTable* that corresponds to an IP routing table. Design a program that will use SNMP to contact multiple routers and see if any entries in their routing tables cause a routing loop. Exactly what ASN.1 names should such a program generate?

30.9 Consider the implementation of an SNMP agent. Does it make sense to arrange MIB variables in memory exactly the way SNMP describes them? Why or why not?

30.10 Argue that SNMP is a misnomer because SNMP is not "simple."

30.11 Read about the IPsec security standard described in Chapter 32. If an organization uses IPsec, is the security in SNMPv3 also necessary? Why or why not?

30.12 Does it make sense to use SNMP to manage all devices? Why or why not? (Hint: consider a simple hardware device such as a dialup modem.)

31

Summary Of Protocol Dependencies

31.1 Introduction

TCP/IP has spawned more applications than we can discuss in a single textbook. In general, each defines its own application protocol and relies on TCP or UDP for end-to-end transport. In fact, any programmer who builds a distributed application using TCP/IP defines an application-level protocol.

Although it is not important to understand the details of all protocols, it is important to know which protocols exist and how they can be used. This chapter provides a brief summary of the relationships among fundamental protocols, and shows which are available for use by applications.

31.2 Protocol Dependencies

The chart in Figure 31.1 shows dependencies among the major protocols we have discussed. Each enclosed polygon corresponds to one protocol, and resides directly above protocols that it uses. For example, the mail protocol, SMTP, depends on TCP, which depends on IP.

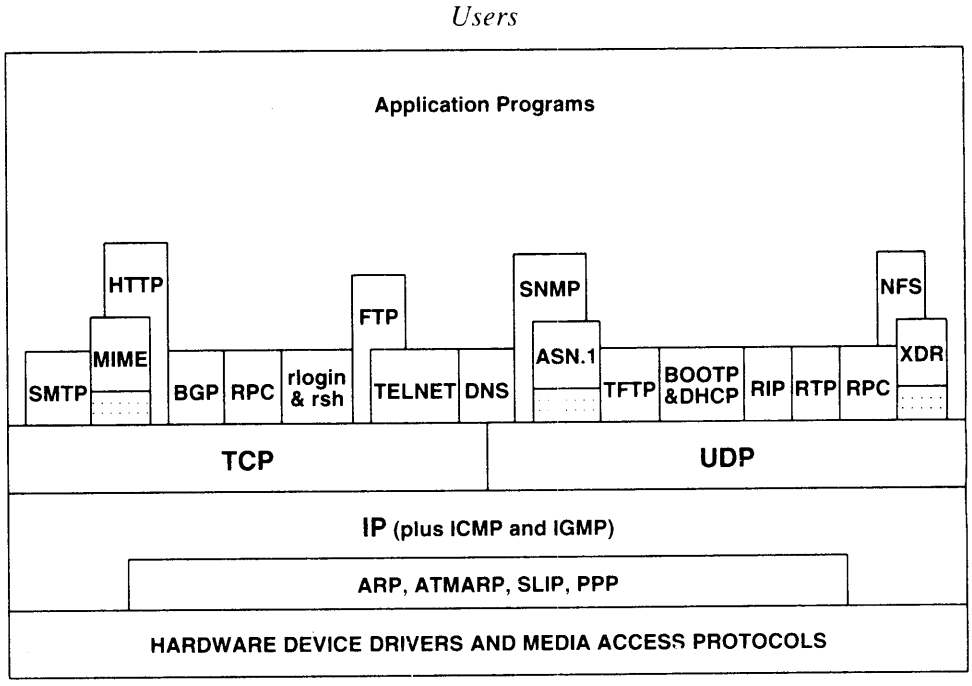


Figure 31.1 Dependencies among major, higher level TCP/IP protocols. A protocol uses the protocols that lie directly below it. Application programs can use all protocols above IP.

Several parts of the diagram need further explanation. The bottom layer represents all protocols that the hardware provides. This level includes all hardware control protocols (e.g., media access and logical link allocation). As we have throughout the text, we will assume that any packet transfer system can be included in this layer as long as IP can use it to transfer datagrams. Thus, if a system is configured to send datagrams through a tunnel, the entry to the tunnel is treated like a hardware interface, despite its software implementation.

The second layer from the bottom lists link layer and address resolution protocols like SLIP, PPP, ARP, and ATMARP. Of course, not all networking technologies require such protocols. ARP is used on connectionless broadcast networks such as Ethernet; ATMARP is used on non-broadcast multiple access networks such as ATM; and RARP is seldom used except for diskless machines. Other link layer or address binding protocols can occur at the same level, but none is currently popular.

The third layer from the bottom contains IP. It includes the required error and control message protocol, ICMP, and the optional multicast group management protocol IGMP. Note that IP is the only protocol that spans an entire layer. All lower-level protocols deliver incoming information to IP, and all higher-level protocols must use IP to send outgoing datagrams. IP is shown with a direct dependency on the hardware layer because it needs to use the hardware link or access protocols to transmit datagrams after it uses ARP to bind addresses.

TCP and UDP comprise the transport layer. Of course, new transport protocols have been suggested, but none has been widely adopted yet.

The application layer illustrates the complex dependencies among the various application protocols. Recall, for example, that FTP uses the network virtual terminal definitions from TELNET to define communication on its control connection and TCP to form data connections. Also recall that HTTP uses syntax and content types from MIME. Thus, the diagram shows that FTP depends on both TELNET and TCP and that HTTP depends on both MIME and TCP. The domain name system (DNS) uses both UDP and TCP for communication, so the diagram shows both dependencies. Sun's NFS depends on the external data representation (XDR) and remote procedure call (RPC) protocols. RPC appears twice because, like the domain name system, it can use either UDP or TCP.

SNMP depends on *Abstract Syntax Notation* (ASN.1). Although SNMP can use either UDP or TCP, only dependence on UDP is shown because few implementations run over TCP. Because XDR, ASN.1, and MIME simply describe syntactic conventions and data representations, they do not use either TCP or UDP. Thus, although it shows that both SNMP and NFS depend on UDP, the diagram contains a dotted area below ASN.1 and XDR to show that neither of them depends on UDP. A few details have been omitted in our diagram. For example, it could be argued that IP depends on BOOTP/DHCP or that many protocols depend on DNS because software that implements such protocols requires name binding.

31.3 The Hourglass Model

Engineers describe Internet protocols as following an *hourglass model*. Because it lies at the heart of all communication, IP forms the center of the hourglass. Of all the protocols we discussed, IP is the only protocol common to all applications — ultimately all internet communication involves IP datagrams. Thus, universal interoperability is achieved by making IP run over all possible network technologies. Figure 31.2 illustrates the concept by showing the dependency among IP, applications, and underlying networks.

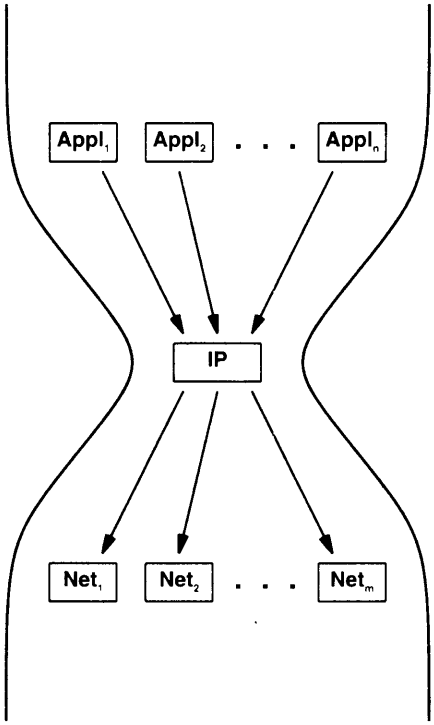


Figure 31.2 An illustration of the hourglass model. IP is at the center of the hourglass because all applications depend on IP and IP runs over all networks.

31.4 Application Program Access

Most systems restrict application programs from accessing lower-level protocols. That is, most systems only allow an application program to access TCP or UDP or to implement higher level protocols that use them (e.g., FTP). In fact, a system may choose to restrict access to transport protocols by allowing only privileged applications to open lower numbered TCP or UDP protocol ports.

Although direct access from an application to IP is unusual, a few systems do provide special purpose mechanisms that make it possible. For example, a mechanism known as a *packet filter* allows privileged programs to control frame demultiplexing. Using the packet filter primitives, an application program establishes the criteria used to capture packets (e.g., the application program can specify that it wishes to capture all packets with a given value in the *type* field of a frame). Once the operating system accepts the filter command, it places all packets that match the specified type on a queue. The application program uses the packet filter mechanism to extract packets from the

queue. For such systems, the diagram in Figure 31.1 should be extended to show application access to lower layers.

31.5 Summary

Much of the rich functionality associated with the TCP/IP protocol suite results from a variety of high-level services supplied by application programs. The high-level protocols these programs use build on the basic transport services: unreliable datagram delivery and reliable stream transport. The applications usually follow the client-server model in which servers operate at known protocols ports so clients know how to contact them.

The highest level of protocols provides user services like Web browsing, remote login, and file and mail transfer. The chief advantages of having an internet on which to build such services are that it provides universal connectivity and simplifies the application protocols. In particular, when used by two machines that attach to an internet, end-to-end transport protocols can guarantee that a client program on the source machine communicates directly with a server on the destination machine. Because services like electronic mail use the end-to-end transport connection, they do not need to rely on intermediate machines to forward (whole) messages.

We have seen a variety of application level protocols and the complex dependencies among them. Although many application protocols have been defined, a few major applications such as Web browsing account for most packets on the Internet.

FOR FURTHER STUDY

One of the issues underlying protocol layering revolves around the optimal location of protocol functionality. Edge [1979] compares end-to-end protocols with the hop-by-hop approach. Saltzer, Reed, and Clark [1984] argues for having the highest level protocols perform end-to-end acknowledgement and error detection. A series of papers by Mills [RFCs 956, 957, and 958] proposes application protocols for clock synchronization, and report on experiments.

EXERCISES

31.1 It is possible to translate some application protocols into others. For example, it might be possible to build a program that accepts an FTP request, translates it to a TFTP request, passes the result to a TFTP server to obtain a file, and translates the reply back to FTP for transmission to the original source. What are the advantages and disadvantages of such protocol translation?

31.2 Consider the translation described in the previous question. Which pairs of protocols in Figure 31.1 are amenable to such translations?

31.3 Some application programs invoked by users may need access to IP without using TCP or UDP. Find examples of such programs. (Hint: think of ICMP.)

31.4 Where do multicast protocols fit into the diagram in Figure 31.1?

31.5 DNS allows access by both TCP and UDP. Find out whether your local operating system allows a single process to accept both TCP connections and UDP requests.

31.6 Choose a complex application like the *X window system*, and find out which protocols it uses.

31.7 Where does OSPF fit into the diagram in Figure 31.1?

31.8 The diagram in Figure 31.1 shows that FTP depends on TELNET. Does your local FTP client invoke the TELNET program, or does the FTP client contain a separate implementation of the TELNET protocol?

31.9 Redraw Figure 31.1 for a Web browser. Which protocols does it use?

32

Internet Security And Firewall Design (IPsec)

32.1 Introduction

Like the locks used to help keep tangible property secure, computers and data networks need provisions that help keep information secure. Security in an internet environment is both important and difficult. It is important because information has significant value — information can be bought and sold directly or used indirectly to create new products and services that yield high profits. Security in an internet is difficult because security involves understanding when and how participating users, computers, services, and networks can trust one another as well as understanding the technical details of network hardware and protocols. Security is required on every computer and every protocol; a single weakness can compromise the security of an entire network. More important, because TCP/IP supports a wide diversity of users, services, and networks and because an internet can span many political and organizational boundaries, participating individuals and organizations may not agree on a level of trust or policies for handling data.

This chapter considers two fundamental techniques that form the basis for internet security: perimeter security and encryption. Perimeter security allows an organization to determine the services and networks it will make available to outsiders and the extent to which outsiders can use resources. Encryption handles most other aspects of security. We begin by reviewing a few basic concepts and terminology.

32.2 Protecting Resources

The terms *network security* and *information security* refer in a broad sense to confidence that information and services available on a network cannot be accessed by unauthorized users. Security implies safety, including assurance of data integrity, freedom from unauthorized access of computational resources, freedom from snooping or wiretapping, and freedom from disruption of service. Of course, just as no physical property is absolutely secure against crime, no network is completely secure. Organizations make an effort to secure networks for the same reason they make an effort to secure buildings and offices: basic security measures can discourage crime by making it significantly more difficult.

Providing security for information requires protecting both physical and abstract resources. Physical resources include passive storage devices such as disks and CD-ROMs as well as active devices such as users' computers. In a network environment, physical security extends to the cables, bridges, and routers that comprise the network infrastructure. Indeed, although physical security is seldom mentioned, it often plays an important role in an overall security plan. Obviously, physical security can prevent wiretapping. Good physical security can also eliminate sabotage (e.g., disabling a router to cause packets to be routed through an alternative, less secure path).

Protecting an abstract resource such as information is usually more difficult than providing physical security because information is elusive. Information security encompasses many aspects of protection:

- *Data integrity.* A secure system must protect information from unauthorized change.
- *Data availability* The system must guarantee that outsiders cannot prevent legitimate access to data (e.g., any outsider should not be able to block customers from accessing a Web site).
- *Privacy or confidentiality.* The system must prevent outsiders from making copies of data as it passes across a network or understanding the contents if copies are made.
- *Authorization.* Although physical security often classifies people and resources into broad categories, (e.g., all nonemployees are forbidden from using a particular hallway), security for information usually needs to be more restrictive (e.g., some parts of an employee's record are available only to the personnel office, others are available only to the employee's boss, and others are available to the payroll office).
- *Authentication.* The system must allow two communicating entities to validate each other's identity.
- *Replay avoidance.* To prevent outsiders from capturing copies of packets and using them later, the system must prevent a retransmitted copy of a packet from being accepted.

32.3 Information Policy

Before an organization can enforce network security, the organization must assess risks and develop a clear policy regarding information access and protection. The policy specifies who will be granted access to each piece of information, the rules an individual must follow in disseminating the information to others, and a statement of how the organization will react to violations.

An information policy begins with people because:

> *Humans are usually the most susceptible point in any security scheme. A worker who is malicious, careless, or unaware of an organization's information policy can compromise the best security.*

32.4 Internet Security

Internet security is difficult because datagrams traveling from source to destination often pass across many intermediate networks and through routers that are not owned or controlled by either the sender or the recipient. Thus, because datagrams can be intercepted or compromised, the contents cannot be trusted. As an example, consider a server that attempts to use *source authentication* to verify that requests originated from valid customers. Source authentication requires the server to examine the source IP address on each incoming datagram, and only accept requests from computers on an authorized list. Source authentication is *weak* because it can be broken easily. In particular, an intermediate router can watch traffic traveling to and from the server, and record the IP address of a valid customer. Later the intermediate router can manufacture a request that has the same source address (and intercept the reply). The point is:

> *An authorization scheme that uses a remote machine's IP address to authenticate its identity does not suffice in an unsecure internet. An imposter who gains control of an intermediate router can obtain access by impersonating an authorized client.*

Stronger authentication requires *encryption*. To encrypt a message, the sender applies a mathematical function that rearranges the bits according to a *key* which is known only to the sender. The receiver uses another mathematical function to decrypt the message. Careful choices of an encryption algorithm, a key, and the contents of messages can make it virtually impossible for intermediate machines to decode messages or manufacture messages that are valid.

32.5 IP Security (IPsec)

The IETF has devised a set of protocols that provide secure Internet communication. Collectively known as *IPsec* (short for *IP security*), the protocols offer authentication and privacy services at the IP layer, and can be used with both IPv4 and IPv6†. More important, instead of completely specifying the functionality or the encryption algorithm to be used, the IETF chose to make the system both flexible and extensible. For example, an application that employs IPsec can choose whether to use an authentication facility that validates the sender or to use an encryption facility that also ensures the payload will remain confidential; the choices can be asymmetric (e.g., authentication in one direction but not the other). Furthermore, IPsec does not restrict the user to a specific encryption or authentication algorithm. Instead, IPsec provides a general framework that allows each pair of communicating endpoints to choose algorithms and parameters (e.g., key size). To guarantee interoperability, IPsec does include a set of encryption algorithms that all implementations must recognize. The point is:

> *IPsec is not a single security protocol. Instead, IPsec provides a set of security algorithms plus a general framework that allows a pair of communicating entities to use whichever algorithms provide security appropriate for the communication.*

32.6 IPsec Authentication Header

Instead of changing the basic datagram header or creating an IP option, IPsec uses a separate *Authentication Header* (*AH*) to carry authentication information. Figure 32.1 illustrates the most straightforward use of an authentication header with IPv4.

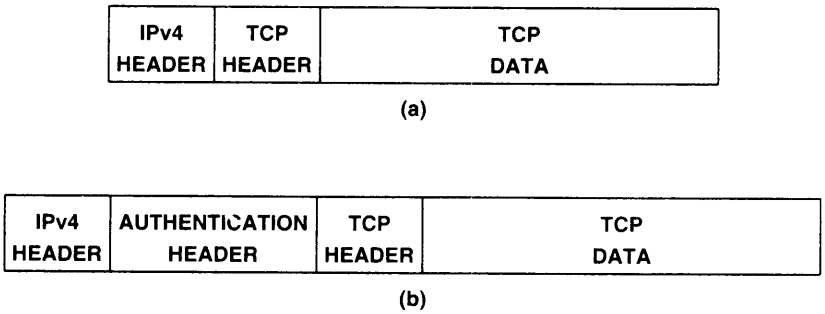


Figure 32.1 Illustration of (a) an IPv4 datagram, and (b) the same datagram after an IPsec authentication header has been added. The new header is inserted immediately after the IP header.

†The examples in this chapter focus on IPv4; Chapter 33 describes IPv6 in detail and illustrates how IPsec headers appear in IPv6 datagrams.

As the figure shows, IPsec inserts the authentication header immediately after the original IP header, but before the transport header. Furthermore, the *PROTOCOL* field in the IP header is changed to value *51* to indicate the presence of an authentication header.

If IPsec modifies the *PROTOCOL* field in the IP header, how does a receiver determine the type of information carried in the datagram? The authentication header has a *NEXT HEADER* field that specifies the type — IPsec records the original *PROTOCOL* value in the *NEXT HEADER* field. When a datagram arrives, the receiver uses security information from the authentication header to verify the sender, and uses the *NEXT HEADER* value to further demultiplex the datagram. Figure 32.2 illustrates the header format.

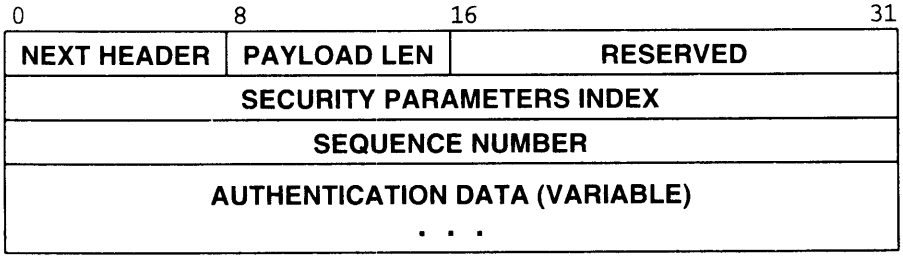


Figure 32.2 The IPsec authentication header format. The field labeled *NEXT HEADER* records the original value of the IP *PROTOCOL* field.

Interestingly, the *PAYLOAD LEN* field does not specify the size of the data area in the datagram. Instead, it specifies the length of the authentication header. Remaining fields are used to ensure security. Field *SEQUENCE NUMBER* contains a unique sequence number for each packet sent; the number starts at zero when a particular security algorithm is selected and increases monotonically. The *SECURITY PARAMETERS INDEX* field specifies the security scheme used, and the *AUTHENTICATION DATA* field contains data for the selected security scheme.

32.7 Security Association

To understand the reason for using a security parameters index, observe that a security scheme defines details that provide many possible variations. For example, the security scheme includes an authentication algorithm, a key (or keys) that the algorithm uses, a lifetime over which the key will remain valid, a lifetime over which the destination agrees to use the algorithm, and a list of source addresses that are authorized to use the scheme. Further observe that the information cannot fit into the header.

To save space in the header, IPsec arranges for each receiver to collect all the details about a security scheme into an abstraction known as a *security association* (*SA*).

Each SA is given a number, known as a *security parameters index*, through which it is identified. Before a sender can use IPsec to communicate with a receiver, the sender must know the index value for a particular SA. The sender then places the value in the field *SECURITY PARAMETERS INDEX* of each outgoing datagram.

Index values are not globally specified. Instead, each destination creates as many SAs as it needs, and assigns an index value to each. The destination can specify a lifetime for each SA, and can reuse index values once an SA becomes invalid. Consequently, the index cannot be interpreted without consulting the destination (e.g., the index *1* can have entirely different meanings to two destinations). To summarize:

> *A destination uses the security parameters index to identify the security association for a packet. The values are not global; a combination of destination address and security parameters index is needed to identify an SA.*

32.8 IPsec Encapsulating Security Payload

To handle privacy as well as authentication, IPsec uses an *Encapsulating Security Payload* (*ESP*), which is more complex than an authentication header. A value *50* in the *PROTOCOL* field of the datagram informs a receiver that the datagram carries ESP. Figure 32.3 illustrates the conceptual organization.

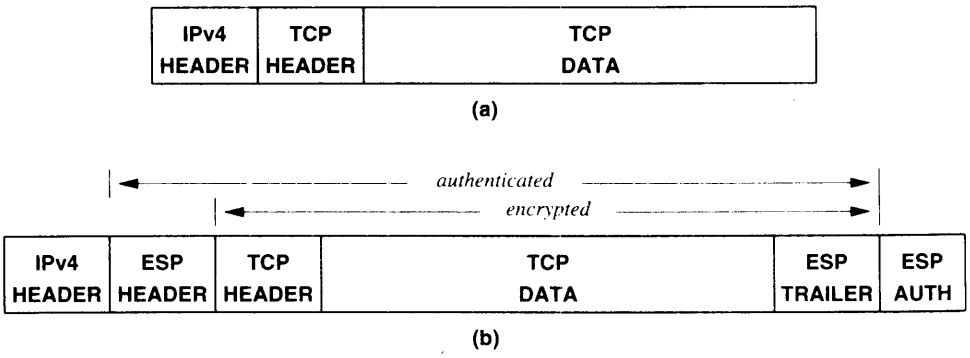
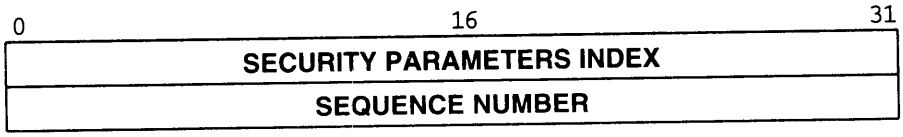


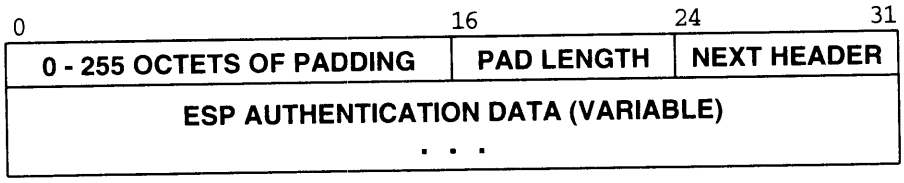
Figure 32.3 (a) A datagram, and (b) the same datagram using IPsec Encapsulating Security Payload. In practice, encryption means that fields are not easily identifiable.

As the figure shows, ESP adds three additional areas to the datagram. The *ESP HEADER* immediately follows the IP header and precedes the encrypted payload. The *ESP TRAILER* is encrypted along with the payload; a variable-size *ESP AUTH* field follows the encrypted section.

ESP uses many of the same items found in the authentication header, but rearranges their order. For example, the *ESP HEADER* consists of 8 octets that identify the security parameters index and a sequence number.



The *ESP TRAILER* consists of optional padding, a padding length field, *PAD LENGTH*, and a *NEXT HEADER* field that is followed by a variable amount of authentication data.



Padding is optional; it may be present for three reasons. First, some decryption algorithms require zeroes following an encrypted message. Second, note that the *NEXT HEADER* field is shown right-justified within a 4-octet field. The alignment is important because IPsec requires the authentication data that follows the trailer to be aligned at the start of a 4-octet boundary. Thus, padding may be needed to ensure alignment. Third, some sites may choose to add random amounts of padding to each datagram so eavesdroppers at intermediate points along the path cannot use the size of a datagram to guess its purpose.

