Internet Architecture Board
INTERNET DRAFT

Steve King, Bay Networks
Ruth Fax, Bay Networks
Dimitry Haskin, Bay Networks
Wenken Ling, Bay Networks
Tom Meehan, Bay Networks
Robert Fink, LBNL
Charles E. Perkins, Sun Microsystems

The Case for IPv6
draft-ietf-iab-case-for-ipv6-04.txt

Status of This Memo

This document is a submission by the Internet Architecture Board (IAB). Comments should be submitted to the iab@isi.edu mailing list.

Status of This Memo

This document is a submission by the IAB Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the iab@isi.edu mailing list.

Distribution of this memo is unlimited.

Abstract

This document outlines the business and technical case for IPv6. It is intended to acquaint both the existing IPv4 community with IPv6, to encourage its support for change, and to attract potential future users of Internet technology.

Contents

1. Introduction

This document was produced at the request of the IAB, based on an existing original. Many of the protocol specifications have become Draft Standards, and are thus quite stable. Some other related specifications are still in progress at the time of this writing, so that the technical details are subject to change, and the references cited may become obsolete. The intended audience includes enterprise network administrators and decision makers, router vendors, host vendors, Internet Service Providers (ISPs) managers, and protocol engineers who are as yet unfamiliar with the basic aspects of IPv6.

The Internet Protocol (IP) has its roots in early research networks of the 1970s, but within the past decade has become the leading network-layer protocol. This means that IP is a primary vehicle for a vast array of client/server and peer-to-peer communications, and the current scale of deployment is straining many aspects of its twenty-year old design [4].

The Internet Engineering Task Force (IETF) has produced specifications (see section 2.1) that define the next-generation IP protocol known as "IPng," or "IPv6." IPv6 is both a near-term and long-range concern for network owners and service providers. IPv6 products have already come to market; on the other hand, IPv6 development work will likely continue well into the next decade. Though it is based on much-needed enhancements to IPv4 standards, IPv6 should be viewed as a new protocol that will provide a firmer base for the continued growth of today's internetworks.

Because it is intended to replace IP (hereafter called IPv4) IPv6 is of considerable importance to businesses, consumers, and network access providers of all sizes. IPv6 is designed to improve upon IPv4's scalability, security, ease-of-configuration, and network management; these issues are central to the competitiveness and performance of all types of network-dependent businesses. IPv4 can be modified to perform some of these functions, but the expectation within the IAB is that the results are likely to be far less useful than what could be obtained by widespread deployment of IPv6. On the other hand IPv6 aims to preserve existing investment as much as possible. End users, industry executives, network administrators, protocol engineers, and many others will benefit from understanding the ways that IPv6 will affect future internetworking and distributed computing applications.

By early 1998 a worldwide IPv6 testing and pre-production deployment network, called the 6BONE, had already reached approximately 400 sites and networks in 40 countries. There are over 50 IPv6 implementations completed or underway worldwide, and over 25 in test or production use on the 6BONE. The 6BONE has been built by an active population of protocol inventors, designers and programmers. They have worked together to solve the questions and problems that might be expected to arise during such a huge project. Their experience has served to validate the expectations of the protocol designers.

This document presents IPv6 issues in two parts:

- The Business Case for IPv6, giving a high-level view of business issues, protocol basics, and current status, and
- The Technical Case for IPv6, which describes more of the functional and technical aspects of IPv6.

2. Part I: The Business Case for IPv6

Given the remarkable growth of the Internet, and business opportunity represented by the Internet, IPv6 is of major interest to business interests, enterprise internetworks, and the global Internet. IPv6 presents all networking interests with a opportunity for global improvements, which is now receiving the collective action that is needed to realize the benefits.