32.9 Authentication And Mutable Header Fields

The IPsec authentication mechanism is designed to ensure that an arriving datagram is identical to the datagram sent by the source. However, such a guarantee is impossible to make. To understand why, recall that IP is a machine-to-machine layer, meaning that the layering principle only applies across one hop. In particular, each intermediate router decrements the time-to-live field and recomputes the checksum.

IPsec uses the term *mutable fields* to refer to IP header fields that are changed in transit. To prevent such changes causing authentication errors, IPsec specifically omits such fields from the authentication computation. Thus, when a datagram arrives, IPsec only authenticates immutable fields (e.g., the source address and protocol type).

32.10 IPsec Tunneling

Recall from Chapter 20 that VPN technology uses encryption along with IP-in-IP tunneling to keep inter-site transfers private. IPsec is specifically designed to accommodate an encrypted tunnel. In particular, the standard defines tunneled versions of both the authentication header and the encapsulating security payload. Figure 32.4 illustrates the layout of datagrams in tunneling mode.

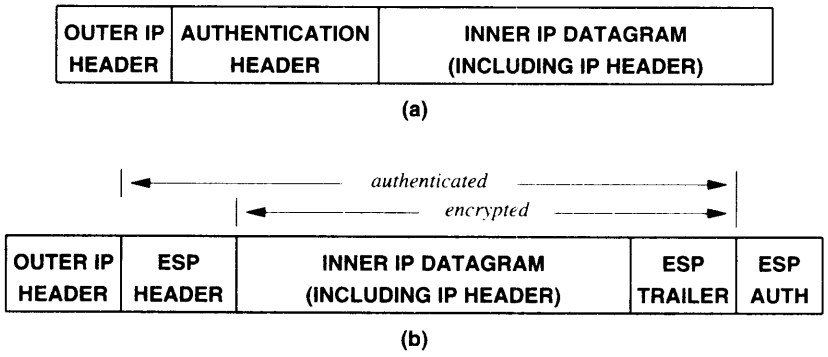


Figure 32.4 Illustration of IPsec tunneling mode for (a) authentication and (b) encapsulating security payload. The entire inner datagram is protected.

32.11 Required Security Algorithms

IPsec defines a minimal set of algorithms that are mandatory (i.e., that all implementations must supply). In each case, the standard defines specific uses. Figure 32.5 lists the required algorithms.

Authentication	
HMAC with MD5	RFC 2403
HMAC with SHA-1	RFC 2404
Encapsulating Security Payload	
DES in CBC mode	RFC 2405
HMAC with MD5	RFC 2403
HMAC with SHA-1	RFC 2404
Null Authentication	
Null Encryption	

Figure 32.5 The security algorithms that are mandatory for IPsec.

32.12 Secure Sockets

By the mid 1990s when it became evident that security was important for Internet commerce, several groups proposed security mechanisms for use with the Web. Although not formally adopted by the IETF, one of the proposals has become a de facto standard.

Known as the *Secure Sockets Layer* (*SSL*), the technology was originally developed by Netscape, Inc. As the name implies, SSL resides at the same layer as the socket API. When a client uses SSL to contact a server, the SSL protocol allows each side to authenticate itself to the other. The two sides then negotiate to select an encryption algorithm that they both support. Finally, SSL allows the two sides to establish an encrypted connection (i.e., a connection that uses the chosen encryption algorithm to guarantee privacy).

32.13 Firewalls And Internet Access

Mechanisms that control *internet access* handle the problem of screening a particular network or an organization from unwanted communication. Such mechanisms can help prevent outsiders from: obtaining information, changing information, or disrupting communication on an organization's intranet. Successful access control requires a careful combination of restrictions on network topology, intermediate information staging, and packet filters.

A single technique known as an *internet firewall*†, has emerged as the basis for internet access control. An organization places a firewall at its connection to external networks (e.g., the global Internet). A firewall partitions an internet into two regions, referred to informally as the *inside* and *outside*.

32.14 Multiple Connections And Weakest Links

Although concept seems simple, details complicate firewall construction. First, an organization's intranet can have multiple external connections. The organization must form a *security perimeter* by installing a firewall at each external connection. To guarantee that the perimeter is effective, all firewalls must be configured to use exactly the same access restrictions. Otherwise, it may be possible to circumvent the restrictions imposed by one firewall by entering the organization's internet through another‡.

We can summarize:

> *An organization that has multiple external connections must install a firewall on each external connection and must coordinate all firewalls. Failure to restrict access identically on all firewalls can leave the organization vulnerable.*

---

†The term *firewall* is derived from building architecture in which a firewall is a thick, fireproof partition that makes a section of a building impenetrable to fire.

‡The well-known idea that security is only as strong as the weakest point has been termed the *weakest link axiom* in reference to the adage that a chain is only as strong as its weakest link.

32.15 Firewall Implementation

How should a firewall be implemented? In theory, a firewall simply blocks all unauthorized communication between computers in the organization and computers outside the organization. In practice, the details depend on the network technology, the capacity of the connection, the traffic load, and the organization's policies. Thus, no single solution works for all organizations; building an effective, customized firewall can be difficult.

To operate at network speeds, a firewall must have hardware and software optimized for the task. Fortunately, most commercial routers include a high-speed filtering mechanism that can be used to perform much of the necessary work. A manager can configure the filter in a router to request that the router block specified datagrams. As we discuss the details of filter mechanisms, we will see how filters form the basic building blocks of a firewall. Later we will see how filters can be used in conjunction with another mechanism to provide communication that is safe, but flexible.

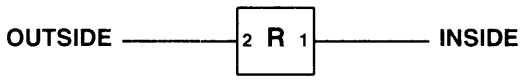
32.16 Packet-Level Filters

Many commercial routers offer a mechanism that augments normal routing and permits a manager to further control packet processing. Informally called a *packet filter*, the mechanism requires the manager to specify how the router should dispose of each datagram. For example, the manager might choose to *filter* (i.e. block) all datagrams that come from a particular source or those used by a particular application, while choosing to route other datagrams to their destination.

The term *packet filter* arises because the filtering mechanism does not keep a record of interaction or a history of previous datagrams. Instead, the filter considers each datagram separately. When a datagram first arrives, the router passes the datagram through its packet filter before performing any other processing. If the filter rejects the datagram, the router drops it immediately.

Because TCP/IP does not dictate a standard for packet filters, each router vendor is free to choose the capabilities of their packet filter as well as the interface a manager uses to configure the filter. Some routers permit a manager to configure separate filter actions for each interface, while others have a single configuration for all interfaces. Usually, when specifying datagrams that the filter should block, a manager can list any combination of source IP address, destination IP address, protocol, source protocol port number, and destination protocol port number. For example, Figure 32.6 illustrates a filter specification.

In the example, the manager has chosen to block incoming datagrams destined for a few well-known services and to block one case of outgoing datagrams. The filter blocks all outgoing datagrams that originate from any host address matching the 16-bit prefix of *128.5.0.0* that are destined for a remote e-mail server (TCP port *25*). The filter also blocks incoming datagrams destined for FTP (TCP port *21*), TELNET (TCP port *23*), WHOIS (UDP port *43*), TFTP (UDP port *69*), or FINGER (TCP port *79*).



ARRIVES ON INTERFACE	IP SOURCE	IP DEST.	PROTOCOL	SOURCE PORT	DEST. PORT
2	*	*	TCP	*	21
2	*	*	TCP	*	23
1	128.5.0.0/16	*	TCP	*	25
2	*	*	UDP	*	43
2	*	*	UDP	*	69
2	*	*	TCP	*	79

Figure 32.6 A router with two interfaces and an example datagram filter specification. A router that includes a packet filter forms the basic building block of a firewall.

32.17 Security And Packet Filter Specification

Although the example filter configuration in Figure 32.6 specifies a small list of services that should be blocked, such an approach does not work well for an effective firewall. There are three reasons. First, the number of well-known ports is large and growing rapidly. Thus, listing each service requires a manager to update the list continually; an error of omission can leave the firewall vulnerable. Second, much of the traffic on an internet does not travel to or from a well-known port. In addition to programmers who can choose port numbers for their private client-server applications, services like *Remote Procedure Call* (*RPC*) assign ports dynamically. Third, listing ports of well-known services leaves the firewall vulnerable to *tunneling*. Tunneling can circumvent security if a host or router on the inside agrees to accept encapsulated datagrams from an outsider, remove one layer of encapsulation, and forward the datagram on to the service that would otherwise be restricted by the firewall.

How can a firewall use a packet filter effectively? The answer lies in reversing the idea of a filter: instead of specifying the datagrams that should be filtered, a firewall should be configured to block all datagrams except those destined for specific networks, hosts, and protocol ports for which external communication has been approved. Thus, a manager begins with the assumption that communication is not allowed, and then must examine the organization's information policy carefully before enabling any port. In fact, many packet filters allow a manager to specify a set of datagrams to admit instead of a set of datagrams to block. We can summarize:

To be effective, a firewall that uses datagram filtering should restrict access to all IP sources, IP destinations, protocols, and protocol ports except those computers, networks, and services the organization explicitly decides to make available externally. A packet filter that allows a manager to specify which datagrams to admit instead of which datagrams to block can make such restrictions easy to specify.

32.18 The Consequence Of Restricted Access For Clients

A blanket prohibition on datagrams arriving for an unknown protocol port seems to solve many potential security problems by preventing outsiders from accessing arbitrary servers in the organization. Such a firewall has an interesting consequence: it also prevents an arbitrary computer inside the firewall from becoming a client that accesses a service outside the firewall. To understand why, recall that although each server operates at a well-known port, a client does not. When a client program begins execution, it requests the operating system to select a protocol port number that is neither among the well-known ports nor currently in use on the client's computer. When it attempts to communicate with a server outside the organization, a client will generate one or more datagrams and send them to the server. Each outgoing datagram has the client's protocol port as the source port and the server's well-known protocol port as the destination port. The firewall will not block such datagrams as they leave. When it generates a response, the server reverses the protocol ports. The client's port becomes the destination port and the server's port becomes the source port. When the datagram carrying the response reaches the firewall, however, it will be blocked because the destination port is not approved. Thus, we can see an important idea:

If an organization's firewall restricts incoming datagrams except for ports that correspond to services the organization makes available externally, an arbitrary application inside the organization cannot become a client of a server outside the organization.

32.19 Proxy Access Through A Firewall

Of course, not all organizations configure their firewalls to block all datagrams destined for unknown protocol ports. In cases where a secure firewall is needed to prevent unwanted access, however, users on the inside need a safe mechanism that provides access to services outside. That mechanism forms the second major piece of firewall architecture.

In general, an organization can only provide safe access to outside services through a secure computer. Instead of trying to make all computer systems in the organization secure (a daunting task), an organization usually associates one secure computer with

each firewall, and installs a set of application gateways on that computer. Because the computer must be strongly fortified to serve as a secure communication channel, it is often called a *bastion host*. Figure 32.7 illustrates the concept.

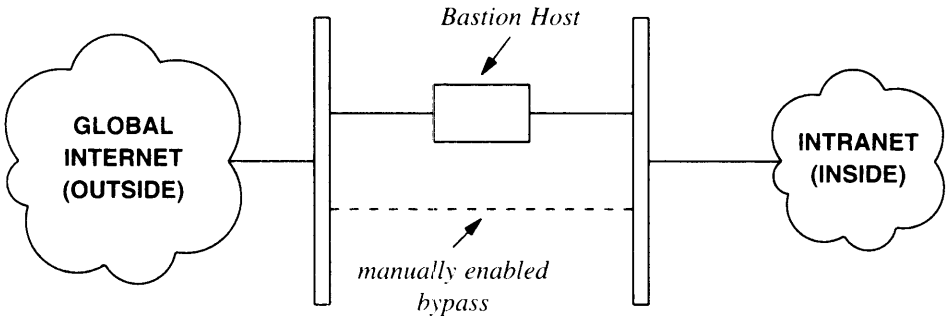


Figure 32.7 The conceptual organization of a bastion host embedded in a firewall. The bastion host provides secure access to outside services without requiring an organization to admit datagrams with arbitrary destinations.

As the figure shows, the firewall has two conceptual barriers. The outer barrier blocks all incoming traffic except (1) datagrams destined for services on the bastion host that the organization chooses to make available externally, and (2) datagrams destined for clients on the bastion host. The inner barrier blocks incoming traffic except datagrams that originate on the bastion host. Most firewalls also include a *manual bypass* that enables managers to temporarily pass some or all traffic between a host inside the organization and a host outside (e.g., for testing or debugging the network).

To understand how a bastion host operates, consider Web access. Because the firewall prevents the user's computer from receiving incoming datagrams, the user cannot use a browser for direct access. Instead, the organization arranges a proxy server on the bastion host. Inside the organization, each browser is configured to use the proxy. Whenever a user selects a link or enters a URL, their browser contacts the proxy. The proxy contacts the server, obtains the specified page, and then delivers it internally.

32.20 The Details Of Firewall Architecture

Now that we understand the basic firewall concept, the implementation should appear straightforward. Conceptually, each of the barriers shown in Figure 32.7 requires a router that has a packet filter†. Networks interconnect the routers and a bastion host. For example, an organization that connects to the global Internet might choose to implement a firewall as Figure 32.8 shows.

†Some organizations use a *one-armed firewall* configuration in which a single physical router implements all the functionality.

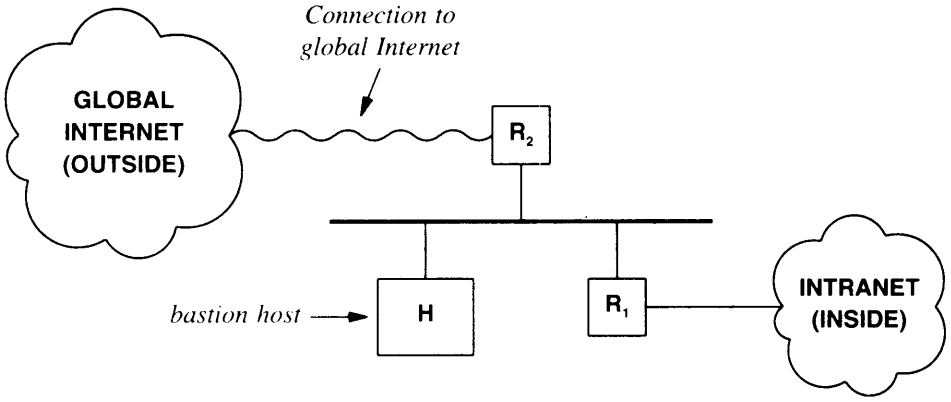


Figure 32.8 A firewall implemented with two routers and a bastion host. One of the routers has a connection to the rest of the Internet.

As the figure shows, router R_2 implements the outer barrier; it filters all traffic except datagrams destined for the bastion host, H. Router R_1 implements the inner barrier that isolates the rest of the corporate intranet from outsiders; it blocks all incoming datagrams except those that originate on the bastion host.

Of course, the safety of an entire firewall depends on the safety of the bastion host. If an intruder can gain access to the computer system running on the bastion host, they will gain access to the entire inside internet. Moreover, an intruder can exploit security flaws in either the operating system on the bastion host or the network applications it runs. Thus, managers must be particularly careful when choosing and configuring software for a bastion host. In summary:

> *Although a bastion host is essential for communication through a firewall, the security of the firewall depends on the safety of the bastion host. An intruder who exploits a security flaw in the bastion host operating system can gain access to hosts inside the firewall.*

32.21 Stub Network

It may seem that Figure 32.8 contains a superfluous network that connects the two routers and the bastion host. Such a network is often called a *stub network* because it is small (i.e., stubby). The question arises, "Is the stub network necessary or could a site place the bastion host on one of its production networks?" The answer depends on the traffic expected from the outside. The stub network isolates the organization from incoming datagram traffic. In particular, because router R_2 admits all datagrams destined for the bastion host, an outsider can send an arbitrary number of such datagrams across

the stub network. If an external connection is slow relative to the capacity of a stub network, a separate physical wire may be unnecessary. However, a stub network is usually an inexpensive way for an organization to protect itself against disruption of service on an internal production network.

32.22 An Alternative Firewall Implementation

The firewall implementation in Figure 32.8 works well for an organization that has a single serial connection to the rest of the global Internet. Some sites have a different interconnection topology. For example, suppose a company has three or four large customers who each need to deposit or extract large volumes of information. The company wishes to have a single firewall, but allow connections to multiple sites†. Figure 32.9 illustrates one possible firewall architecture that accommodates multiple external connections.

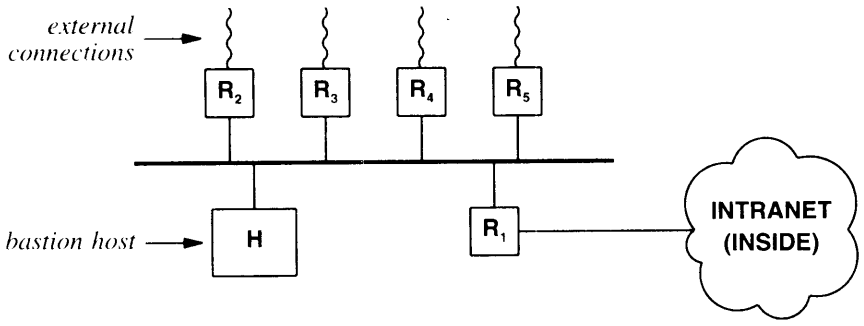


Figure 32.9 An alternative firewall architecture that permits multiple external connections through a single firewall. Using one firewall for multiple connections can reduce the cost.

As the figure shows, the alternative architecture extends a firewall by providing an outer network at which external connections terminate. Router R_1 acts as in Figure 32.8 to protect the site by restricting incoming datagrams to those sent from the bastion host. Routers R_2 through R_5 each connect one external site to the firewall.

To understand why firewalls with multiple connections often use a router per connection, recall that all sites mistrust one another. That is, the organization running the firewall does not trust any of the external organizations completely, and none of the external organizations trust one another completely. The packet filter in a router on a given external connection can be configured to restrict traffic on that particular connection. As a result, the owner of the firewall can guarantee that although all external connections share a single, common network, no datagram from one external connection will pass to another. Thus, the organization running the firewall can assure customers that it is safe to connect. To summarize:

†A single firewall can be less expensive and easier to administrate than a separate firewall per connection.

When multiple external sites connect through a single firewall, an architecture that has a router per external connection can prevent unwanted packet flow from one external site to another.

32.23 Monitoring And Logging

Monitoring is one of the most important aspects of a firewall design. The network manager responsible for a firewall needs to be aware of attempts to bypass security. Unless a firewall reports incidents, a manager may be unaware of problems.

Monitoring can be *active* or *passive*. In active monitoring, a firewall notifies a manager whenever an incident occurs. The chief advantage of active monitoring is speed — a manager finds out about a potential problem immediately. The chief disadvantage is that active monitors often produce so much information that a manager cannot comprehend it or notice problems. Thus, most managers prefer passive monitoring, or a combination of passive monitoring with a few high-risk incidents also reported by an active monitor.

In passive monitoring, a firewall logs a record of each incident in a file on disk. A passive monitor usually records information about normal traffic (e.g., simple statistics) as well as datagrams that are filtered. A manager can access the log at any time; most managers use a computer program. The chief advantage of passive monitoring arises from its record of events — a manager can consult the log to observe trends and when a security problem does occur, review the history of events that led to the problem. More important, a manager can analyze the log periodically (e.g., daily) to determine whether attempts to access the organization increase or decrease over time.

32.24 Summary

Security problems arise because an internet can connect organizations that do not have mutual trust. Several technologies are available to help ensure that information remains secure when being sent across an internet. IPsec allows a user to choose between two basic schemes: one that provides authentication of the datagram and one that provides authentication plus privacy. IPsec modifies a datagram either by inserting an Authentication Header or by using an Encapsulating Security Payload, which inserts a header and trailer and encrypts the data being sent. IPsec provides a general framework that allows each pair of communicating entities to choose an encryption algorithm. Because security is often used with tunneling (e.g., in a VPN), IPsec defines a secure tunnel mode.

The firewall mechanism is used to control internet access. An organization places a firewall at each external connection to guarantee that the organization's intranet remains free from unauthorized traffic. A firewall consists of two barriers and a secure computer called a bastion host. Each barrier uses a packet filter to restrict datagram traffic. The bastion host offers externally-visible servers, and runs proxy servers that al-

low users to access outside servers. The filters are configured according to the organization's information policy. Usually, the firewall blocks all datagrams arriving from external sources except those datagrams destined for the bastion host.

A firewall can be implemented in one of several ways; the choice depends on details such as the number of external connections. In many cases, each barrier in a firewall is implemented with a router that contains a packet filter. A firewall can also use a stub network to keep external traffic off an organization's production networks.

FOR FURTHER STUDY

In the mid 1990s, the IETF announced a major emphasis on security, and required each working group to consider the security implications of its designs. Consequently, many RFCs address issues of internet security and propose policies, procedures, and mechanisms. Kent and Atkinson [RFC 2401] defines the IPsec architecture. Kent and Atkinson [RFC 2402] specifies the IPsec authentication header, and [RFC 2406] specifies the encapsulating security payload.

Many RFCs describe security for particular application protocols. For example, Wijnen et. al. [RFC 2575] presents the view-based security and Blumenthal and Wijnen [RFC 2574] presents a user-based security model, both are intended for use with SNMPv3.

Cheswick and Bellovin [1994] discusses firewalls and other topics related to the secure operation of TCP/IP internets. Kohl and Neuman [RFC 1510] describes the *kerberos* authentication service, and Borman [RFC 1411] discusses how *kerberos* can be used to authenticate TELNET.

EXERCISES

32.1 Many sites that use a bastion host arrange for software to scan all incoming files before admitting them to the organization. Why do organizations scan files?

32.2 Read the description of a packet filter for a commercially available router. What features does it offer?

32.3 Collect a log of all traffic entering your site. Analyze the log to determine the percentage of traffic that arrives from or is destined to a well-known protocol port. Do the results surprise you?

32.4 If encryption software is available on your computer, measure the time required to encrypt a 10 Mbyte file, transfer it to another computer, and decrypt it. Compare the result to the time required for the transfer if no encryption is used.

32.5 Survey users at your site to determine if they send sensitive information in e-mail. Are users aware that SMTP transfers messages in ASCII, and that anyone watching network traffic can see the contents of an e-mail message?

32.6 Survey employees at your site to find out how many use modems and personal computers to import or export information. Ask if they understand the organization's information policy.

32.7 Can a firewall be used with other protocol suites such as AppleTalk or Netware? Why or why not?

32.8 Can a firewall be combined with NAT? What are the consequences?

32.9 The military only releases information to those who "need to know." Will such a scheme work for all information in your organization? Why or why not?

32.10 Give two reasons why the group of people who administer an organization's security policies should be separate from the group of people who administer the organization's computer and network systems.

32.11 Some organizations use firewalls to isolate groups of users internally. Give examples of ways that internal firewalls can improve network performance and examples of ways internal firewalls can degrade network performance.

32.12 If your organization uses IPsec, find out which algorithms are being used. What is the key size?

33

The Future Of TCP/IP (IPv6)

33.1 Introduction

Evolution of TCP/IP technology is intertwined with evolution of the global Internet for several reasons. First, the Internet is the largest installed TCP/IP internet, so many problems related to scale arise in the Internet before they surface in other TCP/IP internets. Second, funding for TCP/IP research and engineering comes from companies and government agencies that use the operational Internet, so they tend to fund projects that impact the Internet. Third, because most researchers use the global Internet daily, they have immediate motivation to solve problems that will improve service and extend functionality.

With millions of users at tens of thousands of sites around the world depending on the global Internet as part of their daily work environment, it might appear that the Internet is a completely stable production facility. We have passed the early stage of development in which every user was also an expert, and entered a stage in which few users understand the technology. Despite appearances, however, neither the Internet nor the TCP/IP protocol suite is static. Groups discover new ways to use the technology. Researchers solve new networking problems, and engineers improve the underlying mechanisms. In short, the technology continues to evolve.

The purpose of this chapter is to consider the ongoing evolutionary process and examine one of the most significant engineering efforts: a proposed revision of IP. When the proposal is adopted by vendors, it will have a major impact on TCP/IP and the global Internet.

33.2 Why Change?

The basic TCP/IP technology has worked well for over two decades. Why should it change? In a broad sense, the motivation revising the protocols arises from changes in underlying technologies and uses.

- *New Computer And Communication Technologies.* Computer and network hardware continues to evolve. As new technologies emerge, they are incorporated into the Internet.

- *New Applications.* As programmers invent new ways to use TCP/IP, additional protocol support is needed. For example, the emphasis on IP telephony has led to investigations of protocols for real-time data delivery.

- *Increases In Size And Load.* The global Internet has experienced many years of sustained exponential growth, doubling in size every nine months or faster. In 1999, on the average, a new host appeared on the Internet every two seconds. Traffic has also increased rapidly as animated graphics and video proliferate.

33.3 New Policies

As it expands into new countries, the Internet changes in a fundamental way: it gains new administrative authorities. Changes in authority produce changes in administrative policies, and mandate new mechanisms to enforce those policies. As we have seen, both the architecture of the connected Internet and the protocols it uses are evolving away from a centralized core model. Evolution continues as more national backbone networks attach, producing increasingly complex policies regulating interaction. When multiple corporations interconnect private TCP/IP internets, they face similar problems as they try to define policies for interaction and then develop mechanisms to enforce those policies. Thus, many of the research and engineering efforts surrounding TCP/IP continue to focus on finding ways to accommodate new administrative groups.

33.4 Motivation For Changing IPv4

Version *4* of the Internet Protocol (*IPv4*) provides the basic communication mechanism of the TCP/IP suite and the global Internet; it has remained almost unchanged since its inception in the late 1970s†. The longevity of version *4* shows that the design is flexible and powerful. Since the time IPv4 was designed, processor performance has increased over two orders of magnitude, typical memory sizes have increased by over a factor of 100, network bandwidth of the Internet backbone has risen by a factor of 7000, LAN technologies have emerged, and the number of hosts on the

†Versions *1* through *3* were never formally assigned, and version number *5* was assigned to the *ST* protocol.

Internet has risen from a handful to over 56 million. Furthermore, because the changes did not occur simultaneously, adapting to them has been a continual process.

Despite its sound design, IPv4 must be replaced soon. Chapter 10 describes the main motivation for updating IP: the imminent address space limitations. When IP was designed, a 32-bit address space was more than sufficient. Only a handful of organizations used a LAN; fewer had a corporate WAN. Now, however, most medium-sized corporations have multiple LANs, and most large corporations have a corporate WAN. Consequently, even with careful assignment and NAT technology, the current 32-bit IP address space cannot accommodate projected growth of the global Internet beyond the year 2020.

Although the need for a larger address space is the most immediate motivation, other factors contributed to the new design. In particular, to make IP better suited to real-time applications, thought was given to supporting systems that associate a datagram with a preassigned resource reservation. To make electronic commerce safer, the next version of IP is designed to include support for security features such as authentication.

33.5 The Road To A New Version Of IP

It took many years for the IETF to formulate a new version of IP. Because the IETF produces *open* standards, it invited the entire community to participate in the process. Computer manufacturers, hardware and software vendors, users, managers, programmers, telephone companies, and the cable television industry all specified their requirements for the next version of IP, and all commented on specific proposals.

Many designs were proposed to serve a particular purpose or a particular community. One of the major proposals would have made IP more sophisticated at the cost of increased complexity and processing overhead. Another design proposed using a modification of the OSI CLNS protocol. A third major design proposed retaining most of the ideas in IP, but making simple extensions to accommodate larger addresses. The design, known as *SIP*† (*Simple IP*), became the basis for an extended proposal that included ideas from other proposals. The extended version was named *Simple IP Plus* (*SIPP*), and eventually emerged as the design selected as a basis for the next IP.

Choosing a new version of IP was not easy. The popularity of the Internet means that the market for IP products around the world is staggering. Many groups see the economic opportunity, and hope that the new version of IP will help them gain an edge over the competition. In addition, personalities have been involved — some individuals hold strong technical opinions; others see active participation as a path to a promotion. Consequently, the discussions generated heated arguments.

†The acronym *SIP* now refers to the *Session Initiation Protocol* which is used for signaling (e.g., for IP telephony).

33.6 The Name Of The Next IP

The IETF decided to assign the revision of IP version number *6* and to name it *IPv6*† to distinguish it from the current *IPv4*. The choice to skip version number *5* arose after a series of mistakes and misunderstandings. In one mistake, the IAB caused widespread confusion by inadvertently publishing a policy statement that referred to the next version of IP as *IP version 7*. In a misunderstanding, an experimental protocol known as the *Stream Protocol* (*ST*) was assigned version number *5*. The assignment led some to conclude that ST had been selected as the replacement for IP. In the end, the IETF chose *6* because doing so eliminated confusion.

33.7 Features Of IPv6

The proposed IPv6 protocol retains many of the features that contributed to the success of IPv4. In fact, the designers have characterized IPv6 as being basically the same as IPv4 with a few modifications. For example, IPv6 still supports connectionless delivery (i.e., each datagram is routed independently), allows the sender to choose the size of a datagram, and requires the sender to specify the maximum number of hops a datagram can make before being terminated. As we will see, IPv6 also retains most of the concepts provided by IPv4 options, including facilities for fragmentation and source routing.

Despite many conceptual similarities, IPv6 changes most of the protocol details. For example, IPv6 uses larger addresses, and adds a few new features. More important, IPv6 completely revises the datagram format by replacing IPv4's variable-length options field by a series of fixed-format headers. We will examine details after considering major changes and the underlying motivation for each.

The changes introduced by IPv6 can be grouped into seven categories:

- *Larger Addresses.* The new address size is the most noticeable change. IPv6 quadruples the size of an IPv4 address from 32 bits to 128 bits. The IPv6 address space is so large that it cannot be exhausted in the foreseeable future.
- *Extended Address Hierarchy.* IPv6 uses the larger address space to create additional levels of addressing hierarchy. In particular, IPv6 can define a hierarchy of ISPs as well as a hierarchical structure within a given site.
- *Flexible Header Format.* IPv6 uses an entirely new and incompatible datagram format. Unlike the IPv4 fixed-format header, IPv6 defines a set of optional headers.
- *Improved Options.* Like IPv4, IPv6 allows a datagram to include optional control information. IPv6 includes new options that provide additional facilities not available in IPv4.

†Some documents refer to the effort as "IP — The Next Generation," *IPng*.

- *Provision For Protocol Extension.* Perhaps the most significant change in IPv6 is a move away from a protocol that fully specifies all details to a protocol that can permit additional features. The extension capability has the potential to allow the IETF to adapt the protocol to changes in underlying network hardware or to new applications.
- *Support For Autoconfiguration And Renumbering.* IPv6 provides facilities that allow computers on an isolated network to assign themselves addresses and begin communicating without depending on a router or manual configuration. The protocol also includes a facility that permits a manager to renumber networks dynamically.
- *Support For Resource Allocation.* IPv6 has two facilities that permit preallocation of network resources: a flow abstraction and a differentiated service specification. The latter will use the same approach as IPv4's differentiated services.

33.8 General Form Of An IPv6 Datagram

IPv6 completely changes the datagram format. As Figure 33.1 shows, an IPv6 datagram has a fixed-size *base header* followed by zero or more *extension headers*, followed by data.

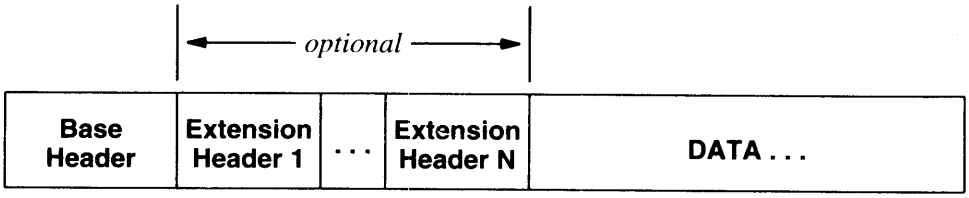


Figure 33.1 The general form of an IPv6 datagram with multiple headers. Only the base header is required; extension headers are optional.

33.9 IPv6 Base Header Format

Interestingly, although it must accommodate larger addresses, an IPv6 base header contains less information than an IPv4 datagram header. Options and some of the fixed fields that appear in an IPv4 datagram header have been moved to extension headers in IPv6. In general, the changes in the datagram header reflect changes in the protocol:

- Alignment has been changed from 32-bit to 64-bit multiples.

- The header length field has been eliminated, and the datagram length field has been replaced by a *PAYLOAD LENGTH* field.
- The size of source and destination address fields has been increased to 16 octets each.
- Fragmentation information has been moved out of fixed fields in the base header into an extension header.
- The *TIME-TO-LIVE* field has been replaced by a *HOP LIMIT* field.
- The *SERVICE TYPE* is renamed to be a *TRAFFIC CLASS* field, and extended with a *FLOW LABEL* field.
- The *PROTOCOL* field has been replaced by a field that specifies the type of the next header.

Figure 33.2 shows the contents and format of an IPv6 base header. Several fields in an IPv6 base header correspond directly to fields in an IPv4 header. As in IPv4, the initial 4-bit *VERS* field specifies the version of the protocol; *VERS* always contains 6 in an IPv6 datagram. As in IPv4, the *SOURCE ADDRESS* and *DESTINATION ADDRESS* fields specify the addresses of the sender and intended recipient. In IPv6, however, each address requires 16 octets. The *HOP LIMIT* field corresponds to the IPv4 *TIME-TO-LIVE* field. Unlike IPv4, which interprets a time-to-live as a combination of hop-count and maximum time, IPv6 interprets the value as giving a strict bound on the maximum number of hops a datagram can make before being discarded.

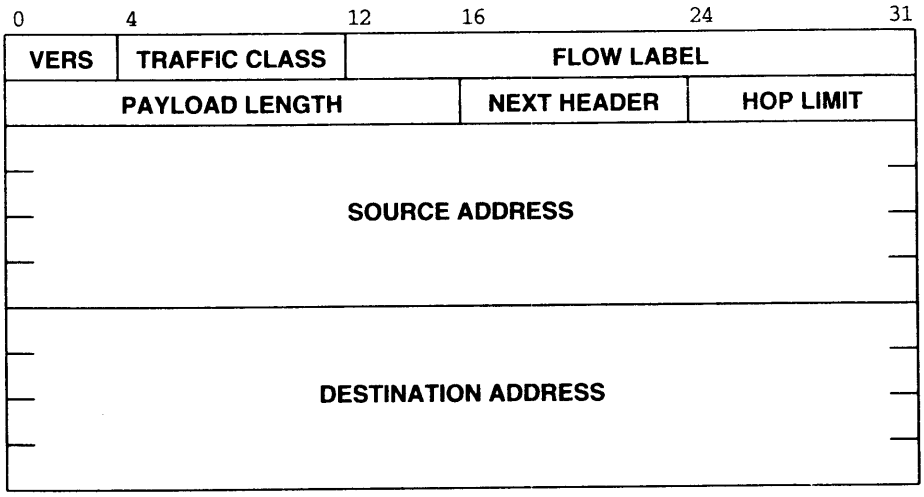


Figure 33.2 The format of the 40-octet IPv6 base header. Each IPv6 datagram begins with a base header.

IPv6 handles datagram length specifications in a new way. First, because the size of the base header is fixed at 40 octets, the base header does not include a field for the header length. Second, IPv6 replaces IPv4's datagram length field by a 16-bit *PAYLOAD LENGTH* field that specifies the number of octets carried in the datagram excluding the header itself. Thus, an IPv6 datagram can contain 64K octets of data.

Two fields in the base header are used in making forwarding decisions. The IPv4 *SERVICE CLASS* field has been renamed *TRAFFIC CLASS*. In addition, a new mechanism in IPv6 supports resource reservation and allows a router to associate each datagram with a given resource allocation. The underlying abstraction, a *flow*, consists of a path through an internet along which intermediate routers guarantee a specific quality of service. Field *FLOW LABEL* in the base header contains information that routers use to associate a datagram with a specific flow and priority. For example, two applications that need to send video can establish a flow on which the delay and bandwidth is guaranteed. Alternatively, a network provider may require a subscriber to specify the quality of service desired, and then use a flow to limit the traffic a specific computer or a specific application sends. Note that flows can also be used within a given organization to manage network resources and ensure that all applications receive a fair share. A router uses the combination of datagram source address and flow identifier when associating a datagram with a specific flow. To summarize:

> *Each IPv6 datagram begins with a 40-octet base header that includes fields for the source and destination addresses, the maximum hop limit, the traffic class, the flow label, and the type of the next header. Thus, an IPv6 datagram must contain at least 40 octets in addition to the data.*

33.10 IPv6 Extension Headers

The paradigm of a fixed base header followed by a set of optional extension headers was chosen as a compromise between generality and efficiency. To be totally general, IPv6 needs to include mechanisms to support functions such as fragmentation, source routing, and authentication. However, choosing to allocate fixed fields in the datagram header for all mechanisms is inefficient because most datagrams do not use all mechanisms; the large IPv6 address size exacerbates the inefficiency. For example, when sending a datagram across a single local area network, a header that contains empty address fields can occupy a substantial fraction of each frame. More important, the designers realize that no one can predict which facilities will be needed.

The IPv6 extension header paradigm works similar to IPv4 options — a sender can choose which extension headers to include in a given datagram and which to omit. Thus, extension headers provide maximum flexibility. We can summarize:

IPv6 extension headers are similar to IPv4 options. Each datagram includes extension headers for only those facilities that the datagram uses.

33.11 Parsing An IPv6 Datagram

Each of the base and extension headers contains a *NEXT HEADER* field. Software on intermediate routers and at the final destination that process a datagram use the values in the *NEXT HEADER* fields to parse the datagram. Extracting all header information from an IPv6 datagram requires a sequential search through the headers. For example, Figure 33.3 shows the *NEXT HEADER* fields of three datagrams that contain zero, one, and two extension headers.

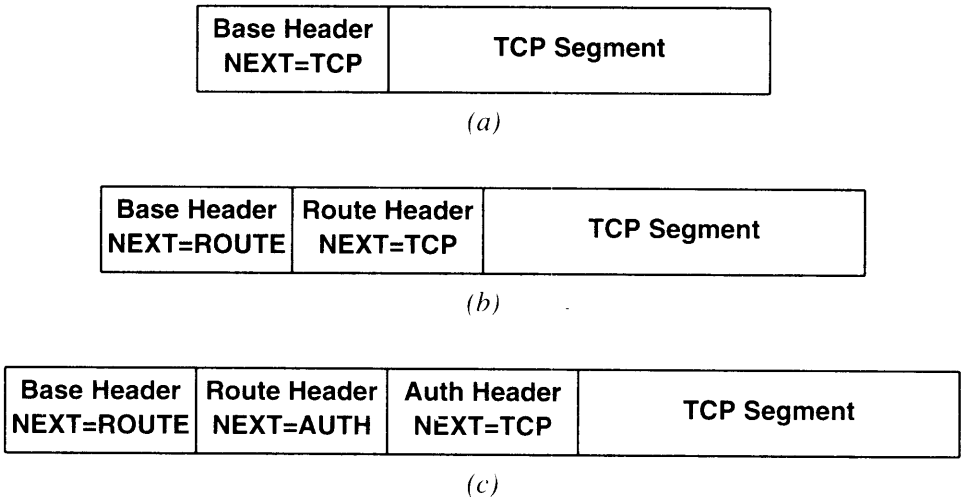


Figure 33.3 Three datagrams with (a) only a base header, (b) a base header and one extension, and (c) a base header plus two extensions. The *NEXT HEADER* field in each header specifies the type of the following header.

Of course, parsing an IPv6 datagram that only has a base header and data is as efficient as parsing an IPv4 datagram. Furthermore, intermediate routers only need to examine the *hop-by-hop* extension header; only endpoints process other extension headers.

33.12 IPv6 Fragmentation And Reassembly

As in IPv4, IPv6 arranges for the ultimate destination to perform datagram reassembly. However, the designers chose to make changes that avoid fragmentation by routers. Recall that IPv4 requires an intermediate router to fragment any datagram that is too large for the MTU of the network over which it must travel. In IPv6, fragmentation is end-to-end; no fragmentation needs to occur in intermediate routers. The source, which is responsible for fragmentation, has two choices: it can either use the *guaranteed minimum MTU* of *1280* octets or perform *Path MTU Discovery* to identify the minimum MTU along the path to the destination. In either case, the source fragments the datagram so that each fragment is less than the expected path MTU.

The IPv6 base header does not contain fields analogous to the fields used for fragmentation in an IPv4 header. Instead, when fragmentation is needed, the source inserts a small extension header after the base header in each fragment. Figure 33.4 shows the contents of a *Fragment Extension Header*.

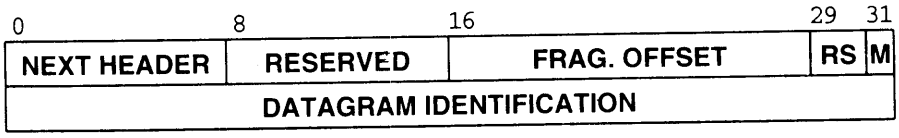


Figure 33.4 The format of a Fragment Extension Header.

IPv6 retains the basic IPv4 fragmentation functionality. Each fragment must be a multiple of *8* octets, the single bit in the *M* field marks the last fragment like the IPv4 *MORE FRAGMENTS* bit, and the *DATAGRAM IDENTIFICATION* field carries a unique ID that the receiver uses to group fragments†. Finally, field *RS* is currently reserved; the two bits are set to zero on transmission and ignored by the receiver.

33.13 The Consequence Of End-To-End Fragmentation

The motivation for using end-to-end fragmentation lies in its ability to reduce overhead in routers and permit each router to handle more datagrams per unit time. Indeed, the CPU overhead required for IPv4 fragmentation can be significant — in a conventional router, the CPU can reach 100% utilization if the router fragments many or all of the datagrams it receives. However, end-to-end fragmentation has an important consequence: it alters the fundamental IPv4 assumption that routes change dynamically.

To understand the consequence of end-to-end fragmentation, recall that IPv4 is designed to permit routes to change at any time. For example, if a network or router fails, traffic can be routed along a different path. The chief advantage of such a system is flexibility — traffic can be routed along an alternate path without disrupting service and without informing the source or destination. In IPv6, however, routes cannot be

†IPv6 expands the IPv4 *IDENTIFICATION* field to 32 bits to accommodate higher speed networks.

changed as easily because a change in a route can also change the path MTU. If the path MTU along a new route is less than the path MTU along the original route, either an intermediate router must fragment the datagram or the original source must be informed. The problem can be summarized:

> *An internet protocol that uses end-to-end fragmentation requires a sender to discover the path MTU to each destination, and to fragment any outgoing datagram that is larger than the path MTU. End-to-end fragmentation does not accommodate route changes.*

To solve the problem of route changes that affect the path MTU, IPv6 includes a new ICMP error message. When a router discovers that fragmentation is needed, it sends the message back to the source. When it receives such a message, the source performs another path MTU discovery to determine the new minimum MTU, and then fragments datagrams according to the new value.

33.14 IPv6 Source Routing

IPv6 retains the ability for a sender to specify a loose source route. Unlike IPv4, in which source routing is provided by options, IPv6 uses a separate extension header. As Figure 33.5 shows, the first four fields of the Routing Header are fixed. Field *ROUTING TYPE* specifies the type of routing information; the only type that has been defined, type *0*, corresponds to loose source routing. The *TYPE-SPECIFIC DATA* field contains a list of addresses of routers through which the datagram must pass. Field *SEG LEFT* specifies the total number of addresses that remain in the list. Finally field *HDR EXT LEN* specifies the size of the Routing Header.

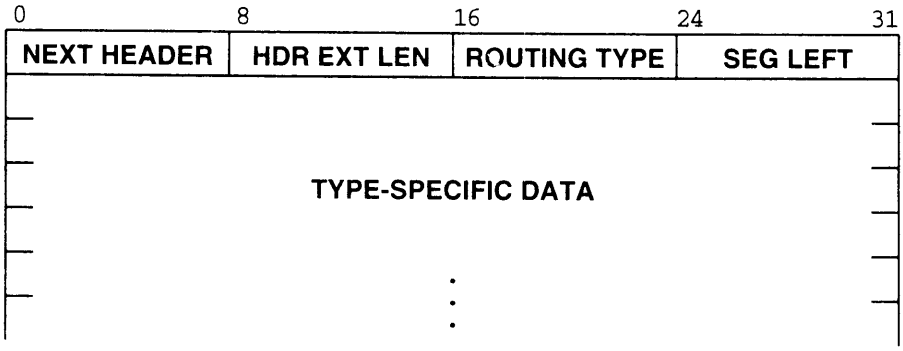


Figure 33.5 The format of an IPv6 Routing Header. Only type 0 (loose source route) is currently defined.

33.15 IPv6 Options

It may seem that IPv6 extension headers completely replace IPv4 options. However, the designers propose two additional extension headers to accommodate miscellaneous information not included in other extension headers. The additional headers are a *Hop By Hop Extension Header* and an *End To End Extension Header*. As the names imply, the two headers separate the set of options that should be examined at each hop from the set that are only interpreted at the destination.

Although each of the two option headers has a unique type code, both headers use the format illustrated in Figure 33.6.

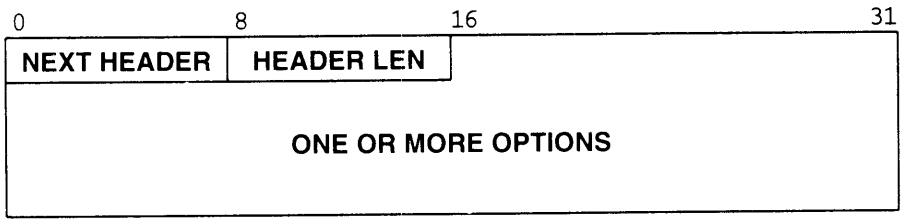


Figure 33.6 The format of an IPv6 option extension header. Both the *hop-by-hop* and *end-to-end* option headers use the same format; the *NEXT HEADER* field of the previous header distinguishes between the two types.

As usual, field *NEXT HEADER* gives the type of the header that follows. Because an option header does not have fixed size, the field labeled *HEADER LEN* specifies the total length of the header. The area labeled *ONE OR MORE OPTIONS* represents a sequence of individual options. Figure 33.7 illustrates how each individual option is encoded with a type, length, and value†; options are not aligned or padded.

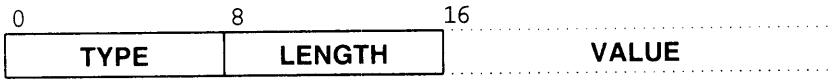


Figure 33.7 Encoding of an individual option in an IPv6 option extension header. Each option consists of a one-octet type and a one-octet length followed by zero or more octets of data for the option.

As the figure shows, IPv6 options follow the same form as IPv4 options. Each option begins with a one-octet *TYPE* field followed by a one-octet *LENGTH* field. If the option requires additional data, octets that comprise the *VALUE* follow the *LENGTH*.

†In the literature, an encoding of type, length, and value is sometimes called a *TLV encoding*.

The two high-order bits of each option *TYPE* field specify how a host or router should dispose of the datagram if it does not understand the option:

Bits In Type	Meaning
00	Skip this option
01	Discard datagram; do not send ICMP message
10	Discard datagram; send ICMP message to source
11	Discard datagram; send ICMP for non-multicast

In addition, the third bit in the *TYPE* field specifies whether the option can change in transit. Having such information is important for authentication — the contents of an option that can change in transit are treated as zeroes for purposes of authentication.

33.16 Size Of The IPv6 Address Space

In IPv6, each address occupies 16 octets, four times the size of an IPv4 address. The large address space guarantees that IPv6 can tolerate any reasonable address assignment scheme. In fact, if the designers decide to change the addressing scheme later, the address space is sufficiently large to accommodate a reassignment.

It is difficult to comprehend the size of the IPv6 address space. One way to look at it relates the magnitude to the size of the population: the address space is so large that every person on the planet can have sufficient addresses to have their own internet as large as the current Internet. A second way to think of IPv6 addressing relates it to the physical space available: the earth's surface has approximately 5.1×10^8 square kilometers, meaning that there are over 10^{24} addresses per square meter of the earth's surface. Another way to understand the size relates it to address exhaustion. For example, consider how long it would take to assign all possible addresses. A 16-octet integer can hold 2^{128} values. Thus, the address space is greater than 3.4×10^{38}. If addresses are assigned at the rate of one million addresses every microsecond, it would take over 10^{20} years to assign all possible addresses.