2.1. IPv6: Standardization and Productization Status

IPv6, the Next-Generation Internet Protocol, has been approved as a Draft Standard. A large number of end-user organizations, standards groups, and network vendors have been working together on the specification and testing of early IPv6 implementations. A number of IETF working groups have produced IPv6 specifications that are finished or well underway. Current Draft Standards include:

- RFC 2373: IP Version 6 Addressing Architecture
- RFC 2374: An IPv6 Aggregatable Global Unicast Address Format
- RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
- RFC 2461: Neighbor Discovery for IP Version 6 (IPv6)
- RFC 2462: IPv6 Stateless Address Autoconfiguration
- RFC 2463: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

Current Proposed Standards include:

- RFC 1886: DNS Extensions to support IP version 6
- RFC 1887: An Architecture for IPv6 Unicast Address Allocation
- RFC 1981: Path MTU Discovery for IP version 6
- RFC 2023: IP Version 6 over PPP
- RFC 2080: RIPng for IPv6
- RFC 2147: TCP and UDP over IPv6 Jumbograms

- RFC 2452: IP Version 6 Management Information Base for the Transmission Control Protocol
- RFC 2454: IP Version 6 Management Information Base for the User Datagram Protocol
- RFC 2464: Transmission of IPv6 Packets over Ethernet Networks
- RFC 2465: Management Information Base for IP Version 6: Textual Conventions and General Group
- RFC 2466: Management Information Base for IP Version 6: ICMPv6 Group
- RFC 2467: Transmission of IPv6 Packets over FDDI Networks
- RFC 2470: Transmission of IPv6 Packets over Token Ring Networks
- RFC 2472: IP Version 6 over PPP
- RFC 2473: Generic Packet Tunneling in IPv6 Specification
- RFC 2507: IP Header Compression

There are too many related RFCs and Internet Drafts to list them all here, but among them are included the following:

- RFC 1888: OSI NSAPs and IPv6
- RFC 2133: Basic Socket Interface Extensions for IPv6
- RFC 2292: Advanced Sockets API for IPv6
- RFC 2375: IPv6 Multicast Address Assignments
- RFC 2450: Proposed TLA and NLA Assignment Rules
- RFC 2471: IPv6 Testing Address Allocation
- OSPF for IPv6
- IPv6 Router Alert Option
- Mobility Support in IPv6
- DHCP for IP Version 6
- Router Renumbering for IPv6
- Site prefixes in Neighbor Discovery
- The IPv6 Jumbo Payload Option
- Reserved IPv6 Subnet Anycast Addresses
- Routing of Scoped Addresses in the Internet Protocol Version 6 (IPv6)

Standards work on IPv6 and related components is far enough along that vendors have already committed to a considerable number of development and testing projects. All of the major router vendors have made plans to support IPv6 in their products.

Vendors such as Apple, Digital Equipment, Hewlett-Packard, IBM, Microsoft, Novell, Silicon Graphics and Sun have likewise begun the task of delivering IPv6 on desktop machines and servers. Many organizations are working on IPv6 drivers for the popular UNIX BSD and Linux operating environments. Network software vendors have announced a wide range of support for IPv6 in network applications and communication software products. Software is available from Microsoft for Windows-based clients.

2.2. IPv6 Design Goals

IPv6 has been designed to enable high-performance, scalable internetworks that should operate as needed for decades. Part of the design process involved correcting the inadequacies of IPv4. IPv6 offers a number of enhanced features, such as a larger address space and improved packet formats. Other benefits relate to the fresh start that IPv6 gives to those who build and administer networks. For instance, a well-structured, efficient and adaptable routing hierarchy will be possible. The following sections give an overview of the improvements that IPv6 brings to enterprise networking and the global Internet.

2.2.1. Addressing and Routing

IPv6 helps to solve a number of problems that currently exist within and between enterprises. On the global scale, IPv6 will allow Internet backbone designers to create a flexible and expandable global routing hierarchy. The Internet backbone, where major enterprises and Internet Service Provider (ISP) networks come together, depends upon the maintenance of a hierarchical address system, similar to that of the national and international telephone systems. Large central-office phone switches, for instance, only need a three-digit national area code prefix to route a long-distance telephone call to the correct local exchange. The current IPv4 system also uses an address hierarchy to sort traffic towards networks attached to the Internet backbone.

Without an address hierarchy, backbone routers would be forced to store route table information on the reachability of every network in the world. Given the current number of IP subnets in the world and the growth of the Internet, it is not feasible to manage route tables and updates for so many routes. With a hierarchy, backbone routers can use IP address prefixes to determine how traffic should be routed through the backbone. In recent years, IPv4 has begun to use a technique called Classless InterDomain Routing (CIDR) [33, 17], which uses bit masks to allocate a variable portion of the 32-bit IPv4 address to a network, subnet, or host. CIDR permits "route aggregation" at various levels of the Internet hierarchy, whereby backbone routers can store a single route table entry that provides reachability to many lower- level networks.