33.17 IPv6 Colon Hexadecimal Notation

Although it solves the problem of having insufficient capacity, the large address size poses an interesting new problem: humans who maintain internets must read, enter, and manipulate such addresses. Obviously, binary notation is untenable. However, the dotted decimal notation used for IPv4 does not make such addresses sufficiently compact either. To understand why, consider an example 128-bit number expressed in dotted decimal notation:

```
104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255
```

To help make address slightly more compact and easier to enter, the IPv6 designers propose using *colon hexadecimal notation* (abbreviated *colon hex*) in which the value of each 16-bit quantity is represented in hexadecimal separated by colons. For example, when the value shown above in dotted decimal notation has been translated to colon hex notation and printed using the same spacing, it becomes:

```
68E6:8C64:FFFF:FFFF:0:1180:96A:FFFF
```

Colon hex notation has the obvious advantage of requiring fewer digits and fewer separator characters than dotted decimal. In addition, colon hex notation includes two techniques that make it extremely useful. First, colon hex notation allows *zero compression* in which a string of repeated zeros is replaced by a pair of colons. For example, the address:

```
FF05:0:0:0:0:0:0:B3
```

can be written:

```
FF05::B3
```

To ensure that zero compression produces an unambiguous interpretation, the proposal specifies that it can be applied only once in any address. Zero compression is especially useful when used with the proposed address assignment scheme because many addresses will contain contiguous strings of zeros. Second, colon hex notation incorporates dotted decimal suffixes; we will see that such combinations are intended to be used during the transition from IPv4 to IPv6. For example, the following string is valid colon hex notation:

```
0:0:0:0:0:0:128.10.2.1
```

Note that although the numbers separated by colons each specify the value of a 16-bit quantity, numbers in the dotted decimal portion each specify the value of one octet. Of course, zero compression can be used with the number above to produce an equivalent colon hex string that looks quite similar to an IPv4 address:

```
::128.10.2.1
```

Finally, IPv6 extends CIDR-like notation by allowing an address to be followed by a slash and an integer that specifies a number of bits. For example,

```
12AB::CD30:0:0:0:0/60
```

specifies the first 60 bits of the address or 12AB00000000CD3 in hexadecimal.

33.18 Three Basic IPv6 Address Types

Like IPv4, IPv6 associates an address with a specific network connection, not with a specific computer. Thus, address assignments are similar to IPv4: an IPv6 router has two or more addresses, and an IPv6 host with one network connection needs only one address. IPv6 also retains (and extends) the IPv4 address hierarchy in which a physical network is assigned a prefix. However, to make address assignment and modification easier, IPv6 permits multiple prefixes to be assigned to a given network, and allows a computer to have multiple, simultaneous addresses assigned to a given interface.

In addition to permitting multiple, simultaneous addresses per network connection, IPv6 expands, and in some cases unifies, IPv4 special addresses. In general, a destination address on a datagram falls into one of three categories:

Unicast	The destination address specifies a single computer (host or router); the datagram should be routed to the destination along a shortest path.
Anycast	The destination is a set of computers, possibly at different locations, that all share a single address; the datagram should be routed along a shortest path and delivered to exactly one member of the group (i.e., the closest member)†.
Multicast	The destination is a set of computers, possibly at multiple locations. One copy of the datagram will be delivered to each member of the group using hardware multicast or broadcast if viable.

33.19 The Duality Of Broadcast And Multicast

IPv6 does not use the terms *broadcast* or *directed broadcast* to refer to delivery to all computers on a physical network or to a logical IP subnet. Instead, it uses the term *multicast*, and treats broadcast as a special form of multicast. The choice may seem odd to anyone who understands network hardware because more hardware technologies support broadcast than support multicast. In fact, a hardware engineer is likely to view multicasting as a restricted form of broadcasting — the hardware sends a multicast packet to all computers on the network exactly like a broadcast packet, and the interface hardware on each computer filters all multicast packets except those that software has instructed the interface hardware to accept.

In theory, the choice between multicast and limited forms of broadcast is irrelevant because one can be simulated with the other. That is, broadcasting and multicasting are duals of one another that provide the same functionality. To understand why, consider how to simulate one with the other. If broadcast is available, a packet can be delivered to a group by sending it to all machines and arranging for software on each computer to decide whether to accept or discard the incoming packet. If multicast is available, a

†Anycast addresses were formerly known as *cluster* addresses.

packet can be delivered to all machines by arranging for all machines to listen to one multicast group similar to the *all hosts* group discussed in Chapter 17.

33.20 An Engineering Choice And Simulated Broadcast

Knowing that broadcasting and multicasting are theoretical duals of one another does not help choose between them. To see why the designers of IPv6 chose multicasting as the central abstraction instead of broadcasting, consider applications instead of looking at the underlying hardware. An application either needs to communicate with a single application or with a group of applications. Direct communication is handled best via unicast; group communication is handled best by multicast or broadcast. To provide the most flexibility, group membership should not be determined by network connections, because group members can reside at arbitrary locations. Using broadcast for all group communication does not scale to handle an internet as large as the global Internet.

Not surprisingly, the designers pre-define some multicast addresses that can be used in place of an IPv4 network broadcast address. Thus, in addition to its own unicast address, each router is required to accept packets addressed to the *All Routers* multicast groups for its local environment.

33.21 Proposed IPv6 Address Space Assignment

The question of how to partition the IPv6 address space has generated much discussion. There are two central issues: how to manage address assignment and how to map an address to a route. The first issue focuses on the practical problem of devising a hierarchy of authority. Unlike the current Internet, which uses a two-level hierarchy of network prefix (assigned by the Internet authority) and host suffix (assigned by the organization), the large address space in IPv6 permits a multi-level hierarchy or multiple hierarchies. The second issue focuses on computational efficiency. Independent of the hierarchy of authority that assigns addresses, a router must examine each datagram and choose a path to the destination. To keep the cost of high-speed routers low, the processing time required to choose a path must be kept small.

As Figure 33.8 shows, the designers of IPv6 propose assigning address classes in a way similar to the scheme used for IPv4. Although the first 8 bits of an address are sufficient to identify its type, the address space is not partitioned into sections of equal size.

Binary Prefix	Type Of Address	Part Of Address Space
0000 0000	Reserved (IPv4 compatibility)	1/256
0000 0001	Unassigned	1/256
0000 001	NSAP Addresses	1/128
0000 010	IPX Addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Unassigned	1/16
001	Aggregatable Global Unicast	1/8
010	Unassigned	1/8
011	Unassigned	1/8
100	Unassigned	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link-Local Unicast Addresses	1/1024
1111 1110 11	Site-Local Unicast Addresses	1/1024
1111 1111	Multicast Addresses	1/256

Figure 33.8 The proposed division of IPv6 addresses into types, which are analogous to IPv4 classes. As in IPv4, the prefix of an address determines its address type.

As the figure shows, only 15% of the address space has been assigned at present. The IETF will use the remaining portions as demand grows. Despite the sparse assignment, addresses have been chosen to make processing more efficient. For example, the high-order octet of an address distinguishes between multicast (all 1 bits) and unicast (a mixture of 0's and 1's).

33.22 Embedded IPv4 Addresses And Transition

Although the prefix *0000 0000* is labeled *Reserved* in the figure, the designers plan to use a small fraction of addresses in that section to encode IPv4 addresses. In particular, any address that begins with 80 zero bits followed by 16 bits of all ones or 16 bits of all zeros contains an IPv4 address in the low-order 32 bits. The value of the 16-bit field indicates whether the node also has a conventional IPv6 unicast address. Figure 33.9 illustrates the two forms.

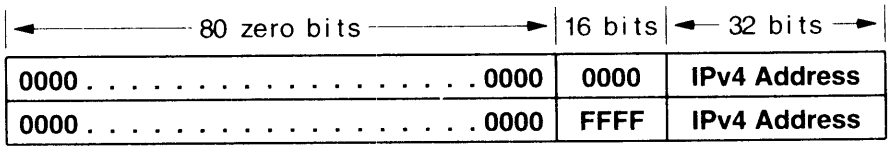


Figure 33.9 The encoding of an IPv4 address in an IPv6 address. The 16-bit field contains *0000* if the node also has a conventional IPv6 address, and *FFFF* if it does not.

The encoding will be needed during a transition from IPv4 to IPv6 for two reasons. First, a computer may choose to upgrade from IPv4 to IPv6 software before it has been assigned a valid IPv6 address. Second, a computer running IPv6 software may need to communicate with a computer that runs only IPv4 software.

Having a way to encode an IPv4 address in an IPv6 address does not solve the problem of making the two version interoperate. In addition to address encoding, translation is needed. To use a translator, an IPv6 computer generates a datagram that contains the IPv6 encoding of the IPv4 destination address. The IPv6 computer sends the datagram to a translator, which uses IPv4 to communicate with the destination. When the translator receives a reply from the destination, it translates the IPv4 datagram to IPv6 and sends it back to the IPv6 source.

It may seem that translating protocol addresses could fail because higher layer protocols verify address integrity. In particular, TCP and UDP, use a *pseudo header* in their checksum computation. The pseudo header includes both the source and destination protocol addresses, so changing such addresses could affect the computation. However, the designers planned carefully to allow TCP or UDP on an IPv4 machine to communicate with the corresponding transport protocol on an IPv6 machine. To avoid checksum mismatch, the IPv6 encoding of an IPv4 address has been chosen so that the 16-bit 1's complement checksum for both an IPv4 address and the IPv6 encoding of the address are identical. The point is:

> *In addition to choosing technical details of a new Internet Protocol, the IETF work on IPv6 has focused on finding a way to transition from the current protocol to the new protocol. In particular, the current proposal for IPv6 allows one to encode an IPv4 address inside an IPv6 address such that address translation does not change the pseudo header checksum.*

33.23 Unspecified And Loopback Addresses

As in IPv4, a few IPv6 addresses have been assigned special meaning. For example, the all 0's address:

0:0:0:0:0:0:0:0

is an *unspecified address* which cannot be assigned to any computer or used as a destination. It is only used as a source address during bootstrap by a computer that has not yet learned its address.

Like IPv4, IPv6 also has a *loopback address* that is used for testing software. The IPv6 loopback address is:

0:0:0:0:0:0:0:1

Any datagram sent to the loopback address will be delivered to the local machine; it must never be used as a destination address on an outgoing datagram.

33.24 Unicast Address Hierarchy

One of the most important changes between IPv4 and IPv6 arises from the allocation strategy used for unicast addresses and the resulting address hierarchy. Recall that the original IPv4 addressing scheme used a two-level hierarchy in which an address is divided into a globally unique prefix and a suffix. IPv6 extends the concept by adopting an address hierarchy with three conceptual levels as Figure 33.10 illustrates.

Level	Purpose
1	Globally-known public topology
2	Individual site
3	Individual network interface

Figure 33.10 The three conceptual levels of the IPv6 unicast address hierarchy. In practice, an address has additional structure.

The two lowest levels of the conceptual hierarchy are easiest to understand because they correspond to identifiable entities. The lowest level corresponds to a single attachment between a computer and a network. The middle level of the hierarchy corresponds to a set of computers and networks located at a *site*, which implies both contiguous physical connectivity and a single organization that owns and operates the equipment. We will see that the addressing scheme accommodates both large and small sites, and allows a site to have complex internal structure.

To provide flexibility, the top level of the hierarchy, which is labeled *public topology*, is not precisely defined. In general, one can think of the public topology as a "section" of the global Internet that is available for public access. Two types of public topology are envisioned. The first type corresponds to a major *Internet Service Provider (ISP)* that provides long-haul service to customers, which are known as *subscribers*. The second type, which is called an *exchange*, is a newly envisioned organization. According to the designers, exchanges will provide two functions. First, an exchange will operate like a NAP to interconnect major ISPs and pass traffic among them. Second, unlike current NAPs, exchanges will also service individual subscribers, which means that the exchange will assign the subscriber an address. The chief advantage of an address assigned by an exchange is that the address will not specify an ISP. Thus, a subscriber will be free to move from one ISP to another.

33.25 Aggregatable Global Unicast Address Structure

Authority for IPv6 address assignment flows down the hierarchy. Each top-level organization (e.g., an ISP or exchange) is assigned a unique prefix. When an organization becomes a subscriber of a top-level ISP, the organization is assigned a unique number for its site. Finally, a manager must assign a number to each network connection. To make routing efficient, successive sets of bits in the address are reserved for each assignment. Figure 33.11 illustrates the format, which is known as a *aggregatable global unicast address* format.

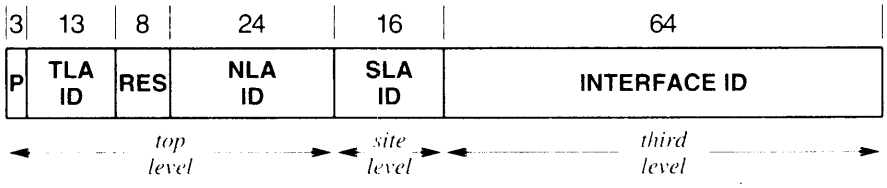


Figure 33.11 The division of an IPv6 aggregatable global unicast address into separate fields along with an indication of how those fields correspond to the three-level hierarchy.

The 3-bit field labeled *P* in the figure corresponds to the *format prefix*, which is 001 for an aggregatable global unicast address. The 8-bit *RES* field is reserved for the future and contains zeroes. Remaining fields in the address are arranged to make routing efficient. In particular, fields that correspond to the highest level of the hierarchy are grouped together to comprise the most significant bits of the address. Field *TLA ID* contains an identifier used for *Top-Level Aggregation* (i.e., a unique identifier assigned to the ISP or exchange that owns the address). The owner of the address uses field *NLA* to provide *Next-Level Aggregation* (e.g., to identify a particular subscriber).

The 16-bit field labeled *SLA ID* (*Site-Level Aggregation*) is available for a specific site to use. The designers envision it being used much like an IPv4 subnet field. Thus, a site with only a few networks can choose to treat the field as a network identifier, and a site that has many networks can use the field to partition networks into groups which can then be arranged in a hierarchy. To create a one-level hierarchy at the site, the organization must use a prefix to identify the group and a suffix to identify a particular network in the group. As with IPv4 subnetting, the division into groups improves routing efficiency because a routing table only contains routes to each of the other groups rather than to each individual network.

33.26 Interface Identifiers

As Figure 33.11 shows, the low-order 64 bits of an IPv6 aggregatable unicast address identifies a specific network interface. Unlike IPv4, however, the IPv6 suffix was chosen to be large enough to accommodate a direct encoding of the interface hardware address. Encoding a hardware address in an IP address has two consequences. First, IPv6 does not use ARP to resolve an IP address to a hardware address. Instead, IPv6 uses a *neighbor discovery protocol* available with a new version of ICMP (*ICMPv6*) to allow a node to determine which computers are its directly connected neighbors. Second, to guarantee interoperability, all computers must use the same encoding for a hardware address. Consequently, the IPv6 standards specify exactly how to encode various forms of hardware address. In the simplest case, the hardware address is placed directly in the IPv6 address; some formats use more complex transformations.

Two example encodings will help clarify the concept. For example, IEEE defines a standard 64-bit globally unique address format known as *EUI-64*. The only change needed when encoding an EUI-64 address in an IPv6 address consists of inverting bit 6 in the high-order octet of the address, which indicates whether the address is known to be globally unique.

A more complex change is required for a conventional 48-bit Ethernet address. Figure 33.12 illustrates the encoding. As the figure shows, bits from the original address are not contiguous in the encoded form. Instead, 16 bits with hexadecimal value 0xFFFE are inserted in the middle. In addition, bit 6, which indicates whether the address has global scope, is changed from *0* to *1*. Remaining bits of the address, including the group bit (labeled *g*), the ID of the company that manufactured the interface (labeled *c*), and the manufacturer's extension are copied as shown.

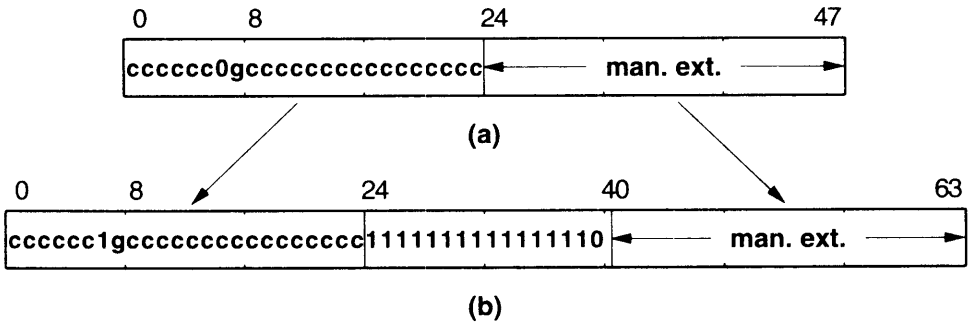


Figure 33.12 (a) The format of a 48-bit IEEE 802 address used with Ethernet, with bits labeled *c* specifying the company that manufactured the interface and bits in the *man. ext.* field specifying an extension the manufacturer chose to uniquely identify the unit, and (b) the encoding of the address in the low order 64 bits of an IPv6 unicast address.

33.27 Additional Hierarchy

Although the unicast address format in Figure 33.11 implies a strict hierarchy, many additional levels are possible. For example, bits of the *NLA ID* can be used to create a hierarchy of providers. Similarly, the 16-bit *SLA ID* can be divided to create a hierarchy within an organization. The large number of bits provides more flexibility than IPv4 subnetting. An organization can choose to divide into a two-level hierarchy of areas and assign subnets within each area. Alternatively, an organization can choose a three-level hierarchy of areas, subareas, and subnets within each subarea.

33.28 Local Addresses

In addition to the global unicast addresses described above, IPv6 includes prefixes for unicast addresses that have local scope. As Figure 33.8 shows, the standard defines two types: a *link-local address* is restricted to a single network, and a *site-local address* is restricted to a single site. Routers honor the scoping rules; they do not forward datagrams containing locally-scoped addresses outside the specified scope.

Local addresses solve two problems. Link-local addresses provide communication across a single physical network without danger of the datagram being forwarded across the internet. For example, when it performs neighbor discovery, an IPv6 node uses a link-local address. The scope rules specify that only computers on the same physical network as the sender will receive neighbor discovery messages. Similarly, computers connected to an *isolated network* (i.e., a network that does not have routers attached) can use link-local addresses to communicate.

Unlike a datagram containing link-local addresses, routers can forward datagrams containing site-local addresses throughout an entire organization. However, routers are prohibited from forwarding such datagrams to the global Internet. Thus, site-local addresses correspond to what IPv4 calls *private* or *nonroutable* addresses. An organization can assign and use site-local addresses throughout its private intranet without obtaining and assigning globally unique prefixes.

33.29 Autoconfiguration And Renumbering

IPv6 is designed to support *serverless autoconfiguration*† that allows computers to communicate without requiring a manager to specify an address. Two facilities discussed above make autoconfiguration possible and efficient: link-local addressing and embedded interface identifiers. To begin, a computer generates a link-local address by combining the link-local prefix:

1111 1110 10

with 54 zero bits and its 64-bit interface identifier.

Once it verifies that the link-local address is unique, a computer uses the address to send a *router solicitation* that requests additional information from a router. If a router is present on the network, the router responds by sending a *router advertisement* to inform the computer about prefixes that can be used for site-local or global addresses. When a router advertisement arrives, the computer makes the sender its default router. Finally, a flag in the advertisement tells the computer whether to rely on autoconfiguration or to use a conventional *managed configuration* (i.e., DHCP).

To facilitate network *renumbering*, IPv6 allows routers to limit the time a computer can retain a prefix. To do so, a router advertisement specifies two time values for each prefix: a valid lifetime and a preferred lifetime. A host must listen for additional router advertisements. When the preferred lifetime of a prefix expires, the prefix remains valid, but the host must use another prefix for all communication when possible. When the valid lifetime expires, the host must stop using the prefix, even if existing communication is in progress.

33.30 Summary

The IETF has defined a next generation of the Internet Protocol which is known as IPv6 because it has been assigned version number 6. IPv6 retains many of the basic concepts from the current protocol, IPv4, but changes most details. Like IPv4, IPv6 provides a connectionless, best-effort datagram delivery service. However, the IPv6 datagram format differs from the IPv4 format, and IPv6 provides new features such as authentication and support for flow-labeling.

IPv6 organizes each datagram as a series of headers followed by data. A datagram always begins with a 40-octet base header, which contains source and destination ad-

†Serverless autoconfiguration is also called *stateless autoconfiguration*.

dresses, a traffic class, and a flow identifier. The base header may be followed by zero or more extension headers, followed by data. Extension headers are optional — IPv6 uses them to hold much of the information IPv4 encodes in options.

An IPv6 address is 128 bits long, making the address space so large that the space cannot be exhausted in the foreseeable future. IPv6 uses address prefixes to determine the location and interpretation of remaining address fields. In addition to traditional unicast and multicast addresses, IPv6 also defines anycast addresses. A single anycast address can be assigned to a set of computers; a datagram sent to the address is delivered to exactly one computer in the set (i.e., the computer closest to the source).

IPv6 supports autoconfiguration and renumbering. Each host on an isolated network generates a unique link-local address which it uses for communication. The host also uses the link-local address to discover routers and obtain site-local and global prefix information. To facilitate renumbering, all prefixes are assigned a lifetime; a host must use a new prefix if the lifetime on an existing prefix expires.

FOR FURTHER STUDY

Many RFCs have appeared that contain information pertinent to IPv6. Deering and Hinden [RFC 2460] specifies the basic protocol. Thomson and Narten [RFC 2462] describes stateless address autoconfiguration. Narten, Nordmark, and Simpson [RFC 2461] discusses neighbor discovery. Conta and Deering [RFC 2463] specifies ICMPv6 as a companion to IPv6. Crawford [RFC 2464] describes encapsulation of IPv6 for transmission over Ethernet networks.

Many RFCs focus on IPv6 addressing. Hinden and Deering [RFC 2373] describes the basic addressing architecture including the meanings of prefixes. Hinden, O'Dell, and Deering [RFC 2374] considers the aggregatable global unicast address format. Hinden and Deering [RFC 2375] specifies multicast address assignments. Johnson and Deering [RFC 2526] describes reserved anycast addresses. Information about the 64-bit EUI format can be found in:

http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

EXERCISES

33.1 The current standard for IPv6 has no header checksum. What are the advantages and disadvantages of this approach?

33.2 How should extension headers be ordered to minimize processing time?

33.3 Although IPv6 addresses are assigned hierarchically, a router does not need to parse an address completely to select a route. Devise an algorithm and data structure for efficient routing. (Hint: consider a longest-match approach.)

33.4 Argue that 128-bit addresses are larger than needed, and that 96 bits provides sufficient capacity.

33.5 Assume your organization intends to adopt IPv6. Devise an address scheme the organization will use to assign each host an address. Did you choose a hierarchical assignment within your organization? Why or why not?

33.6 What is the chief advantage of encoding an Ethernet address in an IPv6 address? The chief disadvantage?

33.7 Consider a host that forms a link-local address by encoding its 48-bit Ethernet address with the standard link-local prefix. Is the resulting address guaranteed to be unique on the network? Why or why not?

33.8 In the previous exercise, does the standard specify that the host must use the Neighbor Discovery Protocol to verify that the address is unique? Why or why not?

33.9 If you were asked to choose sizes for the top-level, next-level, and site ID fields of an IPv6 unicast address, how large would you make each? Why?

33.10 Read about the IPv6 authentication and security headers. Why are two headers proposed?

33.11 How does the IPv6 minimum MTU of 1280 affect its flexibility?

Appendix 1

A Guide To RFCs

Introduction

Most of the written information about TCP/IP and the connected Internet, including its architecture, protocols, and history, can be found in a series of reports known as *Request For Comments* or *RFC*s. An informal, loosely coordinated set of notes, RFCs are unusually rich in information and color. Before we consider the more serious aspects of RFCs, it is fitting that we take a few minutes to pay attention to the colorful side. A good place to begin is with Cerf's poem *'Twas the Night Before Start-up* (RFC 968), a humorous parody that describes some of the problems encountered when starting a new network. Knowing not to take itself too seriously has pervaded the Internet effort. Anyone who can remember both their first Internet meeting, filled with networking jargon, and Lewis Carroll's *Jabberwocky*, filled with strangely twisted English, will know exactly why D. L. Covill put them together in *ARPAWOCKY* (RFC 527).

Other RFCs seem equally frivolous. Interspersed amid the descriptions of ideas that would turn out to dramatically change networking, we find notes like RFC 416, written in early November, 1972: *The ARC System will be Unavailable for Use During Thanksgiving Week.* It says exactly what you think it says. Or consider Crispin's tongue-in-cheek humor found in RFC 748, which describes the *TELNET Randomly-Lose Option* (a proposed option for TELNET that makes it randomly drop characters). In fact, any RFC dated April 1 should be considered a joke. If such items do not seem insignificant, think about the seventy-five RFCs listed as *never issued.* All were assigned a number and had an author, but none ever saw the light of day. The holes in the numbering scheme remain, preserved as little reminders of ideas that vaporized or work that remains incomplete.

Even after the silly, lighthearted, and useless RFCs have been removed, the remaining documents do not conform to most standards for scientific writing. Unlike scholarly scientific journals that concentrate on identifying papers of important archival interest, screening them carefully, and filing them for posterity, RFCs provide a record of ongoing conversations among the principals involved in designing, building, measuring, and using the global Internet. The reader understands at once that RFCs include the thoughts of researchers on the leading edge of technological innovation, not the studied opinions of scholars who have completely mastered a subject. The authors are not always sure of the consequences of their proposals, or even of the contents, but they clearly realize the issues are too complex to understand without community discussion. For example, RFC 1173 purports to document the "oral traditions" (which is an oxymoron because it became part of the written tradition once the RFC was published).

Despite the inconsistencies in RFCs that sometimes make them difficult for beginners to understand, the RFC mechanism has evolved and now works extremely well. Because RFCs are available electronically, information is propagated to the community quickly. Because they span a broad range of interests, practitioners as well as designers contribute. Because they record informal conversations, RFCs capture discussions and not merely final conclusions. Even the disagreements and contradictory proposals are useful in showing what the designers considered before settling on a given protocol (and readers interested in the history of a particular idea or protocol can use RFCs to follow it from its inception to its current state).

Importance Of Host And Gateway Requirements Documents

Unlike most RFCs, which concentrate on a single idea or protocol, three special RFCs cover a broad range of protocols. The special documents are entitled *Requirements for Internet Routers* and *Requirements for Internet Hosts* (parts 1 and 2).

The requirements documents, published after many years of experience with the TCP/IP protocols, are considered a major revision to the protocol standards. In essence, requirement documents each review many protocols. They point out known weaknesses or ambiguities in the RFCs that define the protocols, state conventions that have been adopted by vendors, document problems that occur in practice, and list solutions to those problems that have been accumulated through experience. The RFCs for individual protocols have *not* been updated to include changes and updates from the requirements documents. Thus, readers must be careful to always consult the requirements documents when studying a particular protocol.

RFC Numerology

RFCs cover a surprisingly large range of sizes, with the average size being 47504.5 bytes. The largest, RFC 1166 (Internet numbers), contains 566778 bytes, while the smallest consists of a 27-byte text file:

```
This RFC was never issued.
```

A few interesting coincidences have occurred. For example, the ASCII text file for RFC 41 contains exactly 41 lines of text, and the ASCII text file for RFC 854, exactly 854 lines. RFC 1996 has a number that matches the year in which it was published. However, the number for no other RFC will match the year of publication.

The quantity of RFCs published per year varies widely. Figure A1.1 illustrates how the rate has changed over time. The surge of work in the 1970s represents an initial period of building; the high rate of publication in the 1990s has resulted from commercialization.

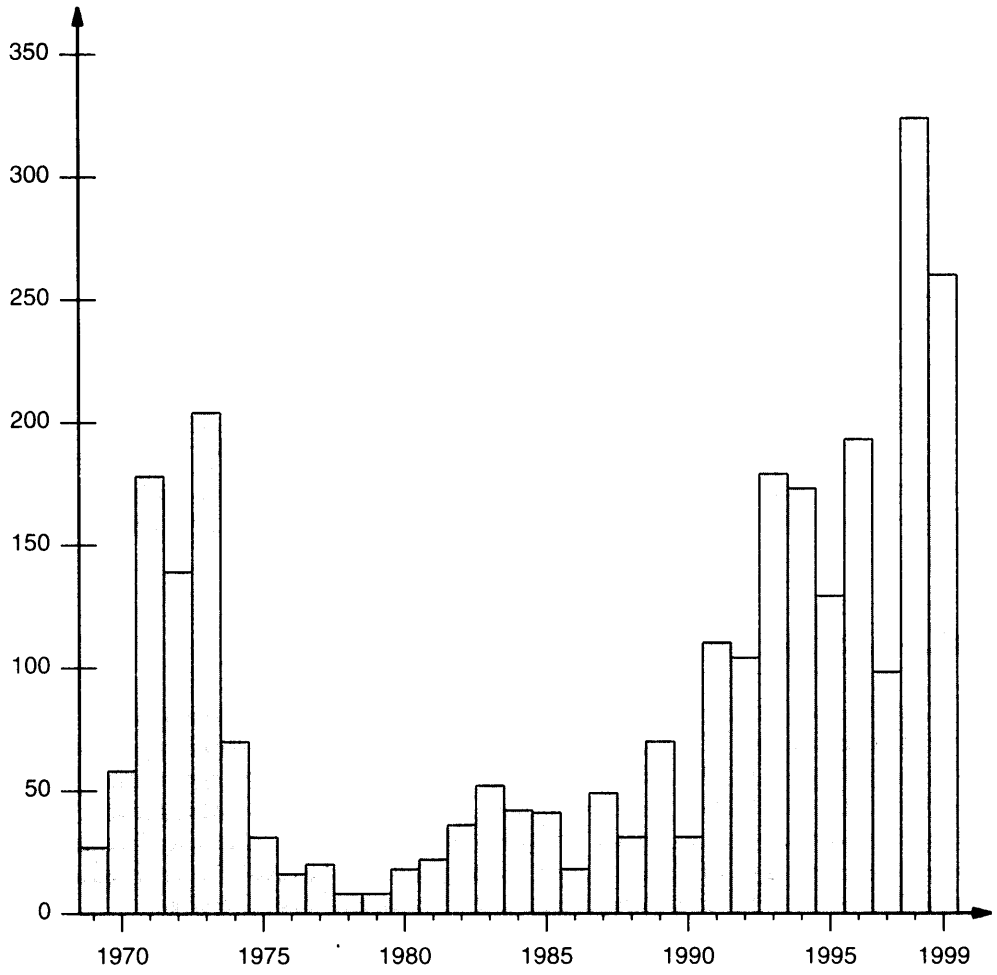


Figure A1.1 The number of RFCs published per year.

How To Obtain An RFC Over The Internet

RFCs are available electronically from many repositories around the world. Check with your local network administrator to find the site nearest you or begin with the following URL:

http://www.rfc-editor.org

Browsing Through RFCs

There are several indexes that can help one browse through RFCs. ISI publishes an index of all RFCs in chronological order. Readers often need to know which RFC contains the latest version of an official Internet protocol or which protocols are official and which are unofficial. To accommodate such needs, the IAB periodically publishes an RFC entitled *INTERNET OFFICIAL PROTOCOL STANDARDS*, which provides a list of all protocols that have been adopted as TCP/IP standards, along with the number of the most recent RFC or RFCs describing each protocol. RFC 1602, *The Internet Standards Process – Revision 2*, describes the Internet standardization process and defines the meaning of the terms *proposed standard*, *draft standard*, *Internet standard*, *required*, *recommended*, and *historic*.

Despite the available indexes, browsing through RFCs can be difficult, especially when the reader is searching for information pertinent to a given topic, which may be spread across RFCs published in many years. Browsing is particularly difficult because titles do not provide sufficient identification of the content. (How could one guess from the title *Leaving Well Enough Alone* that the RFC pertains to FTP?) Finally, having multiple RFCs with a single title (e.g., Internet Numbers) can be confusing because the reader cannot easily tell whether a document is out-of-date without checking the archive.

RFCs Arranged By Topic

The final section of this appendix provides help in finding information in RFCs because it contains a list of the first 2728 RFCs arranged by topic. Readers can find an earlier topical index in RFC 1000, which also includes an annotated chronological listing of the first 1000 RFCs. Although long, RFC 1000 is highly recommended as a source of authoritative and valuable critique – its introduction is especially fascinating. Recalling the origin of RFCs along with the origin of the ARPANET, the introduction captures the spirit of adventure and energy that still characterizes the Internet.

RFCs Organized By Major Category And Subtopic

1. Administrative

1a. Assigned Internet Numbers (official values used by protocols)

1700, 1340, 1166, 1117, 1062, 1060, 1020, 1010, 997, 990, 960, 943, 923, 900, 870, 820, 790, 776, 770, 762, 758, 755, 750, 739, 717, 604, 503, 433, 349, 322, 317, 204, 179, 175, 167

1b. Official IAB Standards and Other Lists of Protocols

2500, 2400, 2300, 2200, 2000, 1920, 1880, 1800, 1780, 1720, 1610, 1600, 1540, 1500, 1410, 1360, 1280, 1250, 1200, 1140, 1130, 1100, 1083, 1011, 991, 961, 944, 924, 901, 880, 840, 694, 661, 617, 582, 580, 552
774, 766

1c. Meeting Notes and Minutes

2316 –Report of the IAB Security Architecture Workshop
2130 –The Report of the IAB Character Set Workshop held 29 February - 1 March, 1996
1862 –Report of the IAB Workshop on Internet Information Infrastructure, October 12-14, 1994
1636 –Report of IAB Workshop on Security in the Internet Architecture - February 8-10, 1994
1588 –White Pages Meeting Report
1210 –Network and infrastructure user requirements for transatlantic research collaboration: Brussels, July 16-18, and Washington July 24-25, 1990
1152 –Workshop report: Internet research steering group workshop on very-high-speed networks
1077 –Critical issues in high bandwidth networking
1019 –Report of the Workshop on Environments for Computational Mathematics
1017 –Network requirements for scientific research: Internet task force on scientific computing
910, 807 - Multimedia mail meeting notes
898 – Gateway special interest group meeting notes
808, 805, 469 - Summary of computer mail services meeting held at BBN on 10 January 1979
585 – ARPANET users interest working group meeting
549, 396, 282, 253 - Minutes of Network Graphics Group meeting, 15-17 July 1973
371 – Demonstration at International Computer Communications Conference
327 – Data and File Transfer workshop notes
316 – ARPA Network Data Management Working Group
164, 131, 108, 101, 82, 77, 63, 37, 21 - Minutes of Network Working Group meeting, 5/16 through 5/19/71

1d. Meeting Announcements and Group Overviews

1160, 1120 - Internet Activities Board
828 – Data communications: IFIP's international "network" of experts
631 – International meeting on minicomputers and data communication: Call for papers
584 – Charter for ARPANET Users Interest Working Group
537 – Announcement of NGG meeting July 16-17
526 – Technical meeting: Digital image processing software systems
504 – Distributed resources workshop announcement
483 – Cancellation of the resource notebook framework meeting

474, 314, 246, 232, 134 - Announcement of NGWG meeting: Call for papers
471 – Workshop on multi-site executive programs
461 – Telnet Protocol meeting announcement
457 – TIPUG
456 – Memorandum: Date change of mail meeting
454 – File Transfer Protocol - meeting announcement and a new proposed document
453 – Meeting announcement to discuss a network mail system
374 – IMP System Announcement
359 – Status of the Release of the New IMP System (2600)
343, 331 - IMP System change notification
324 – RJE Protocol meeting
323 – Formation of Network Measurement Group (NMG)
320 – Workshop on Hard Copy Line Printers
309 – Data and File Transfer Workshop Announcement
299 – Information Management System
295 – Report of the Protocol Workshop, 12 October 1971
291, 188, 173 - Data Management Meeting Announcement
245, 234, 207, 140, 116, 99, 87, 85, 75, 43, 35 - Reservations for Network Group meeting
222 – Subject: System programmer's workshop
212 – NWG meeting on network usage
157 – Invitation to the Second Symposium on Problems in the Optimization of Data Communications Systems
149 – Best Laid Plans
130 – Response to RFC 111: Pressure from the chairman
111 – Pressure from the Chairman
48 – Possible protocol plateau
46 – ARPA Network protocol notes

1e. Distribution Lists

402, 363, 329, 303, 300, 211, 168, 155 - ARPA Network Mailing Lists
69 – Distribution List Change for MIT
52 – Updated distribution list

1f. Policies Documents

2717 –Registration Procedures for URL Scheme Names
2506 –Media Feature Tag Registration Procedure
2489 –Procedure for Defining New DHCP Options
2418, 1603 - IETF Working Group Guidelines and Procedures
2282, 2027 - IAB and IESG Selection, Confirmation, and Recall Process: Operation of the Nominating and Recall Committees
2278 –IANA Charset Registration Procedures
2277 –IETF Policy on Character Sets and Languages
2146, 1816, 1811 - US Government Internet Domain Names
2135 –Internet Society By-Laws
2050 –Internet Registry IP Allocation Guidelines
2042 –Registering New BGP Attribute Types
2014 –IRTF Research Group Guidelines and Procedures
1956 –Registration in the MIL Domain
1930 –Guidelines for creation, selection, and registration of an Autonomous System (AS)
1875 –UNINETT PCA Policy Statements
1371 –Choosing a Common IGP for the IP Internet

1124 –Policy issues in interconnecting networks
1087 –Ethics and the Internet
1052 –IAB recommendations for the development of Internet network management standards
1039 –DoD statement on Open Systems Interconnection protocols
980 – Protocol document order information
952, 810, 608 - DoD Internet host table specification
945 – DoD statement on the NRC report
902 – ARPA Internet Protocol policy
849 – Suggestions for improved host table distribution
678 – Standard file formats
602 – "The stockings were hung by the chimney with care"
115 – Some Network Information Center policies on handling documents
53 – Official protocol mechanism

1g. Request for Comments Administrative

2648 –A URN Namespace for IETF Documents
2629 –Writing I-Ds and RFCs using XML
2499, 2399, 2299, 2199, 2099, 1999, 1899, 1799, 1699, 1599, 1499, 1399, 1299, 999, 899, 800, 699, 598, 200, 170, 160, 100, 84 - Request for Comments Summary
2434 –Guidelines for Writing an IANA Considerations Section in RFCs
2360 –Guide for Internet Standards Writers
2223, 1543, 1111 - Instructions to RFC Authors
2119 –Key words for use in RFCs to Indicate Requirement Levels
1818 –Best Current Practices
1796 –Not All RFCs are Standards
1311 –Introduction to the STD Notes
1150 –FYI on FYI: Introduction to the FYI Notes
1000 –Request For Comments reference guide
825 – Request for comments on Requests For Comments
629 – Scenario for using the Network Journal
628 – Status of RFC numbers and a note on pre-assigned journal numbers

1h. Other

2691 –A Memorandum of Understanding for an ICANN Protocol Support Organization
2690 –A Proposal for an MOU-Based ICANN Protocol Support Organization
2436 –Collaboration between ISOC/IETF and ITU-T
2339, 1790 - An Agreement Between the Internet Society, the IETF, and Sun Microsystems, Inc
2134 –Articles of Incorporation of Internet Society
2053 –The AM (Armenia) Domain
2031 –IETF-ISOC relationship
2028 –The Organizations Involved in the IETF Standards Process
2026, 1871, 1602, 1310 - The Internet Standards Process -- Revision 3
1988 –Conditional Grant of Rights to Specific Hewlett-Packard Patents In Conjunction With the Internet Engineering Task Force's Internet-Standard Network Management Framework
1984 –IAB and IESG Statement on Cryptographic Technology and the Internet
1917 –An Appeal to the Internet Community to Return Unused IP Networks (Prefixes) to the IANA
1822 –A Grant of Rights to Use a Specific IBM patent with Photuris
1718, 1539, 1391 - The Tao of IETF - A Guide for New Attendees of the Internet Engineering Task Force

1690 –Introducing the Internet Engineering and Planning Group (IEPG)
1689 –A Status Report on Networked Information Retrieval: Tools and Groups
1640 –The Process for Organization of Internet Standards Working Group (POISED)
1601, 1358 - Charter of the Internet Architecture Board (IAB)
1527 –What Should We Plan Given the Dilemma of the Network?
1481 –IAB Recommendation for an Intermediate Strategy to Address the Issue of Scaling
1401 –Correspondence between the IAB and DISA on the use of DNS
1396 –The Process for Organization of Internet Standards Working Group (POISED)
1380 –IESG Deliberations on Routing and Addressing
1297 –NOC Internal Integrated Trouble Ticket System Functional Specification Wishlist ("NOC TT REQUIREMENTS")
1287 –Towards the Future Internet Architecture
1272 –Internet Accounting: Background
1261 –Transition of Nic Services
1174 –IAB recommended policy on distributing internet identifier assignment and IAB recommended policy change to internet "connected" status
637 – Change of network address for SU-DSL
634 – Change in network address for Haskins Lab
616 – Latest network maps
609 – Statement of upcoming move of NIC/NLS service
590 – MULTICS address change
588 – London node is now up
551 – NYU, ANL, and LBL Joining the Net
544 – Locating on-line documentation at SRI-ARC
543 – Network journal submission and delivery
518 – ARPANET accounts
511 – Enterprise phone service to NIC from ARPANET sites
510 – Request for network mailbox addresses
440 – Scheduled network software maintenance
432 – Network logical map
423, 389 - UCLA Campus Computing Network Liaison Staff for ARPANET
421 – Software Consulting Service for Network Users
419 – To: Network liaisons and station agents
416 – ARC System Will Be Unavailable for Use During Thanksgiving Week
405 – Correction to RFC 404
404 – Host Address Changes Involving Rand and ISI
403 – Desirability of a network 1108 service
386 – Letter to TIP users-2
384 – Official site idents for organizations in the ARPA Network
381 – Three aids to improved network operation
365 – Letter to All TIP Users
356 – ARPA Network Control Center
334 – Network Use on May 8
305 – Unknown Host Numbers
301 – BBN IMP (#5) and NCC Schedule March 4, 1971
289 – What we hope is an official list of host names
276 – NIC course
249 – Coordination of equipment and supplies purchase
223 – Network Information Center schedule for network users
185 – NIC distribution of manuals and handbooks

154 – Exposition Style
136 – Host accounting and administrative procedures
118 – Recommendations for facility documentation
95 – Distribution of NWG/RFC's through the NIC
16 – M.I.T

2. Requirements Documents and Major Protocol Revisions

2a. Host Requirements

1127 –Perspective on the Host Requirements RFCs
1123 –Requirements for Internet hosts - application and support
1122 –Requirements for Internet hosts - communication layers

2b. Gateway Requirements

2644 –Changing the Default for Directed Broadcasts in Routers
1812, 1009 - Requirements for IF Version 4 Routers

3. Network Interface Level (Also see Section 8)

3a. Address Binding (ARP, RARP)

2390, 1293 - Inverse Address Resolution Protocol
1931 –Dynamic RARP Extensions for Automatic Network Address Acquisition
1868 –ARP Extension - UNARP
1433 –Directed ARP
1329 –Thoughts on Address Resolution for Dual MAC FDDI Networks
1027 –Using ARP to implement transparent subnet gateways
925 – Multi-LAN address resolution
903 – Reverse Address Resolution Protocol
826 – Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware

3b. Internet Protocol over another network (encapsulation)

2728 –The Transmission of IP Over the Vertical Blanking Interval of a Television Signal
2625 –IP and ARP over Fibre Channel
2176 –IPv4 over MAPOS Version 1
2143 –Encapsulating IP with the Small Computer System Interface
2067, 1374 - IP over HIPPI
2004, 2003, 1853 - Minimal Encapsulation within IP
1390, 1188, 1103 - Transmission of IP and ARP over FDDI Networks
1241 –Scheme for an internet encapsulation protocol: Version 1
1226 –Internet protocol encapsulation of AX.25 frames
1221, 907 - Host Access Protocol (HAP) specification: Version 2
1209 –Transmission of IP datagrams over the SMDS Service
1201, 1051 - Transmitting IP traffic over ARCNET networks
1088 –Standard for the transmission of IP datagrams over NetBIOS networks
1055 –Nonstandard for transmission of IP datagrams over serial lines: SLIP
1044 –Internet Protocol on Network System's HYPERchannel: Protocol specification
1042 –Standard for the transmission of IP datagrams over IEEE 802 networks
948 – Two methods for the transmission of IP datagrams over IEEE 802.3 networks
895 – Standard for the transmission of IP datagrams over experimental Ethernet networks
894 – Standard for the transmission of IP datagrams over Ethernet networks
893 – Trailer encapsulations
877 – Standard for the transmission of IP datagrams over public data networks

3c. Nonbroadcast Multiple Access Networks (ATM, IP Switching, MPLS)

2702 –Requirements for Traffic Engineering Over MPLS
2684 –Multiprotocol Encapsulation over ATM Adaptation Layer 5
2682 –Performance Issues in VC-Merge Capable ATM LSRs
2643 –Cabletron's SecureFast VLAN Operational Model
2642 –Cabletron's VLS Protocol Specification
2641 –Cabletron's VlanHello Protocol Specification Version 4
2603 –ILMI-Based Server Discovery for NHRP
2602 –ILMI-Based Server Discovery for MARS
2601 –ILMI-Based Server Discovery for ATMARP
2583 –Guidelines for Next Hop Client (NHC) Developers
2520 –NHRP with Mobile NHCs
2443 –A Distributed MARS Service Using SCSP
2383 –ST2+ over ATM Protocol Specification - UNI 3.1 Version
2340 –Nortel's Virtual Network Switching (VNS) Overview
2337 –Intra-LIS IP multicast among routers over ATM using Sparse Mode PIM
2336 –Classical IP to NHRP Transition
2335 –A Distributed NHRP Service Using SCSP
2334 –Server Cache Synchronization Protocol (SCSP)
2333 –NHRP Protocol Applicability Statement
2332 –NBMA Next Hop Resolution Protocol (NHRP)
2331 –ATM Signalling Support for IP over ATM - UNI Signalling 4.0 Update
2297, 1987 - Ipsilon's General Switch Management Protocol Specification Version 2.0
2269 –Using the MARS Model in non-ATM NBMA Networks
2226 –IP Broadcast over ATM Networks
2225, 1577 - Classical IP and ARP over ATM
2191 –VENUS - Very Extensive Non-Unicast Service
2170 –Application REQuested IP over ATM (AREQUIPA)
2149 –Multicast Server Architectures for MARS-based ATM multicasting
2129 –Toshiba's Flow Attribute Notification Protocol (FANP) Specification
2124 –Cabletron's Light-weight Flow Admission Protocol Specification Version 1.0
2121 –Issues affecting MARS Cluster Size
2105 –Cisco Systems' Tag Switching Architecture Overview
2098 –Toshiba's Router Architecture Extensions for ATM : Overview
2022 –Support for Multicast over UNI 3.0/3.1 based ATM Networks
1954 –Transmission of Flow Labelled IPv4 on ATM Data Links Ipsilon Version 1.0
1953 –Ipsilon Flow Management Protocol Specification for IPv4 Version 1.0
1932 –IP over ATM: A Framework Document
1755 –ATM Signaling Support for IP over ATM
1754 –IP over ATM Working Group's Recommendations for the ATM Forum's Multiprotocol BOF Version 1
1735 –NBMA Address Resolution Protocol (NARP)
1626 –Default IP MTU for use over ATM AAL5
1483 –Multiprotocol Encapsulation over ATM Adaptation Layer 5

3d. Other

2469 –A Caution On The Canonical Ordering Of Link-Layer Addresses
2427, 1490, 1294 - Multiprotocol Interconnect over Frame Relay
2341 –Cisco Layer Two Forwarding (Protocol) "L2F"
2175 –MAPOS 16 - Multiple Access Protocol over SONET/SDH with 16 Bit Addressing
2174 –A MAPOS version 1 Extension - Switch-Switch Protocol

2173 –A MAPOS version 1 Extension - Node Switch Protocol
2172 –MAPOS Version 1 Assigned Numbers
2171 –MAPOS - Multiple Access Protocol over SONET/SDH Version 1
1326 –Mutual Encapsulation Considered Dangerous

4. Internet Level

4a. Internet Protocol (IP)

2113 –IP Router Alert Option
1624, 1141 - Computation of the Internet Checksum via Incremental Update
1191, 1063 - Path MTU discovery
1071 –Computing the Internet checksum
1025 –TCP and IP bake off
815 – IP datagram reassembly algorithms
791, 760 - Internet Protocol
781 – Specification of the Internet Protocol (IP) timestamp option

4b. Internet Control Message Protocol (ICMP)

2521 –ICMP Security Failures Messages
1788 –ICMP Domain Name Messages
1256 –ICMP Router Discovery Messages
1018 –Some comments on SQuID
1016 –Something a host could do with source quench: The Source Quench Introduced Delay (SQuID)
792, 777 - Internet Control Message Protocol

4c. Multicast (IGMP)