But CIDR does not guarantee an efficient and scalable hierarchy. In order to avoid maintaining a separate entry for each route individually, it is important for routes at lower levels of the routing hierarchy, that naturally have longer prefixes, to be collected together (or "summarized") into fewer and less specific routes at higher levels of the routing hierarchy.

Legacy IPv4 address assignments that originated before CIDR and the current access provider hierarchy often do not facilitate summarization. The lack of uniformity of the current hierarchical system, coupled with the rationing of IPv4 addresses, makes Internet addressing and routing quite complicated. These issues affect high-level service providers and consequently individual end users in all types of businesses. Furthermore, renumbering IPv4 sites when changing from one ISP to another, to maintain and improve address/route aggregation, is unnecessarily complicated (and thus more expensive) compared to IPv6's ease of site renumbering (see section 2.2.3).

2.2.2. Eliminating Special Cases

Many of the same problems that exist today in the Internet backbone are also being felt at the level of the enterprise and the individual business user. When an enterprise can't summarize its routes effectively, it becomes puts a larger load on the backbone route tables. If an enterprise can't present globally unique addresses to the Internet, it may be forced to deploy private, isolated address space that isn't visible to the Internet.

Users in private address spaces with non-unique addresses typically require gateways, and possibly Network Address Translators (NATs), to manage their connectivity to the outside world. In such situations, some services are simply not available. A NAT is meant to allow an enterprise to have whatever internal address structure it desires, without concern for integrating internal addresses with the global Internet. This is seen as particularly convenient in the existing IPv4 world, with its more cumbersome address space management. The NAT device sits on the border between the enterprise and the Internet, converting private internal addresses to a smaller pool of globally unique addresses that are passed to the backbone and vice versa (see Figure 1).

NAT may be appropriate in some organizations, particularly if full connectivity with the outside world is not desired. But for enterprises that require robust interaction with the Internet, NAT devices often get in the way. The NAT technique of substituting address fields in each and every packet that leaves and enters the enterprise is very demanding, and presents a bottleneck between the enterprise and the Internet. A NAT may keep up with address conversion in a small network, but as the enterprise's Internet access increases, the NAT's performance must increase in parallel. The bottleneck effect is exacerbated by the difficulty of integrating and synchronizing multiple NAT devices within a single enterprise. Enterprises with NAT are less likely to achieve the reliable high-performance Internet connectivity that is common today with

```
                                     |
                                     |
                Private address space | Unique global addresses
                                     |
                                     |
              ---------------        |
             /               \    +-----+     +----------+
             |  Enterprise   |    |     |     |          |
             |               |----| NAT |-----| Internet |
             |    Network    |    |     |     |          |
             \               /    +-----+     +----------+
              ---------------        |
                                     |
                                     |
                                     |
```

Figure 1: Network Address Translator (NAT)

multiple routers attached to an ISP backbone in an arbitrary mesh fashion. Furthermore, use of NAT devices takes away the additional element of reliability afforded by the possibility for asymmetric routing, since NAT devices require control of traffic directions both to and from internally addressed network nodes.

NAT translators also run into trouble when applications embed IP addresses in the packet payload, above the network layer. This is the case for a number of applications, including certain File Transfer Protocol (FTP) programs, Mobile IP, and the Windows Internet Name Service (WINS) registration process of Windows 95 and Windows NT. Unless a NAT parses every packet all the way to the application level, it is likely to fail to translate some embedded addresses, which will lead to application failures. NAT can also break Domain Name Servers, because they work above the network layer. NATs prevent the use of IP-level security between the endpoints of a transaction. Today, NAT devices are helpful in certain limited scenarios for smaller enterprises, but are considered by many to be generally disadvantageous for the long-term health of the Internet. See [18] for a fuller discussion about the effects of NAT use on the Internet.

2.2.3. Minimizing Administrative Workload

A major component of today's network administration involves the assignment of networking parameters to computers and other network nodes, that are needed before they can begin any sort of network operation. Information such as an IP address, DNS server, default router, and other configuration details have to be installed at each network node. In many cases, this is still done by manual configuration, either by the network administration, or worse yet by the users themselves. Recent efforts to shift this administrative load onto departmental servers have focussed on deployment of the Dynamic Host Configuration Protocol (DHCP) [16, 1], but this comes along with its own administrative difficulties.