2588 –IP Multicast and Firewalls
2502 –Limitations of Internet Protocol Suite for Distributed Simulation the Large Multicast Environment
2365 –Administratively Scoped IP Multicast
2357 –IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols
2236 –Internet Group Management Protocol, Version 2
1768 –Host Group Extensions for CLNP Multicasting
1469 –IP Multicast over Token-Ring Local Area Networks
1458 –Requirements for Multicast Protocols
1301 –Multicast Transport Protocol
1112, 1054, 988, 966 - Host extensions for IP multicasting

4d. Routing and Gateway Algorithms (BGP, GGP, RIP, OSPF)

2715 –Interoperability Rules for Multicast Routing Protocols
2676 –QoS Routing Mechanisms and OSPF Extensions
2650 –Using RPSL in Practice
2622, 2280 - Routing Policy Specification Language (RPSL
2519 –A Framework for Inter-Domain Route Aggregation
2453, 1723, 1388 - RIP Version 2
2439 –BGP Route Flap Damping
2385 –Protection of BGP Sessions via the TCP MD5 Signature Option
2370 –The OSPF Opaque LSA Option
2362, 2117 - Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification
2338 –Virtual Router Redundancy Protocol
2329 –OSPF Standardization Report
2328, 2178, 1583, 1247, 1131 - OSPF Version 2

2283 –Multiprotocol Extensions for BGP-4
2281 –Cisco Hot Standby Router Protocol (HSRP)
2270 –Using a Dedicated AS for Sites Homed to a Single Provider
2260 –Scalable Support for Multi-homed Multi-provider Connectivity
2201, 2189 - Core Based Trees (CBT) Multicast Routing Architecture
2154 –OSPF with Digital Signatures
2103 –Mobility Support for Nimrod : Challenges and Solution Approaches
2102 –Multicast Support for Nimrod : Requirements and Solution Approaches
2092 –Protocol Analysis for Triggered RIP
2091 –Triggered Extensions to RIP to Support Demand Circuits
2082 –RIP-2 MD5 Authentication
2009 –GPS-Based Addressing and Routing
1998 –An Application of the BGP Community Attribute in Multi-home Routing
1997 –BGP Communities Attribute
1992 –The Nimrod Routing Architecture
1966 –BGP Route Reflection An alternative to full mesh IBGP
1965 –Autonomous System Confederations for BGP
1923 –RIPv1 Applicability Statement for Historic Status
1863 –A BGP/IDRP Route Server alternative to a full mesh routing
1817 –CIDR and Classful Routing
1793 –Extending OSPF to Support Demand Circuits
1787 –Routing in a Multi-provider Internet
1786 –Representation of IP Routing Policies in a Routing Registry (ripe-81++)
1774 –BGP-4 Protocol Analysis
1773, 1656 - Experience with the BGP-4 protocol
1772, 1655, 1268, 1164 - Application of the Border Gateway Protocol in the Internet
1771, 1654, 1267, 1163 - A Border Gateway Protocol 4 (BGP-4)
1765 –OSPF Database Overflow
1745 –BGP4/IDRP for IP---OSPF Interaction
1722 –RIP Version 2 Protocol Applicability Statement
1721, 1387 - RIP Version 2 Protocol Analysis
1702, 1701 - Generic Routing Encapsulation over IPv4 networks
1587 –The OSPF NSSA Option
1586 –Guidelines for Running OSPF Over Frame Relay Networks
1585 –MOSPF: Analysis and Experience
1584 –Multicast Extensions to OSPF
1582 –Extensions to RIP to Support Demand Circuits
1581 –Protocol Analysis for Extensions to RIP to Support Demand Circuits
1520 –Exchanging Routing Information Across Provider Boundaries in the CIDR Environment
1519, 1338 - Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
1517 –Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR)
1504 –Appletalk Update-Based Routing Protocol: Enhanced Appletalk Routing
1482 –Aggregation Support in the NSFNET Policy-Based Routing Database
1479 –Inter-Domain Policy Routing Protocol Specification: Version 1
1478 –An Architecture for Inter-Domain Policy Routing
1477 –IDPR as a Proposed Standard
1465 –Routing Coordination for X.400 MHS Services Within a Multi Protocol / Multi Network Environment Table Format V3 for Static Routing
1403, 1364 - BGP OSPF Interaction
1397 –Default Route Advertisement In BGP2 and BGP3 Version of The Border Gateway Protocol

1383 –An Experiment in DNS Based IP Routing
1370 –Applicability Statement for OSPF
1322 –A Unified Approach to Inter-Domain Routing
1266 –Experience with the BGP Protocol
1265 –BGP Protocol Analysis
1264 –Internet Engineering Task Force Internet Routing Protocol Standardization Criteria
1254 –Gateway Congestion Control Survey
1246 –Experience with the OSPF Protocol
1245 –OSPF Protocol Analysis
1222 –Advancing the NSFNET routing architecture
1195 –Use of OSI IS-IS for routing in TCP/IP and dual environments
1142 –OSI IS-IS Intra-domain Routing Protocol
1136 –Administrative Domains and Routing Domains: A model for routing in the Internet
1133 –Routing between the NSFNET and the DDN
1126 –Goals and functional requirements for inter-autonomous system routing
1125 –Policy requirements for inter Administrative Domain routing
1105 –Border Gateway Protocol (BGP)
1104 –Models of policy based routing
1102 –Policy routing in Internet protocols
1093 –NSFNET routing architecture
1092 –EGP and policy based routing in the new NSFNET backbone
1075 –Distance Vector Multicast Routing Protocol
1074 –NSFNET backbone SPF based Interior Gateway Protocol
1058 –Routing Information Protocol
1046 –Queuing algorithm to provide type-of-service for IP links
985 – Requirements for Internet gateways - draft
975 – Autonomous confederations
970 – On packet switches with infinite storage
911 – EGP Gateway under Berkeley UNIX 4.2
904, 890, 888, 827 - Exterior Gateway Protocol formal specification
875 – Gateways, architectures, and heffalumps
823 – DARPA Internet gateway

4e. IP: The Next Generation (IPng, IPv6)

2711 –IPv6 Router Alert Option
2710 –Multicast Listener Discovery (MLD) for IPv6
2675, 2147 - IPv6 Jumbograms
2590 –Transmission of IPv6 Packets over Frame Relay
2553, 2133 - Basic Socket Interface Extensions for IPv6
2546 –6Bone Routing Practice
2545 –Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
2529 –Transmission of IPv6 over IPv4 Domains without Explicit Tunnels
2526 –Reserved IPv6 Subnet Anycast Addresses
2497 –Transmission of IPv6 Packets over ARCnet Networks
2492 –IPv6 over ATM Networks
2491 –IPv6 over Non-Broadcast Multiple Access (NBMA) networks
2473 –Generic Packet Tunneling in IPv6 Specification
2472, 2023 - IP Version 6 over PPP
2471, 1897 - IPv6 Testing Address Allocation
2470 –Transmission of IPv6 Packets over Token Ring Networks
2467, 2019 - Transmission of IPv6 Packets over FDDI Networks
2466, 2465 - Management Information Base for IP Version 6: ICMPv6 Group

2464, 1972 - Transmission of IPv6 Packets over Ethernet Networks
2463, 1885 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
2462, 1971 - IPv6 Stateless Address Autoconfiguration
2461, 1970 - Neighbor Discovery for IP Version 6 (IPv6)
2460, 1883 - Internet Protocol, Version 6 (IPv6) Specification
2454 –IP Version 6 Management Information Base for the User Datagram Protocol
2452 –IP Version 6 Management Information Base for the Transmission Control Protocol
2450 –Proposed TLA and NLA Assignment Rule
2375 –IPv6 Multicast Address Assignments
2374, 2073 - An IPv6 Aggregatable Global Unicast Address Format
2373, 1884 - IP Version 6 Addressing Architecture
2292 –Advanced Sockets API for IPv6
2185 –Routing Aspects of IPv6 Transition
2081 –RIPng Protocol Applicability Statement
2080 –RIPng for IPv6
1981 –Path MTU Discovery for IP version 6
1955 –New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG
1933 –Transition Mechanisms for IPv6 Hosts and Routers
1888 –OSI NSAPs and IPv6
1887 –An Architecture for IPv6 Unicast Address Allocation
1886 –DNS Extensions to support IP version 6
1881 –IPv6 Address Allocation Management
1809 –Using the Flow Label Field in IPv6
1753 –IPng Technical Requirements Of the Nimrod Routing and Addressing Architecture
1752 –The Recommendation for the IP Next Generation Protocol
1726 –Technical Criteria for Choosing IP The Next Generation (IPng)
1719 –A Direction for IPng
1710 –Simple Internet Protocol Plus White Paper
1707 –CATNIP: Common Architecture for the Internet
1705 –Six Virtual Inches to the Left: The Problem with IPng
1688 –IPng Mobility Considerations
1687 –A Large Corporate User's View of IPng
1686 –IPng Requirements: A Cable Television Industry Viewpoint
1683 –Multiprotocol Interoperability In IPng
1682 –IPng BSD Host Implementation Analysis
1680 –IPng Support for ATM Services
1679 –HPN Working Group Input to the IPng Requirements Solicitation
1678 –IPng Requirements of Large Corporate Networks
1677 –Tactical Radio Frequency Communication Requirements for IPng
1676 –INFN Requirements for an IPng
1675 –Security Concerns for IPng
1674 –A Cellular Industry View of IPng
1673 –Electric Power Research Institute Comments on IPng
1672 –Accounting Requirements for IPng
1671 –IPng White Paper on Transition and Other Considerations
1670 –Input to IPng Engineering Considerations
1669 –Market Viability as a IPng Criteria
1668 –Unified Routing Requirements for IPng
1667 –Modeling and Simulation Requirements for IPng
1622 –Pip Header Processing

1621 –Pip Near-term Architecture
1550 –IP: Next Generation (IPng) White Paper Solicitation
1526 –Assignment of System Identifiers for TUBA/CLNP Hosts
1475 –TP/IX: The Next Internet
1454 –Comparison of Proposals for Next Version of IP
1385 –EIP: The Extended Internet Protocol
1375 –Suggestion for New Classes of IP Addresses
1365 –An IP Address Extension Proposal
1347 –TCP and UDP with Bigger Addresses (TUBA), A Simple Proposal for Internet Addressing and Routing
1335 –A Two-Tier Address Structure for the Internet: A Solution to the Problem of Address Space Exhaustion

4f. IP Address Allocation and Network Numbering

2391 –Load Sharing using IP Network Address Translation (LSNAT)
2101 –IPv4 Address Behaviour Today
2072 –Router Renumbering Guide
2071 –Network Renumbering Overview: Why would I want it and what is it anyway?
2036 –Observations on the use of Components of the Class A Address Space within the Internet
2008 –Implications of Various Address Allocation Policies for Internet Routing
1918, 1597 - Address Allocation for Private Internets
1916 –Enterprise Renumbering: Experience and Information Solicitation
1900 –Renumbering Needs Work
1879, 1797 - Class A Subnet Experiment Results and Recommendations
1878, 1860 - Variable Length Subnet Table For IPv4
1814 –Unique Addresses are Good
1744 –Observations on the Management of the Internet Address Space
1715 –The H Ratio for Address Assignment Efficiency
1681 –On Many Addresses per Host
1627 –Network 10 Considered Harmful (Some Practices Shouldn't be Codified)
1466, 1366 - Guidelines for Management of IP Address Space
1219 –On the assignment of subnet numbers
950 – Internet Standard Subnetting Procedure
940, 936, 932, 917 - Toward an Internet standard scheme for subnetting

4g. Network Isolation (VPN, Firewall, NAT)

2694 –DNS extensions to Network Address Translators (DNS_ALG)
2685 –Virtual Private Networks Identifier
2663 –IP Network Address Translator (NAT) Terminology and Considerations
2647 –Benchmarking Terminology for Firewall Performance
2637 –Point-to-Point Tunneling Protocol
2547 –BGP/MPLS VPNs
1961 –GSS-API Authentication Method for SOCKS Version 5
1929, 1928 - Username/Password Authentication for SOCKS V5
1858 –Security Considerations for IP Fragment Filtering
1631 –The IP Network Address Translator (NAT)

4h. Other

2698 –A Two Rate Three Color Marker
2697 –A Single Rate Three Color Marker
2638 –A Two-bit Differentiated Services Architecture for the Internet
2598 –An Expedited Forwarding PHB
2597 –Assured Forwarding PHB Group

2508 –Compressing IP/UDP/RTP Headers for Low-Speed Serial Links
2507 –IP Header Compression
2481 –A Proposal to add Explicit Congestion Notification (ECN) to IP
2475 –An Architecture for Differentiated Service
2474, 1349 - Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
2395 –IP Payload Compression Using LZS
2394 –IP Payload Compression Using DEFLATE
2393 –IP Payload Compression Protocol (IPComp)
2075 –IP Echo Host Service
1946 –Native ATM Support for ST2+
1940 –Source Demand Routing: Packet Format and Forwarding Specification (Version 1)
1937 –"Local/Remote" Forwarding Decision in Switched Data Link Subnetworks
1936 –Implementing the Internet Checksum in Hardware
1919 –Classical versus Transparent IP Proxies
1819, 1190 - Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+
1770 –IPv4 Option for Sender Directed Multi-Destination Delivery
1716 –Towards Requirements for IP Routers
1620 –Internet Architecture Extensions for Shared Media
1560 –The MultiProtocol Internet
1518 –An Architecture for IP Address Allocation with CIDR
1476 –RAP: Internet Route Access Protocol
1467, 1367 - Status of CIDR Deployment in the Internet
1393 –Traceroute Using an IP Option
1363 –A Proposed Flow Specification
986 – Guidelines for the use of Internet-IP addresses in the ISO Connectionless-Mode Network Protocol
981 – Experimental multiple-path routing algorithm
963 – Some problems with the specification of the Military Standard Internet Protocol
947 – Multi-network broadcasting within the Internet
922, 919 - Broadcasting Internet datagrams in the presence of subnets
871 – Perspective on the ARPANET reference model
831 – Backup access to the European side of SATNET
817 – Modularity and efficiency in protocol implementation
816 – Fault isolation and recovery
814 – Name, addresses, ports, and routes
796 – Address mappings
795 – Service mappings
730 – Extensible field addressing

5. Host Level

5a. User Datagram Protocol (UDP)

768 – User Datagram Protocol

5b. Transmission Control Protocol (TCP)

2582 –The NewReno Modification to TCP's Fast Recovery Algorithm
2581, 2001 - TCP Congestion Control
2525 –Known TCP Implementation Problems
2488 –Enhancing TCP Over Satellite Channels using Standard Mechanisms
2416 –When TCP Starts Up With Four Packets Into Only Three Buffers

2415 –Simulation Studies of Increased Initial TCP Window Size
2414 –Increasing TCP's Initial Window
2398 –Some Testing Tools for TCP Implementors
2140 –TCP Control Block Interdependence
2018 –TCP Selective Acknowledgement Options
1693 –An Extension to TCP : Partial Order Service
1644 –T/TCP -- TCP Extensions for Transactions Functional Specification
1379 –Extending TCP for Transactions -- Concepts
1337 –TIME-WAIT Assassination Hazards in TCP
1323, 1185 - TCP Extensions for High Performance
1263 –TCP Extensions Considered Harmful
1146, 1145 - TCP alternate checksum options
1144 –Compressing TCP/IP headers for low-speed serial links
1110 –Problem with the TCP big window option
1106 –TCP big window and NAK options
1078 –TCP port service Multiplexer (TCPMUX)
1072 –TCP extensions for long-delay paths
964 – Some problems with the specification of the Military Standard Transmission Control Protocol
962 – TCP-4 prime
896 – Congestion control in IP/TCP internetworks
889 – Internet delay experiments
879 – TCP maximum segment size and related topics
872 – TCP-on-a-LAN
813 – Window and Acknowledgement Strategy in TCP
794 – Pre-emption
793, 761, 675 - Transmission Control Protocol
721 – Out-of-Band Control Signals in a Host-to-Host Protocol
700 – Protocol experiment

5c. Point-to-Point Protocols (PPP)

2701 –Nortel Networks Multi-link Multi-node PPP Bundle Discovery Protocol
2687 –PPP in a Real-time Oriented HDLC-like Framing
2686 –The Multi-Class Extension to Multi-Link PPP
2615, 1619 - PPP over SONET/SDH
2516 –Method for Transmitting PPP Over Ethernet (PPPoE)
2509 –IP Header Compression over PPP
2484 –PPP LCP Internationalization Configuration Option
2433 –Microsoft PPP CHAP Extensions
2420 –The PPP Triple-DES Encryption Protocol (3DESE)
2419, 1969 - The PPP DES Encryption Protocol, Version 2 (DESE-bis)
2364 –PPP Over AAL5
2363 –PPP Over FUNI
2284 –PPP Extensible Authentication Protocol (EAP)
2153 –PPP Vendor Extensions
2125 –The PPP Bandwidth Allocation Protocol (BAP) / The PPP Bandwidth Allocation Control Protocol (BACP)
2118 –Microsoft Point-To-Point Compression (MPPC) Protocol
2097 –The PPP NetBIOS Frames Control Protocol (NBFCP)
2043 –The PPP SNA Control Protocol (SNACP)
1994, 1334 - PPP Challenge Handshake Authentication Protocol (CHAP)
1993 –PPP Gandalf FZA Compression Protocol
1990, 1717 - The PPP Multilink Protocol (MP)
1989, 1333 - PPP Link Quality Monitoring

1979 –PPP Deflate Protocol
1978 –PPP Predictor Compression Protocol
1977 –PPP BSD Compression Protocol
1976 –PPP for Data Compression in Data Circuit-Terminating Equipment (DCE)
1975 –PPP Magnalink Variable Resource Compression
1974 –PPP Stac LZS Compression Protocol
1973 –PPP in Frame Relay
1968 –The PPP Encryption Control Protocol (ECP)
1967 –PPP LZS-DCP Compression Protocol (LZS-DCP)
1963 –PPP Serial Data Transport Protocol (SDTP)
1962 –The PPP Compression Control Protocol (CCP)
1934 –Ascend's Multilink Protocol Plus (MP+)
1915 –Variance for The PPP Connection Control Protocol and The PPP Encryption Control Protocol
1877 –PPP Internet Protocol Control Protocol Extensions for Name Server Addresses
1841 –PPP Network Control Protocol for LAN Extension
1764 –The PPP XNS IDP Control Protocol (XNSCP)
1763 –The PPP Banyan Vines Control Protocol (BVCP)
1762, 1376 - The PPP DECnet Phase IV Control Protocol (DNCP)
1663 –PPP Reliable Transmission
1662, 1549 - PPP in HDLC-like Framing
1661, 1548 - The Point-to-Point Protocol (PPP)
1638, 1220 - PPP Bridging Control Protocol (BCP)
1618 –PPP over ISDN
1598 –PPP in X.25
1570 –PPP LCP Extensions
1552 –The PPP Internetworking Packet Exchange Control Protocol (IPXCP)
1547 –Requirements for an Internet Standard Point-to-Point Protocol
1378 –The PPP AppleTalk Control Protocol (ATCP)
1377 –The PPP OSI Network Layer Control Protocol (OSINLCP)
1332, 1172 - The PPP Internet Protocol Control Protocol (IPCP)
1331, 1171, 1134 - The Point-to-Point Protocol (PPP) for the Transmission of Multi-protocol Datagrams over Point-to-Point Links

5e. Transaction Protocols and Distributed Operating Systems

2372 –Transaction Internet Protocol - Requirements and Supplemental Information
2371 –Transaction Internet Protocol Version 3.0
955 – Towards a transport service for transaction processing applications
938 – Internet Reliable Transaction Protocol functional and interface specification
722 – Thoughts on Interactions in Distributed Services
713 – MSDTP-Message Services Data Transmission Protocol
712 – Distributed Capability Computing System (DCCS)
708 – Elements of a Distributed Programming System
707 – High-level framework for network-based resource sharing
684 – Commentary on procedure calling as a network protocol
677 – Maintenance of duplicate databases
674 – Procedure call documents: Version 2
672 – Multi-site data collection facility
671 – Note on Reconnection Protocol
645 – Network Standard Data Specification syntax
615 – Proposed Network Standard Data Pathname syntax
610 – Further datalanguage design concepts
592 – Some thoughts on system design to facilitate resource sharing

578 – Using MIT-Mathlab MACSYMA from MIT-DMS Muddle
515 – Specifications for datalanguage: Version 0/9
500 – Integration of data management systems on a computer network
441 – Inter-Entity Communication - an experiment
437 – Data Reconfiguration Service at UCSB
203 – Achieving reliable communication
76 – Connection by name: User oriented protocol
62 – Systems for Interprocess Communication in a Resource Sharing Computer Network
61 – Note on Interprocess Communication in a Resource Sharing Computer Network
51 – Proposal for a Network Interchange Language
31 – Binary Message Forms in Computer

5f. Protocols for Local Area Networks (NETBIOS)

1002 –Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications
1001 –Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods

5g. IP Mobility and Roaming

2607 –Proxy Chaining and Policy Implementation in Roaming
2548, 2138, 2058 - Microsoft Vendor-specific RADIUS Attributes
2501 –Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations
2486 –The Network Access Identifier
2477 –Criteria for Evaluating Roaming Protocols
2356 –Sun's SKIP Firewall Traversal for Mobile IP
2344 –Reverse Tunneling for Mobile IP
2290 –Mobile-IPv4 Configuration Option for PPP IPCP
2194 –Review of Roaming Implementations
2139, 2059 - RADIUS Accounting
2041 –Mobile Network Tracing
2005 –Applicability Statement for IP Mobility Support
2002 –IP Mobility Support

5h. Other

1546 –Host Anycasting Service
1312, 1159 - Message Send Protocol 2
1151, 908 - Version 2 of the Reliable Data Protocol (RDP)
1045 –VMTP: Versatile Message Transaction Protocol: Protocol specification
998, 969 - NETBLT: A bulk data transfer protocol
979 – PSN End-to-End functional specification
869 – Host Monitoring Protocol
643 – Network Debugging Protocol
162 – NETBUGGER3

6. Application Level

6a. Telnet Protocol (TELNET)

2355, 1647 - TN3270 Enhancements
1921 –TNVIP Protocol
1646 –TN3270 Extensions for LUname and Printer Selection
1576 –TN3270 Current Practices
1205 –5250 Telnet interface

1184 –Telnet Linemode Option
854, 764 - Telnet Protocol Specification
818 – Remote User Telnet service
782 – Virtual Terminal management model
728 – Minor pitfall in the Telnet Protocol
703, 702, 701, 679, 669 - July, 1975, survey of New-Protocol Telnet Servers
688 – Tentative schedule for the new Telnet implementation for the TIP
681 – Network UNIX
600 – Interfacing an Illinois plasma terminal to the ARPANET
596 – Second thoughts on Telnet Go-Ahead
595 – Second thoughts in defense of the Telnet Go-Ahead
593 – Telnet and FTP implementation schedule change
576 – Proposal for modifying linking
570 – Experimental input mapping between NVT ASCII and UCSB On Line System
562 – Modifications to the Telnet specification
559 – Comments on The New Telnet Protocol and its Implementation
529 – Note on protocol synch sequences
513 – Comments on the new Telnet specifications
495 – Telnet Protocol specifications
466 – Telnet logger/server for host LL-67
452 – TELENET Command at Host LL
435 – Telnet issues
426 – Reconnection Protocol
393 – Comments on Telnet Protocol Changes
377 – Using TSO via ARPA Network Virtual Terminal
357 – Echoing strategy for satellite links
355, 346 - Response to NWG/RFC 346
340 – Proposed Telnet Changes
339 – MLTNET: A "Multi Telnet" Subsystem for Tenex
328 – Suggested Telnet Protocol Changes
318 – Telnet Protocols
216 – Telnet access to UCSB's On-Line System
215 – NCP, ICP, and Telnet: The Terminal IMP implementation
206 – User Telnet - description of an initial implementation
205 – NETCRT - a character display protocol
190 – DEC PDP-10-IMLAC communications system
158 – Telnet Protocol: A Proposed Document
139 – Discussion of Telnet Protocol
137 – Telnet Protocol - a proposed document
135, 110 - Response to NWG/RFC 110
103 – Implementation of Interrupt Keys
97 – First Cut at a Proposed Telnet Protocol
91 – Proposed User-User Protocol

6b. Telnet Options

2217 –Telnet Com Port Control Option
2066 –TELNET CHARSET Option
1572, 1408 - Telnet Environment Option
1571 –Telnet Environment Option Interoperability Issues
1416, 1409 - Telnet Authentication Option
1412 –Telnet Authentication: SPX
1411 –Telnet Authentication: Kerberos Version 4
1372, 1080 - Telnet Remote Flow Control Option

1143 –The Q Method of Implementing TELNET Option Negotiation
1116 –Telnet Linemode option
1096 –Telnet X display location option
1091 –Telnet terminal-type option
1079 –Telnet terminal speed option
1073 –Telnet window size option
1053 –Telnet X.3 PAD option
1043 –Telnet Data Entry Terminal option: DODIIS implementation
1041 –Telnet 3270 regime option
946 – Telnet terminal location number option
933 – Output marking Telnet option
930 – Telnet terminal type option
927 – TACACS user identification Telnet option
885 – Telnet end of record option
884 – Telnet terminal type option
861 – Telnet Extended Options: List Option
860 – Telnet Timing Mark Option
859 – Telnet Status Option
858 – Telnet Suppress Go Ahead Option
857 – Telnet Echo Option
856 – Telnet Binary Transmission
855 – Telnet Option Specifications
779 – Telnet send-location option
749 – Telnet SUPDUP-Output option
747 – Recent extensions to the SUPDUP Protocol
746 – SUPDUP graphics extension
736 – Telnet SUPDUP option
735 – Revised Telnet byte macro option
732 – Telnet Data Entry Terminal option
731 – Telnet Data Entry Terminal option
729 – Telnet byte macro option
727 – Telnet logout option
726 – Remote Controlled Transmission and Echoing Telnet option
719 – Discussion on RCTE
718 – Comments on RCTE from the Tenex Implementation Experience
698 – Telnet extended ASCII option
659 – Announcing additional Telnet options
658 – Telnet output linefeed disposition
657 – Telnet output vertical tab disposition option
656 – Telnet output vertical tabstops option
655 – Telnet output formfeed disposition option
654 – Telnet output horizontal tab disposition option
653 – Telnet output horizontal tabstops option
652 – Telnet output carriage-return disposition option
651 – Revised Telnet status option
587 – Announcing new Telnet options
581 – Corrections to RFC 560: Remote Controlled Transmission and Echoing Telnet Option
563 – Comments on the RCTE Telnet option
560 – Remote Controlled Transmission and Echoing Telnet option

6c. File Transfer and Access Protocols (FTP, TFTP, SFTP, NFS)

2640 –Internationalization of the File Transfer Protocol
2624 –NFS Version 4 Design Considerations

2623 –NFS Version 2 and Version 3 Security Issues and the NFS Protocol's Use of RPCSEC_GSS and Kerberos V5
2577 –FTP Security Considerations
2428 –FTP Extensions for IPv6 and NATs
2389 –Feature negotiation mechanism for the File Transfer Protocol
2349, 1784 - TFTP Timeout Interval and Transfer Size Options
2348, 1783 - TFTP Blocksize Option
2347, 1782 - TFTP Option Extension
2228 –FTP Security Extensions
2224 –NFS URL Scheme
2204 –ODETTE File Transfer Protocol
2090 –TFTP Multicast Option
2055 –WebNFS Server Specification
2054 –WebNFS Client Specification
1986 –Experiments with a Simple File Transfer Protocol for Radio Links using Enhanced Trivial File Transfer Protocol (ETFTP)
1813 –NFS Version 3 Protocol Specification
1785 –TFTP Option Negotiation Analysis
1639, 1545 - FTP Operation Over Big Address Records (FOOBAR)
1635 –How to Use Anonymous FTP
1579 –Firewall-Friendly FTP
1440 –SIFT/UFT: Sender-Initiated/Unsolicited File Transfer
1415 –FTP-FTAM Gateway Specification
1350, 783 - The TFTP Protocol (Revision 2)
1282, 1258 - BSD Rlogin
1235 –Coherent File Distribution Protocol
1094 –NFS: Network File System Protocol specification
1068 –Background File Transfer Program (BFTP)
1037 –NFILE - a file access protocol
959, 765, 542, 354, 265, 172, 114 - File Transfer Protocol
949 – FTP unique-named store command
913 – Simple File Transfer Protocol
906 – Bootstrap loading using TFTP
775 – Directory oriented FTP commands
743 – FTP extension: XRSQ/XRCP
737 – FTP extension: XSEN
697 – CWD command of FTP
691 – One more try on the FTP
686 – Leaving well enough alone
683 – FTPSRV - Tenex extension for paged files
662 – Performance improvement in ARPANET file transfers from Multics
640 – Revised FTP reply codes
630 – FTP error code usage for more reliable mail service
624 – Comments on the File Transfer Protocol
614 – Response to RFC 607: "Comments on the File Transfer Protocol"
607 – Comments on the File Transfer Protocol
571 – Tenex FTP problem
535 – Comments on File Access Protocol
532 – UCSD-CC Server-FTP facility
520 – Memo to FTP group: Proposal for File Access Protocol
506 – FTP command naming problem
505 – Two solutions to a file transfer access problem
501 – Un-muddling "free file transfer"

487 – Free file transfer
486 – Data transfer revisited
480 – Host-dependent FTP parameters
479 – Use of FTP by the NIC Journal
478 – FTP server-server interaction - II
468 – FTP data compression
463 – FTP comments and response to RFC 430
448 – Print files in FTP
438 – FTP server-server interaction
430 – Comments on File Transfer Protocol
418 – Server file transfer under TSS/360 at NASA Ames
414 – File Transfer Protocol (FTP) status and further comments
412 – User FTP Documentation
385 – Comments on the File Transfer Protocol
310 – Another Look at Data and File Transfer Protocols
294 – The Use of "Set Data Type" Transaction in File Transfer Protocol
281 – Suggested addition to File Transfer Protocol
269 – Some Experience with File Transfer
264, 171 - The Data Transfer Protocol
250 – Some thoughts on file transfer
242 – Data Descriptive Language for Shared Data
238 – Comments on DTP and FTP proposals
163 – Data transfer protocols
141 – Comments on RFC 114: A File Transfer Protocol
133 – File Transfer and Recovery

6d. Domain Name System (DNS)

2673 –Binary Labels in the Domain Name System
2672 –Non-Terminal DNS Name Redirection
2671 –Extension Mechanisms for DNS (EDNS0)
2606 –Reserved Top Level DNS Names
2541 –DNS Security Operational Considerations
2540 –Detached Domain Name System (DNS) Information
2539 –Storage of Diffie-Hellman Keys in the Domain Name System (DNS)
2535 –Domain Name System Security Extensions
2517 –Building Directories from DNS: Experiences from WWWSeeker
2352, 2240 - A Convention For Using Legal Names as Domain Names
2317 –Classless IN-ADDR.ARPA delegation
2308 –Negative Caching of DNS Queries (DNS NCACHE)
2230 –Key Exchange Delegation Record for the DNS
2219 –Use of DNS Aliases for Network Services
2182 –Selection and Operation of Secondary DNS Servers
2181 –Clarifications to the DNS Specification
2137 –Secure Domain Name System Dynamic Update
2136 –Dynamic Updates in the Domain Name System (DNS UPDATE)
2065 –Domain Name System Security Extensions
2052 –A DNS RR for specifying the location of services (DNS SRV)
2010 –Operational Criteria for Root Name Servers
1996 –A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)
1995 –Incremental Zone Transfer in DNS
1982 –Serial Number Arithmetic
1912, 1537 - Common DNS Operational and Configuration Errors
1876 –A Means for Expressing Location Information in the Domain Name System
1794 –DNS Support for Load Balancing

1713 –Tools for DNS debugging
1712 –DNS Encoding of Geographical Location
1706, 1637, 1348 - DNS NSAP Resource Records
1591 –Domain Name System Structure and Delegation
1536 –Common DNS Implementation Errors and Suggested Fixes
1535 –A Security Problem and Proposed Correction With Widely Deployed DNS Software
1480, 1386 - The US Domain
1464 –Using the Domain Name System To Store Arbitrary String Attributes
1394 –Relationship of Telex Answerback Codes to Internet Domains
1183 –New DNS RR Definitions
1101 –DNS encoding of network names and other types
1035 –Domain names - implementation and specification
1034 –Domain names - concepts and facilities
1033 –Domain administrators operations guide
1032 –Domain administrators guide
1031 –MILNET name domain transition
973 – Domain system changes and observations
953, 811 - Hostname Server
921, 897 - Domain name system implementation schedule - revised
920 – Domain requirements
883 – Domain names: Implementation specification
882 – Domain names: Concepts and facilities
881 – Domain names plan and schedule
830 – Distributed system for Internet name service
819 – Domain naming convention for Internet user applications
799 – Internet name domains
756 – NIC name server - a datagram-based information utility
752 – Universal host table

6e. Mail and Message Systems (SMTP, MIME, POP, IMAP, X.400)

2683 –IMAP4 Implementation Recommendations
2646 –The Text/Plain Format Parameter
2645 –ON-DEMAND MAIL RELAY (ODMR) SMTP with Dynamic IP Addresses
2634 –Enhanced Security Services for S/MIME
2633 –S/MIME Version 3 Message Specification
2632 –S/MIME Version 3 Certificate Handling
2595 –Using TLS with IMAP, POP3 and ACAP
2586 –The Audio/L16 MIME content type
2557, 2110 - MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)
2554 –SMTP Service Extension for Authentication
2530 –Indicating Supported Media Features Using Extensions to DSN and MDN
2524 –Neda's Efficient Mail Submission and Delivery (EMSD) Protocol Specification Version 1.3
2505 –Anti-Spam Recommendations for SMTP MTAs
2503 –MIME Types for Use with the ISO ILL Protocol
2487 –SMTP Service Extension for Secure SMTP over TLS
2480 –Gateways and MIME Security Multiparts
2476 –Message Submission
2449 –POP3 Extension Mechanism
2442 –The Batch SMTP Media Type
2426 –vCard MIME Directory Profile
2425 –A MIME Content-Type for Directory Information

2424 –Content Duration MIME Header Definition
2387, 2112, 1872 - The MIME Multipart/Related Content-type
2384 –POP URL Scheme
2359 –IMAP4 UIDPLUS extension
2342 –IMAP4 Namespace
2318 –The text/css Media Type
2312 –S/MIME Version 2 Certificate Handling
2311 –S/MIME Version 2 Message Specification
2302 –Tag Image File Format (TIFF) - image/tiff MIME Sub-type Registration
2298 –An Extensible Message Format for Message Disposition Notifications
2231, 2184 - MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations
2221 –IMAP4 Login Referrals
2220 –The Application/MARC Content-type
2197, 1854 - SMTP Service Extension for Command Pipelining
2195, 2095 - IMAP/POP AUTHorize Extension for Simple Challenge/Response
2193 –IMAP4 Mailbox Referrals
2192 –IMAP URL Scheme
2180 –IMAP4 Multi-Accessed Mailbox Practice
2177 –IMAP4 IDLE command
2164, 1838 - Use of an X.500/LDAP directory to support MIXER address mapping
2163, 1664 - Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM)
2162, 1405 - MaXIM-11 - Mapping between X.400 / Internet mail and Mail-11 mail
2161 –A MIME Body Part for ODA
2160 –Carrying PostScript in X.400 and MIME
2158 –X.400 Image Body Parts
2157 –Mapping between X.400 and RFC-822/MIME Message Bodies
2156, 1495, 1327, 1148, 1138 - MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME
2152, 1642 - UTF-7 A Mail-Safe Transformation Format of Unicode
2142 –Mailbox Names for Common Services, Roles and Functions
2088 –IMAP4 non-synchronizing literals
2087 –IMAP4 QUOTA extension
2086 –IMAP4 ACL extension
2077 –The Model Primary Content Type for Multipurpose Internet Mail Extensions
2076 –Common Internet Message Headers
2062 –Internet Message Access Protocol - Obsolete Syntax
2061, 2060, 1730 - IMAP4 Compatibility with IMAP2bis
2049 –Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples
2048 –Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures
2047, 1522, 1342 - MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text
2046 –Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
2045, 1521, 1341 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
2034 –SMTP Service Extension for Returning Enhanced Error Codes
2033 –Local Mail Transfer Protocol
2017 –Definition of the URL MIME External-Body Access-Type
2015 –MIME Security with Pretty Good Privacy (PGP)
1985 –SMTP Service Extension for Remote Message Queue Starting

1957 –Some Observations on Implementations of the Post Office Protocol (POP3)
1939, 1725, 1460, 1225, 1082, 1081 - Post Office Protocol - Version 3
1896, 1563, 1523 - The text/enriched MIME Content-type
1895 –The Application/CALS-1840 Content-type
1894 –An Extensible Message Format for Delivery Status Notifications
1893 –Enhanced Mail System Status Codes
1892 –The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages
1891 –SMTP Service Extension for Delivery Status Notifications
1873 –Message/External-Body Content-ID Access Type
1870, 1653, 1427 - SMTP Service Extension for Message Size Declaration
1869, 1651, 1425 - SMTP Service Extensions
1864, 1544 - The Content-MD5 Header Field
1848 –MIME Object Security Services
1847 –Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted
1846 –SMTP 521 Reply Code
1845 –SMTP Service Extension for Checkpoint/Restart
1844, 1820 - Multimedia E-mail (MIME) User Agent Checklist
1830 –SMTP Service Extensions for Transmission of Large and Binary MIME Messages
1807, 1357 - A Format for Bibliographic Records
1806 –Communicating Presentation Information in Internet Messages: The Content-Disposition Header
1767 –MIME Encapsulation of EDI Objects
1741 –MIME Content Type for BinHex Encoded Files
1740 –MIME Encapsulation of Macintosh Files - MacMIME
1734 –POP3 AUTHentication command
1733 –Distributed Electronic Mail Models in IMAP4
1732 –IMAP4 Compatibility with IMAP2 and IMAP2bis
1731 –IMAP4 Authentication Mechanisms
1711 –Classifications in E-mail Routing
1685 –Writing X.400 O/R Names
1652, 1426 - SMTP Service Extension for 8bit-MIMEtransport
1649 –Operational Requirements for X.400 Management Domains in the GO-MHS Community
1648 –Postmaster Convention for X.400 Operations
1641 –Using Unicode with MIME
1616 –X.400(1988) for the Academic and Research Community in Europe
1615 –Migrating from X.400(84) to X.400(88)
1590 –Media Type Registration Procedure
1556 –Handling of Bi-directional Texts in MIME
1524 –A User Agent Configuration Mechanism For Multimedia Mail Format Information
1506 –A Tutorial on Gatewaying between X.400 and Internet Mail
1505, 1154 - Encoding Header Field for Internet Messages
1502 –X.400 Use of Extended Character Sets
1496 –Rules for downgrading messages from X.400/88 to X.400/84 when MIME content-types are present in the messages
1494 –Equivalences between 1988 X.400 and RFC-822 Message Bodies
1428 –Transition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME
1344 –Implications of MIME for Internet Mail Gateways
1343 –A User Agent Configuration Mechanism for Multimedia Mail Format Information

1339 –Remote Mail Checking Protocol
1328 –X.400 1988 to 1984 downgrading
1211 –Problems with the maintenance of large mailing lists
1204 –Message Posting Protocol (MPP)
1203, 1176, 1064 - Interactive Mail Access Protocol: Version 3
1168 –Intermail and Commercial Mail Relay services
1153 –Digest message format
1137 –Mapping between full RFC 822 and RFC 822 with restricted encoding
1090 –SMTP on X.25
1056, 993, 984 - PCMAIL: A distributed mail system for personal computers
1049 –Content-type header field for Internet messages
1047 –Duplicate messages and SMTP
1026, 987 - Addendum to RFC 987: (Mapping between X.400 and RFC-822)
977 – Network News Transfer Protocol
976 – UUCP mail interchange format standard
974 – Mail routing and the domain system
937, 918 - Post Office Protocol: Version 2
934 – Proposed standard for message encapsulation
915 – Network mail path service
886 – Proposed standard for message header munging
841 – Specification for message format for Computer Based Message Systems
822 – Standard for the format of ARPA Internet text messages
821, 788 - Simple Mail Transfer Protocol
806 – Proposed Federal Information Processing Standard: Specification for message format for computer based message systems
786 – Mail Transfer Protocol: ISI TOPS20 MTP-NIMAIL interface
785 – Mail Transfer Protocol: ISI TOPS20 file definitions
784 – Mail Transfer Protocol: ISI TOPS20 implementation
780, 772 - Mail Transfer Protocol
771 – Mail transition plan
763 – Role mailboxes
757 – Suggested solution to the naming, addressing, and delivery problem for ARPANET message systems
754 – Out-of-net host addresses for mail
753 – Internet Message Protocol
751 – Survey of FTP mail and MLFL
744 – MARS - a Message Archiving and Retrieval Service
733 – Standard for the format of ARPA network text messages
724 – Proposed official standard for the format of ARPA Network messages
720 – Address Specification Syntax for Network Mail
706 – On the junk mail problem
680 – Message Transmission Protocol
644 – On the problem of signature authentication for network mail
577 – Mail priority
574 – Announcement of a mail facility at UCSB
561 – Standardizing Network Mail Headers
555 – Responses to critiques of the proposed mail protocol
539, 524 - Thoughts on the mail protocol proposed in RFC 524
498 – On mail service to CCN
491 – What is "Free"?
475 – FTP and network mail system
458 – Mail retrieval via FTP
333 – Proposed experiment with a Message Switching Protocol

278, 224, 221, 196 - Revision of the Mail Box Protocol

6f. Facsimile and Bitmaps

2639 –Internet Printing Protocol/1.0: Implementer's Guide
2569 –Mapping between LPD and IPP Protocols
2568 –Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol
2567 –Design Goals for an Internet Printing Protocol
2566 –Internet Printing Protocol/1.0: Model and Semantics
2565 –Internet Printing Protocol/1.0: Encoding and Transport
2542 –Terminology and Goals for Internet Fax
2534 –Media Features for Display, Print, and Fax
2532 –Extended Facsimile Using Internet Mail
2531 –Content Feature Schema for Internet Fax
2306 –Tag Image File Format (TIFF) - F Profile for Facsimile
2305 –A Simple Mode of Facsimile Using Internet Mail
2304 –Minimal FAX address format in Internet Mail
2303 –Minimal PSTN address format in Internet Mail
2301 –File Format for Internet Fax
2159 –A MIME Body Part for FAX
2083 –PNG (Portable Network Graphics) Specification Version 1.0
1529, 1528, 1486 - Principles of Operation for the TPC.INT Subdomain: Remote Printing -- Administrative Policies
1314 –A File Format for the Exchange of Images in the Internet
809 – UCL facsimile system
804 – CCITT draft recommendation T.4
803 – Dacom 450/500 facsimile data transcoding
798 – Decoding facsimile data from the Rapicom 450
797 – Format for Bitmap files
769 – Rapicom 450 facsimile file format

6g. Graphics and Window Systems

1198 –FYI on the X window system
1013 –X Window System Protocol, version 11: Alpha update April 1987
965 – Format for a graphical communication protocol
553 – Draft design for a text/graphics protocol
493 – Graphics Protocol
401 – Conversion of NGP-0 Coordinates to Device Specific Coordinates
398 – ICP Sockets
387 – Some experiences in implementing Network Graphics Protocol Level 0
351 – Graphics information form for the ARPANET graphics resources notebook
336 – Level 0 Graphic Input Protocol
296 – DS-1 display system
292 – Graphics Protocol: Level 0 only
285 – Network graphics
268 – Graphics facilities information
199 – Suggestions for a network data-tablet graphics protocol
192 – Some factors which a Network Graphics Protocol must consider
191 – Graphics implementation and conceptualization at Augmentation Research Center
186 – Network graphics loader
184 – Proposed graphic display modes
181, 177 - Modifications to RFC 177
178 – Network graphic attention handling

125, 86 - Response to RFC 86: Proposal for Network Standard Format for a Graphics Data Stream
94 – Some thoughts on Network Graphics

6h. Data Management

304 – Data management system proposal for the ARPA network
195 – Data computers-data descriptions and access language
194 – The Data Reconfiguration Service -- Compiler/Interpreter Implementation Notes
166 – Data Reconfiguration Service: An implementation specification
144 – Data sharing on computer networks
138 – Status report on proposed Data Reconfiguration Service
83 – Language-machine for data reconfiguration

6i. Remote Job Entry (NETRJE, NETRJS)

740, 599, 589, 325, 189, 88 - NETRJS Protocol
725 – RJE protocol for a resource sharing network
499 – Harvard's network RJE
490 – Surrogate RJS for UCLA-CCN
477, 436 - Remote Job Service at UCSB
407 – Remote Job Entry Protocol
368 – Comments on "Proposed Remote Job Entry Protocol"
360 – Proposed Remote Job Entry Protocol
338 – EBCDIC/ASCII Mapping for Network RJE
307 – Using network Remote Job Entry
283 – NETRJT: Remote Job Service Protocol for TIPS
105 – Network Specifications for Remote Job Entry and Remote Job Output Retrieval at UCSB

6j. Remote Procedure Call (RPC)

2695 –Authentication Mechanisms for ONC RPC
2203 –RPCSEC_GSS Protocol Specification
1833 –Binding Protocols for ONC RPC Version 2
1831 –RPC: Remote Procedure Call Protocol Specification Version 2
1057 –RPC: Remote Procedure Call Protocol specification: Version 2
1050 –RPC: Remote Procedure Call Protocol specification

6k. Time and Date (NTP)

2030, 1769, 1361 - Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI
1708 –NTP PICS PROFORMA - For the Network Time Protocol Version 3
1589 –A Kernel Model for Precision Timekeeping
1305, 1119, 1059 - Network Time Protocol (Version 3) Specification, Implementation
1165 –Network Time Protocol (NTP) over the OSI Remote Operations Service
1129 –Internet time synchronization: The Network Time Protocol
1128 –Measured performance of the Network Time Protocol in the Internet system
958, 957, 956 - Network Time Protocol (NTP)
868 – Time Protocol
867 – Daytime Protocol
778 – DCNET Internet Clock Service
738 – Time server
685 – Response time in cross network debugging
34 – Some Brief Preliminary Notes on the Augmentation Research Center Clock
32 – Connecting M.I.T

28 – Time Standards

6l. Presentation and Representation (XDR, Character Encoding, HTML, XML)

2706 –ECML v1: Field Names for E-Commerce
2659 –Security Extensions For HTML
2482 –Language Tagging in Unicode Plain Text
2413 –Dublin Core Metadata for Resource Discovery
2376 –XML Media Types
2346 –Making Postscript and PDF International
2319 –Ukrainian Character Set KOI8-U
2279, 2044 - UTF-8, a transformation format of ISO 10646
2237 –Japanese Character Encoding for Internet Messages
2183 –Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field
2070 –Internationalization of the Hypertext Markup Language
1980 –A Proposed Extension to HTML : Client-Side Image Maps
1952 –GZIP file format specification version 4.3
1951 –DEFLATE Compressed Data Format Specification version 1.3
1950 –ZLIB Compressed Data Format Specification version 3.3
1947 –Greek Character Encoding for Electronic Mail Messages
1942 –HTML Tables
1922 –Chinese Character Encoding for Internet Messages
1874 –SGML Media Types
1867 –Form-based File Upload in HTML
1866 –Hypertext Markup Language - 2.0
1843 –HZ - A Data Format for Exchanging Files of Arbitrarily Mixed Chinese and ASCII characters
1842 –ASCII Printable Characters-Based Chinese Character Encoding for Internet Messages
1832 –XDR: External Data Representation Standard
1815 –Character Sets ISO-10646 and ISO-10646-J-1
1766 –Tags for the Identification of Languages
1557 –Korean Character Encoding for Internet Messages
1555 –Hebrew Character Encoding for Internet Messages
1554 –ISO-2022-JP-2: Multilingual Extension of ISO-2022-JP
1489 –Registration of a Cyrillic Character Set
1468 –Japanese Character Encoding for Internet Messages
1456 –Conventions for Encoding the Vietnamese Language VISCII: VIetnamese Standard Code for Information Interchange VIQR: VIetnamese Quoted-Readable Specification
1278 –A string encoding of Presentation Address
1197 –Using ODA for translating multimedia information
1014 –XDR: External Data Representation standard
1003 –Issues in defining an equations representation standard

6m. Network Management (SNMP, CMOT, RMON)

2593 –Script MIB Extensibility Protocol Version 1.0
2580, 1904, 1444 - Conformance Statements for SMIv2
2579, 1903, 1443 - Textual Conventions for SMIv2
2578, 1902, 1442 - Structure of Management Information Version 2 (SMIv2)
2575, 2275, 2265 - View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
2574, 2274, 2264 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)