IPv4's limitations also aggravate the occasional need in many organizations to renumber network devices -- i.e., assign new IP addresses to them. When an enterprise changes ISPs, it may have to either renumber all addresses to match the new ISP-assigned prefix, or implement Network Address Translation devices (NATs). Renumbering may be indicated when a corporation undergoes a merger or an acquisition with consequent network consolidation. Since routing prefixes are assigned to reflect the routing topology of the enterprise networks and the number of nodes attached to the particular network links, there are two ways that the choice of routing prefixes can become inconvenient or incorrect:

1. The routing prefix can become too long for the administration to be able to increase the number of nodes that can be attached to the particular link, and

2. The ways that the network links are connected together, or are connected to the outside world, can change.

Either of these occurrence would indicate the need to renumber one or more enterprise networks. It would be quite profitable to be able to renumber enterprise networks without requiring expensive downtime for the networks and or the nodes on the network.

Address shortages and routing hierarchy problems threaten the network operations of larger enterprises, but they also affect small sites -- even the home worker who dials in to the office via the Internet. Smaller networks can be completely dropped from Internet backbone route tables if they do not adapt to the address hierarchy, while larger networks may refuse to renumber and cause a larger routing problem for the backbone providers of the Internet. With today's IPv4 address registries, ISPs with individual dial-in clients cannot allocate IP numbers as freely as they wish. Consequently, many dial-in users must use an address allocated from a pool on a temporary basis. In other cases, small dial-in sites are forced to share a single IP address among multiple end systems.

A unique IP address sets the stage for users to gain direct connectivity to other users on the Internet, as determined by local policy. It also simplifies a wide range of productive interactive applications, of which telecommuting and remote diagnostics are only two examples. Today's hierarchy of limited and poorly allocated IPv4 addresses has already caused problems, and will continue to do so as more and more devices of varying capabilities are added to the Internet.

2.2.4. Security

Encryption, authentication, and data integrity safeguards are needed for enterprise internetworking and virtual private networks (VPNs). For these purposes, IPv6 offers security header extensions.

The IPv6 authentication extension header guarantees that a packet did indeed originate from the host indicated in its source address. This prevents malicious users from configuring an IP host to impersonate another, to gain access to secure resources. Such source-address masquerading (spoofing) is among the techniques that could be used to obtain valuable financial and corporate data, or could give adversaries of the enterprise control of servers for malicious purposes. Spoofing might fool a server into granting access to valuable data, passwords, or network control utilities. IP spoofing is known to be one of the most common forms of denial-of-service attack; with IPv4 it is typically impossible for a server to determine whether packets are being received from the legitimate end node. Some enterprises have responded by installing firewalls, but these devices introduce a number of new problems, including performance bottlenecks, restrictive network policies, and limited connectivity to the Internet or even between divisions of the same company.

IPv6 uses a standard method to determine the authenticity of packets received at the network layer, ensuring that network products from different vendors can use interoperable authentication services. IPv6 implementations are required to support the MD5 algorithm for authentication and integrity checking to insure that any two IPv6 nodes can interoperate securely. Since the specification is algorithm-independent, other techniques may be used as well.

Along with packet spoofing, another major hole in Internet security is the widespread deployment of traffic analyzers and network "sniffers" which can surreptitiously eavesdrop on network traffic. These generally helpful diagnostic devices can be misused by those seeking access to credit card and bank account numbers, passwords, trade secrets, and other valuable data. In IPv6 privacy (data confidentiality) is provided by a standard header extension for end-to-end encryption at the network layer. IPv6 encryption headers indicate which encryption keys to use, and carry other handshaking information. IPv4 network-layer extensions for this have been defined and are compatible with those for IPv6, but are not yet in wide use.

Both IPv6 security headers can be used directly between hosts or in conjunction with a specialized security gateway that adds an additional level of security with its own packet signing and encryption methods.