2573, 2273, 2263 - SNMP Applications
2572, 2272, 2262 - Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
2571, 2271, 2261 - An Architecture for Describing SNMP Management Frameworks
2570 –Introduction to Version 3 of the Internet-standard Network Management Framework
2493 –Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals
2438 –Advancement of MIB specifications on the IETF Standards Track
2257 –Agent Extensibility (AgentX) Protocol Version 1
2107 –Ascend Tunnel Management Protocol - ATMP
2089 –V2ToV1 Mapping SNMPv2 onto SNMPv1 within a bi-lingual SNMP agent
2039 –Applicability of Standards Track MIBs to Management of World Wide Web Servers
1910 –User-based Security Model for SNMPv2
1909 –An Administrative Infrastructure for SNMPv2
1908, 1452 - Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework
1906, 1449 - Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)
1905, 1448 - Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
1901 –Introduction to Community-based SNMPv2
1856 –The Opstat Client-Server Model for Statistics Retrieval
1592, 1228 - Simple Network Management Protocol Distributed Protocol Interface Version 2.0
1503 –Algorithms for Automating Administration in SNMPv2 Managers
1446 –Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)
1445 –Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)
1441 –Introduction to version 2 of the Internet-standard Network Management Framework
1420, 1298 - SNMP over IPX
1419 –SNMP over AppleTalk
1418, 1283, 1161 - SNMP over OSI
1369 –Implementation Notes and Experience for the Internet Ethernet MIB
1352 –SNMP Security Protocols
1351 –SNMP Administrative Model
1346 –Resource Allocation, Control, and Accounting for the Use of Network Resources
1303 –A Convention for Describing SNMP-based Agents
1270 –SNMP Communications Services
1239 –Reassignment of experimental MIBs to standard MIBs
1224 –Techniques for managing asynchronously generated alerts
1215 –Convention for defining traps for use with the SNMP
1212 –Concise MIB definitions
1189, 1095 - Common Management Information Services and Protocols for the Internet (CMOT and CMIP)
1187 –Bulk Table Retrieval with the SNMP
1157, 1098, 1067 - Simple Network Management Protocol (SNMP)
1155, 1065 - Structure and identification of management information for TCP/IP-based internets

1109 –Report of the second Ad Hoc Network Management Review Group
1089 –SNMP over Ethernet
1076 –HEMS monitoring and control language
1028 –Simple Gateway Monitoring Protocol
1024 –HEMS variable definitions
1023 –HEMS monitoring and control language
1022 –High-level Entity Management Protocol (HEMP)
1021 –High-level Entity Management System (HEMS)

6n. Management Information Base Definitions (MIB)

2720, 2064 - Traffic Flow Measurement: Meter MIB
2677 –Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP)
2674 –Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions
2670 –Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces
2669 –DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems
2668, 2239 - Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)
2667 –IP Tunnel MIB
2666 –Definitions of Object Identifiers for Identifying Ethernet Chip Sets
2665, 2358, 1650 - Definitions of Managed Objects for the Ethernet-like Interface Types
2662 –Definitions of Managed Objects for the ADSL Lines
2621 –RADIUS Accounting Server MIB
2620 –RADIUS Accounting Client MIB
2619 –RADIUS Authentication Server MIB
2618 –RADIUS Authentication Client MIB
2613 –Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0
2605, 1567 - Directory Server Monitoring MIB
2594 –Definitions of Managed Objects for WWW Services
2592 –Definitions of Managed Objects for the Delegation of Management Script
2591 –Definitions of Managed Objects for Scheduling Management Operations
2584 –Definitions of Managed Objects for APPN/HPR in IP Networks
2564 –Application Management MIB
2562 –Definitions of Protocol and Managed Objects for TN3270E Response Time Collection Using SMIv2 (TN3270E-RT-MIB)
2561 –Base Definitions of Managed Objects for TN3270E Using SMIv2
2558, 1595 - Definitions of Managed Objects for the SONET/SDH Interface Type
2515, 1695 - Definitions of Managed Objects for ATM Management
2514 –Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management
2513 –Managed Objects for Controlling the Collection and Storage of Accounting Information for Connection-Oriented Networks
2512 –Accounting Information for ATM Networks
2496, 1407, 1233 - Definitions of Managed Object for the DS3/E3 Interface Type
2495, 1406, 1232 - Definitions of Managed Objects for the DS1, E1, DS2 and E2 Interface Types
2494 –Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type
2457 –Definitions of Managed Objects for Extended Border Node
2456 –Definitions of Managed Objects for APPN TRAPS

2455, 2155 - Definitions of Managed Objects for APPN
2417, 2366 - Definitions of Managed Objects for Multicast over UNI 3.0/3.1 based ATM Networks
2320 –Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB)
2287 –Definitions of System-Level Managed Objects for Applications
2266 –Definitions of Managed Objects for IEEE 802.12 Repeater Devices
2249, 1566 - Mail Monitoring MIB
2248, 1565 - Network Services Monitoring MIB
2238 –Definitions of Managed Objects for HPR using SMIv2
2233, 1573, 1229 - The Interfaces Group MIB using SMIv2
2232 –Definitions of Managed Objects for DLUR using SMIv2
2214 –Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2
2213 –Integrated Services Management Information Base using SMIv2
2128 –Dial Control Management Information Base using SMIv2
2127 –ISDN Management Information Base using SMIv2
2115, 1315 - Management Information Base for Frame Relay DTEs Using SMIv2
2108, 1516, 1368 - Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2
2096, 1354 - IP Forwarding Table MIB
2074 –Remote Network Monitoring MIB Protocol Identifiers
2063 –Traffic Flow Measurement: Architecture
2051 –Definitions of Managed Objects for APPC using SMIv2
2037 –Entity MIB using SMIv2
2024 –Definitions of Managed Objects for Data Link Switching using SMIv2
2021 –Remote Network Monitoring Management Information Base Version 2 using SMIv2
2020 –IEEE 802.12 Interface MIB
2013 –SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2
2012 –SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2
2011 –SNMPv2 Management Information Base for the Internet Protocol using SMIv2
2006 –The Definitions of Managed Objects for IP Mobility Support using SMIv2
1907, 1450 - Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
1850, 1253, 1252, 1248 - OSPF Version 2 Management Information Base
1792 –TCP/IPX Connection Mib Specification
1759 –Printer MIB
1757, 1271 - Remote Network Monitoring Management Information Base
1749 –IEEE 802.5 Station Source Routing MIB using SMIv2
1748, 1743, 1231 - IEEE 802.5 MIB using SMIv2
1747 –Definitions of Managed Objects for SNA Data Link Control (SDLC) using SMIv2
1742, 1243 - AppleTalk Management Information Base II
1724, 1389 - RIP Version 2 MIB Extension
1697 –Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIv2
1696 –Modem Management Information Base (MIB) using SMIv2
1694, 1304 - Definitions of Managed Objects for SMDS Interfaces using SMIv2
1666 –Definitions of Managed Objects for SNA NAUs using SMIv2

1665 –Definitions of Managed Objects for SNA NAUs using SMIv2
1660, 1318 - Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIv2
1659, 1317 - Definitions of Managed Objects for RS-232-like Hardware Devices using SMIv2
1658, 1316 - Definitions of Managed Objects for Character Stream Devices using SMIv2
1657 –Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2
1643, 1623, 1398, 1284 - Definitions of Managed Objects for the Ethernet-like Interface Types
1628 –UPS Management Information Base
1612 –DNS Resolver MIB Extensions
1611 –DNS Server MIB Extensions
1604, 1596 - Definitions of Managed Objects for Frame Relay Service
1593 –SNA APPN Node MIB
1559, 1289 - DECnet Phase IV MIB Extensions
1525, 1493, 1286 - Definitions of Managed Objects for Source Routing Bridges
1515 –Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)
1514 –Host Resources MIB
1513 –Token Ring Extensions to the Remote Network Monitoring MIB
1512, 1285 - FDDI Management Information Base
1474 –The Definitions of Managed Objects for the Bridge Network Control Protocol of the Point-to-Point Protocol
1473 –The Definitions of Managed Objects for the IP Network Control Protocol of the Point-to-Point Protocol
1472 –The Definitions of Managed Objects for the Security Protocols of the Point-to-Point Protocol
1471 –The Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol
1461 –SNMP MIB extension for Multiprotocol Interconnect over X.25
1451 –Manager-to-Manager Management Information Base
1447 –Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)
1414 –Identification MIB
1382 –SNMP MIB Extension for the X.25 Packet Layer
1381 –SNMP MIB Extension for X.25 LAPB
1353 –Definitions of Managed Objects for Administration of SNMP Parties
1269 –Definitions of Managed Objects for the Border Gateway Protocol: Version 3
1230 –IEEE 802.4 Token Bus MIB
1227 –SNMP MUX protocol and MIB
1214 –OSI internet management: Management Information Base
1213, 1158, 1156, 1066 - Management Information Base for Network Management of TCP/IP-based internets:MIB-II

6o. Directory Services (X.500, LDAP, Whitepages)

2714 –Schema for Representing CORBA Object References in an LDAP Directory
2713 –Schema for Representing Java(tm) Objects in an LDAP Directory
2696 –LDAP Control Extension for Simple Paged Results Manipulation
2657 –LDAPv2 Client vs the Index Mesh
2649 –An LDAP Control and Schema for Holding Operation Signatures
2596 –Use of Language Codes in LDAP
2589 –Lightweight Directory Access Protocol (v3): Extensions for Dynamic Directory Services

2587 –Internet X.509 Public Key Infrastructure LDAPv2 Schema
2585 –Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP
2560 –X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP
2559 –Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2
2528 –Internet X.509 Public Key Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates
2527 –Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
2511 –Internet X.509 Certificate Request Message Format
2510 –Internet X.509 Public Key Infrastructure Certificate Management Protocols
2459 –Internet X.509 Public Key Infrastructure Certificate and CRL Profile
2377 –Naming Plan for Internet Directory-Enabled Applications
2307 –An Approach for Using LDAP as a Network Information Service
2294, 1836 - Representing the O/R Address hierarchy in the X.500 Directory Information Tree
2293, 1837 - Representing Tables and Subtrees in the X.500 Directory
2256 –A Summary of the X.500(96) User Schema for use with LDAPv3
2255 –The LDAP URL Format
2254, 1960, 1558 - The String Representation of LDAP Search Filters
2253 –Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
2252 –Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
2251 –Lightweight Directory Access Protocol (v3)
2247 –Using Domains in LDAP/X.500 Distinguished Names
2218 –A Common Schema for the Internet White Pages Service
2148 –Deployment of the Internet White Pages Service
2120 –Managing the X.500 Root Naming Context
2116, 1632, 1292 - X.500 Implementations Catalog-96
2079 –Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)
1959 –An LDAP URL Format
1943 –Building an X.500 Directory Service in the US
1823 –The LDAP Application Program Interface
1804 –Schema Publishing in X.500 Directory
1803 –Recommendations for an X.500 Production Directory Service
1802 –Introducing Project Long Bud: Internet Pilot Project for the Deployment of X.500 Directory Information in Support of X.400 Routing
1801 –MHS use of the X.500 Directory to support MHS Routing
1798 –Connection-less Lightweight X.500 Directory Access Protocol
1781, 1484 - Using the OSI Directory to Achieve User Friendly Naming
1779, 1485 - A String Representation of Distinguished Names
1778, 1488 - The String Representation of Standard Attribute Syntaxes
1777, 1487 - Lightweight Directory Access Protocol
1684 –Introduction to White Pages Services based on X.500
1617, 1384 - Naming and Structuring Guidelines for X.500 Directory Pilots
1609 –Charting Networks in the X.500 Directory
1608 –Representing IP Information in the X.500 Directory
1564 –DSA Metrics (OSI-DS 34 (v3))
1562 –Naming Guidelines for the AARNet X.500 Directory Service
1491 –A Survey of Advanced Usages of X.500

1431 –DUA Metrics (OSI-DS 33 (v2))
1430 –A Strategic Plan for Deploying an Internet X.500 Directory Service
1373 –Portable DUAs
1309 –Technical Overview of Directory Services Using the X.500 Protocol
1308 –Executive Introduction to Directory Services Using the X.500 Protocol
1279 –X.500 and Domains
1277 –Encoding Network Addresses to Support Operation over Non-OSI Lower Layers
1276 –Replication and Distributed Operations extensions to provide an Internet Directory using X.500
1275 –Replication Requirements to provide an Internet Directory using X.500
1274 –The COSINE and Internet X.500 Schema
1255, 1218 - A Naming Scheme for c=US
1249 –DIXIE Protocol Specification
1202 –Directory Assistance service
1107 –Plan for Internet directory services

6p. Information Services (HTTP, Gopher, WAIS)

2718 –Guidelines for new URL Schemes
2660 –The Secure HyperText Transfer Protocol
2656 –Registration Procedures for SOIF Template Types
2655 –CIP Index Object Format for SOIF Objects
2654 –A Tagged Index Object for use in the Common Indexing Protocol
2653 –CIP Transport Protocols
2652 –MIME Object Definitions for the Common Indexing Protocol (CIP)
2651 –The Architecture of the Common Indexing Protocol (CIP)
2617, 2069 - HTTP Authentication: Basic and Digest Access Authentication
2616, 2068 - Hypertext Transfer Protocol -- HTTP/1.1
2611 –URN Namespace Definition Mechanisms
2518 –HTTP Extensions for Distributed Authoring -- WEBDAV
2483 –URI Resolution Services Necessary for URN Resolution
2397 –The "data" URL scheme
2396 –Uniform Resource Identifiers (URI): Generic Syntax
2392, 2111 - Content-ID and Message-ID Uniform Resource Locators
2388 –Returning Values from Forms: multipart/form-data
2378 –The CCSO Nameserver (Ph) Architecture
2369 –The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields
2368 –The mailto URL scheme
2345 –Domain Names and Company Name Retrieval
2310 –The Safe Response Header Field
2296 –HTTP Remote Variant Selection Algorithm -- RVSA/1.0
2295 –Transparent Content Negotiation in HTTP
2291 –Requirements for a Distributed Authoring and Versioning Protocol for the World Wide Web
2288 –Using Existing Bibliographic Identifiers as Uniform Resource Names
2276 –Architectural Principles of Uniform Resource Name Resolution
2259, 2258 - Simple Nomenclator Query Protocol (SNQP)
2227 –Simple Hit-Metering and Usage-Limiting for HTTP
2187, 2186 - Application of Internet Cache Protocol (ICP), version 2
2169 –A Trivial Convention for using HTTP in URN Resolution
2168 –Resolution of Uniform Resource Identifiers using the Domain Name System
2167, 1714 - Referral Whois (RWhois) Protocol V1.5
2145 –Use and Interpretation of HTTP Version Numbers

2141 –URN Syntax
2122 –VEMMI URL Specification
2109 –HTTP State Management Mechanism
2084 –Considerations for Web Transaction Security
2056 –Uniform Resource Locators for Z39.50
1945 –Hypertext Transfer Protocol -- HTTP/1.0
1914 –How to Interact with a Whois++ Mesh
1913 –Architecture of the Whois++ Index Service
1835 –Architecture of the WHOIS++ service
1834 –Whois and Network Information Lookup Service, Whois++
1808 –Relative Uniform Resource Locators
1738 –Uniform Resource Locators (URL)
1737 –Functional Requirements for Uniform Resource Names
1736 –Functional Recommendations for Internet Resource Locators
1729 –Using the Z39.50 Information Retrieval Protocol
1728 –Resource Transponders
1727 –A Vision of an Integrated Internet Information Service
1630 –Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web
1625 –WAIS over Z39.50-1988
1614 –Network Access to Multimedia Information
1436 –The Internet Gopher Protocol (a distributed document search and retrieval protocol)
954, 812 - NICNAME/WHOIS

6q. Bootstrap and Configuration Protocols (BOOTP, DHCP)

2563 –DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients
2485 –DHCP Option for The Open Group's User Authentication Protocol
2242 –NetWare/IP Domain Name and Information
2241 –DHCP Options for Novell Directory Services
2132, 1533, 1497, 1395, 1084, 1048 - DHCP Options and BOOTP Vendor Extensions
2131, 1541, 1531 - Dynamic Host Configuration Protocol
1542, 1532 - Clarifications and Extensions for the Bootstrap Protocol
1534 –Interoperation Between DHCP and BOOTP
951 – Bootstrap Protocol

6r. Real-Time Multimedia and Quality of Service (RSVP, RTP)

2719 –Framework Architecture for Signaling Transport
2705 –Media Gateway Control Protocol (MGCP) Version 1.0
2689 –Integrated Services Mappings for Low Speed Networks
2688 –Integrated Services Mappings for Low Speed Networks
2658 –RTP Payload Format for PureVoice(tm) Audio
2543 –SIP: Session Initiation Protocol
2490 –A Simulation Model for IP Multicast with RSVP
2458 –Toward the PSTN/Internet Inter-Networking--Pre-PINT Implementations
2448 –AT&T's Error Resilient Video Transmission Technique
2435, 2035 - RTP Payload Format for JPEG-compressed Video
2431 –RTP Payload Format for BT.656 Video Encoding
2430 –A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)
2429 –RTP Payload Format for the 1998 Version of ITU-T Rec
2423, 2422, 2421, 1911 - VPIM Voice Message MIME Sub-type Registration

2386 –A Framework for QoS-based Routing in the Internet
2382 –A Framework for Integrated Services and RSVP over ATM
2381 –Interoperation of Controlled-Load Service and Guaranteed Service with ATM
2380 –RSVP over ATM Implementation Requirements
2379 –RSVP over ATM Implementation Guidelines
2361 –WAVE and AVI Codec Registries
2354 –Options for Repair of Streaming Media
2343 –RTP Payload Format for Bundled MPEG
2327 –SDP: Session Description Protocol
2326 –Real Time Streaming Protocol (RTSP)
2250, 2038 - RTP Payload Format for MPEG1/MPEG2 Video
2216 –Network Element Service Specification Template
2215 –General Characterization Parameters for Integrated Service Network Elements
2212 –Specification of Guaranteed Quality of Service
2211 –Specification of the Controlled-Load Network Element Service
2210 –The Use of RSVP with IETF Integrated Services
2209 –Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules
2208 –Resource ReSerVation Protocol (RSVP) -- Version 1 Applicability Statement Some Guidelines on Deployment
2207 –RSVP Extensions for IPSEC Data Flows
2206 –RSVP Management Information Base using SMIv2
2205 –Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification
2198 –RTP Payload for Redundant Audio Data
2190 –RTP Payload Format for H.263 Video Streams
2032 –RTP Payload Format for H.261 Video Streams
2029 –RTP Payload Format of Sun's CellB Video Encoding
1890 –RTP Profile for Audio and Video Conferences with Minimal Control
1889 –RTP: A Transport Protocol for Real-Time Applications
1821 –Integration of Real-time Services in an IP-ATM Network Architecture
1789 –INETPhone: Telephone Services and Servers on Internet
1257 –Isochronous applications do not require jitter-controlled networks
1193 –Client requirements for real-time communication services
741 – Specifications for the Network Voice Protocol (NVP)

6s. Other

2703 –Protocol-independent Content Negotiation Framework
2614 –An API for Service Location
2610 –DHCP Options for Service Location Protocol
2609 –Service Templates and Service: Schemes
2608, 2165 - Service Location Protocol, Version 2
2552 –Architecture for the Information Brokerage in the ACTS Project GAIA
2533 –A Syntax for Describing Media Feature Sets
2447, 2446, 2445 - iCalendar Message-Based Interoperability Protocol (iMIP)
2244 –ACAP -- Application Configuration Access Protocol
2229 –A Dictionary Server Protocol
2188 –AT&T/Neda's Efficient Short Remote Operations (ESRO) Protocol Specification Version 1.2
2016 –Uniform Resource Agents (URAs)
1861, 1645, 1568 - Simple Network Paging Protocol - Version 3 -Two-Way Enhanced
1756 –Remote Write Protocol - Version 1.0
1703, 1569 - Principles of Operation for the TPC.INT Subdomain: Radio Paging -- Technical Procedures

1692 –Transport Multiplexing Protocol (TMux)
1530 –Principles of Operation for the TPC.INT Subdomain: General Principles and Policy
1492 –An Access Control Protocol, Sometimes Called TACACS
1459 –Internet Relay Chat Protocol
1429 –Listserv Distribute Protocol
1413, 931, 912 - Identification Protocol
1307 –Dynamically Switched Link Control Protocol
1288, 1196, 1194, 742 - The Finger User Information Protocol
1179 –Line printer daemon protocol
978 – Voice File Interchange Protocol (VFIP)
909 – Loader Debugger Protocol
891 – DCN local-network protocols
887 – Resource Location Protocol
866 – Active users
865 – Quote of the Day Protocol
864 – Character Generator Protocol
863, 348 - Discard Protocol
862, 347 - Echo Protocol
767 – Structured format for transmission of multi-media documents
759 – Internet Message Protocol
734 – SUPDUP Protocol
666 – Specification of the Unified User-Level Protocol
621 – NIC user directories at SRI ARC
569 – NETED: A Common Editor for the ARPA Network
470 – Change in socket for TIP news facility
451 – Tentative proposal for a Unified User Level Protocol
109 – Level III Server Protocol for the Lincoln Laboratory NIC 360/67 Host
98, 79 - Logger Protocol Proposal
29 – Response to RFC 28

7. Program Documentation

1761 –Snoop Version 2 Packet Capture File Format
496 – TNLS quick reference card is available
494 – Availability of MIX and MIXAL in the Network
488 – NLS classes at network sites
485 – MIX and MIXAL at UCSB
431 – Update on SMFS Login and Logout
411 – New MULTICS Network Software Features
409 – Tenex interface to UCSB's Simple-Minded File System
399 – SMFS Login and Logout
390 – TSO Scenario
382 – Mathematical Software on the ARPA Network
379 – Using TSO at CCN
373 – Arbitrary Character Sets
350 – User Accounts for UCSB On-Line System
345 – Interest in Mixed Integer Programming (MPSX on NIC 360/91 at CCN)
321 – CBI Networking Activity at MITRE
311 – New Console Attachments to the USCB Host
251 – Weather data
217 – Specifications changes for OLS, RJE/RJOR, and SMFS
174 – UCLA - Computer Science Graphics Overview
122 – Network specifications for UCSB's Simple-Minded File System
121 – Network on-line operators

120 – Network PL1 subprograms
119 – Network Fortran subprograms
74 – Specifications for network use of the UCSB On-Line System

8. Network Specific (also see Section 3)

8a. ARPANET

1005, 878, 851, 802 - ARPANET AHIP-E Host Access Protocol (enhanced AHIP)
852 – ARPANET short blocking feature
789 – Vulnerabilities of network control protocols: An example
745 – JANUS interface specifications
716 – Interim Revision to Appendix F of BBN 1822
704 – IMP/Host and Host/IMP Protocol change
696 – Comments on the IMP/Host and Host/IMP Protocol changes
695 – Official change in Host-Host Protocol
692 – Comments on IMP/Host Protocol changes (RFCs 687 and 690)
690 – Comments on the proposed Host/IMP Protocol changes
687 – IMP/Host and Host/IMP Protocol changes
667 – BBN host ports
660 – Some changes to the IMP and the IMP/Host interface
642 – Ready line philosophy and implementation
638, 633 - IMP/TIP preventive maintenance schedule
632 – Throughput degradations for single packet messages
627 – ASCII text file of hostnames
626 – On a possible lockup condition in IMP subnet due to message sequencing
625 – On-line hostnames service
623 – Comments on on-line host name service
622 – Scheduling IMP/TIP down time
620 – Request for monitor host table updates
619 – Mean round-trip times in the ARPANET
613 – Network connectivity: A response to RFC 603
611 – Two changes to the IMP/Host Protocol to improve user/network communications
606 – Host names on-line
594 – Speedup of Host-IMP interface
591 – Addition to the Very Distant Host specifications
568, 567 - Response to RFC 567 - cross country network bandwidth
548 – Hosts using the IMP Going Down message
547 – Change to the Very Distant Host specification
533 – Message-ID numbers
528 – Software checksumming in the IMP and network reliability
521 – Restricted use of IMP DDT
508 – Real-time data transmission on the ARPANET
476, 434 - IMP/TIP memory retrofit schedule (rev 2)
449, 442 - Current flow-control scheme for IMPSYS
447, 445 - IMP/TIP memory retrofit schedule
417 – Link usage violation
410 – Removal of the 30-Second Delay When Hosts Come Up
406 – Scheduled IMP Software Releases
395 – Switch Settings on IMPs and TIPs
394 – Two Proposed Changes to the IMP-Host Protocol
369 – Evaluation of ARPANET services January-March, 1972
335 – New Interface - IMP/360
312 – Proposed Change in IMP-to-Host Protocol