2.2.5. Mobility

IPv4 has difficulties managing mobile computers, for several reasons:

- A mobile computer needs to make use of a forwarding address at each new point of attachment to the Internet, and it's not always so easy to get such an address with IPv4

- Informing any agent in the routing infrastructure about the mobile node's new location requires good authentication facilities which are not commonly deployed in IPv4 nodes.

- In IPv4, it may be difficult for mobile nodes to determine whether or not they are attached to the same network.

- It is unlikely in IPv4 that mobile nodes would be able to inform their communication partners about any change in location.

Each of these problems is solved in a natural way by using features in IPv6. The benefits for mobile computing are apparent in quite a number of aspects of the IPv6 protocol design. The improvements in option processing for destination options, autoconfiguration, routing headers, encapsulation, security, and anycast addresses all contribute to the natural design of mobility for IPv6 [22]. In fact, some satellite work in Europe is already starting to become IPv6 based. The IPv6 mobility advantage may be further emphasized by combining flow label management to provide better Quality of Service to mobile nodes.

2.3. The IPv6 solution

IPv6, with its immensely larger address space, defines a multi-level hierarchical global routing architecture. Using CIDR-style prefixes [33], the IPv6 address space can be allocated in a way that facilitates route summarization, and controls expansion of route tables in backbone routers. The vastly greater availability of IPv6 addresses eliminates the need for private address spaces. ISPs will have enough addresses to allocate to smaller businesses and dial-in users that need globally unique addresses to fully exploit the Internet. Using an example from crowded telephone networks, one might say that IPv6 eliminates the need for "extensions", so that all offices have direct communication lines and do not need operators (automatic or otherwise) to redirect calls.

2.3.1. Address Autoconfiguration

Each IPv6 node initially creates a local IPv6 address for itself using "stateless" address autoconfiguration, not requiring a manually configured server. Stateless autoconfiguration further makes it possible for nodes to configure their own globally routable addresses in cooperation with a local IPv6 router. Typically, the node combines its 48 or 64 bit MAC (i.e., layer-2) address, assigned by the equipment manufacturer, with a network prefix it learns from a neighboring router. This keeps end user costs down by not requiring knowledgeable staff to properly configure each workstation before it can be deployed. These costs are currently part of the initial equipment expense for almost all IPv4 computing platforms. With the possibility of low or zero administrative costs, and the possibility of extremely low cost network interfaces, new market possibilities can be created for control of embedded computer systems. This feature will also help when residential networks emerge as an important market segment.

IPv4 networks often employ the Dynamic Host Configuration Protocol (DHCP) to reduce the effort associated with manually assigning addresses to end nodes. DHCP is termed a "stateful" address configuration tool because it maintains static tables that determine which addresses are assigned to newly connected network nodes. A new version of DHCP has been developed for IPv6 to provide similar stateful address assignment as may be desired by many network administrators. DHCPv6 [2, 30] also assists with efficient reconfiguration in addition to initial address configuration, by using multicast from the DHCP server to any desired population of clients.

The robust autoconfiguration capabilities of IPv6 will benefit internetwork users at many levels. When an enterprise is forced to renumber because of an ISP change, IPv6 autoconfiguration will allow hosts to be given new prefixes without manual reconfiguration of workstations or DHCP clients. This function also assists enterprises in keeping up with dynamic end-user populations. Autoconfiguration allows mobile computers to receive valid forwarding addresses automatically, no matter where they connect to the network.

2.3.2. IPv6 Header Format

IPv6 regularizes and enhances the basic header layout of the IP packet (see Figures 5,6 in section 3.1). In IPv6, some of the IPv4 header information was dropped or made optional. The simplified packet structure is expected to offset the bandwidth cost of the longer IPv6 address fields. The 16-byte (128-bit) IPv6 addresses are four times longer than the 4-byte IPv4 addresses, but as a result of the retooling, the total IPv6 header size is only twice as large; many processing aspects are substantially more efficient. Note that a number of other designs were considered, including variable length addresses; in the end, simplicity won out over infinite extensibility, partially because 128 bits offers such a huge total address space. Recent work [15] in IP header compression promises to reduce or perhaps even effectively eliminate any additional network load associated with the use of 128-bit addresses.

IPv6 encodes IP header options in a way that streamlines the forwarding process. Optional IPv6 header information is conveyed in independent "extension headers" located after the IPv6 header and before the transport-layer header in each packet. Most IPv6 extension headers are not examined or processed by intermediate nodes (in contrast with IPv4). This enables a big improvement in the deployability of optional IPv6 features, compared to IPv4 where IP options typically cause a major performance loss for the packet at every intermediate router. IPv6 header extensions are variable in length and can contain more information than before. Network protocol designers can introduce new header options in a straightforward manner. More details about the comparisons between the IPv4 and IPv6 headers are discussion in section 3.1.

So far, option fields have been specified for carrying explicit routing information created by the source node, as well as for mobility, authentication, encryption, and fragmentation control. At the application level, header extensions are available for specialized end-to-end network applications that require their own header fields within the IP packet.

2.3.3. Multicast

Modern internetworks need to transmit streams of video, audio, animated graphics, news, financial, or other timely data to groups of functionally related but dispersed endstations. This is best achieved by network layer multicast. Typically, a server sends out a single stream of multimedia or time-sensitive data to be received by subscribers. A multicast-capable network routes the server's packets to each subscriber in the multicast group using an efficient path (see Figure 2), replicating only as needed. In the figure, a single

```
                             Multicast Source
                                  +---+
                                  |   |
                                  |   |
                                  +-+-+
                                    |
                                    |
                                    |
      ---+------+----+----------+---+----+-----+--------+------+-----+-
         |      |    |          |        |     |        |      |     |
         |      |    |          |        |     |        |      |     |
         |      |    |          |        |     |        |      |     |
         |    +-+-+  |          |      +-+-+   |        |      |     |
         |    |   |  |          |      |   |   |      +-+-+    |     |
         |    |   |  |          |      |   |   |      |   |    |   +-+-+
       +-+-+  +---+  |        +-+-+    +---+   |      |   |    |   |   |
       |   |         |        |   |            |      +---+    |   |   |
       |   |       +-+-+      |   |          +-+-+  Multicast  |   +---+
       +---+       |   |      +---+          |   |    Group    |
     Multicast     |   |    Multicast        |   |    Member +-+-+
       Group       +---+      Group          +---+           |   |
       Member                 Member                         |   |
                                                             +---+
                                                           Multicast
                                                             Group
                                                             Member
```

Figure 2: Multicast in Action

packet from the source will be received by all the multicast group members. When there are multiple networks containing multicast group members, a packet distribution "tree" is created for the multicast group. Routers use multicast protocols such as DVMRP (Distance Vector Multicast Routing Protocol) [13] and PIM (Protocol Independent Multicast) [10] or MOSPF (Multicast Open Shortest Path First) [26] to dynamically construct the packet distribution tree that connects all members of a group with the multicast server. Only members that have joined the multicast group perform the processing to receive the data. A new member becomes part of a multicast group by sending a "join" message to a nearby router. The distribution tree is then adjusted to include the new route. Servers can then multicast a single packet, and it will be replicated as needed and forwarded through the internetwork to the multicast group. This conserves both server and network resources and, hence, is superior to unicast and broadcast solutions. Multicast applications have been developed for IPv4, but IPv6 extends IP multicasting capabilities by defining a much larger multicast address space. All IPv6 routers are required to support multicast. In fact, in IPv6 broadcast is viewed as a special case of multicasting.

2.3.4. Anycast

```
      -----  -----  -----
      | X |  | Y |  | Z |
      -----  -----  -----
         \    |     /               ------- ISP transit domain ---------
          \   |    /                |                                  |
           -------                  |              -------             |
           | rtr |---------------------------------| rtr |             |
           -------                  |              -------             |
          /       \                 |             /       \            |
         /         \                |            /         \           |
  -------           -------         |     -------           -------    |
  | rtr | Enterprise| rtr |---------------| rtr |  Anycast  | rtr |    |
  -------  Network  -------         |     -------   Group   -------    |
         \         /                |            \         /           |
          \       /                 |             \       /            |
           -------                  |              -------             |
           | rtr |---------------------------------| rtr |             |
           -------                  |              -------             |
              |                     |                                  |
            -----                   |                                  |
            | Q |                   ------- ISP transit domain ---------
            -----
```

Figure 3: Anycast Addressing

Anycast services, supported in the IPv6 specification, are not defined architecturally in IPv4. Conceptually, anycast is a cross between unicast and multicast: an arbitrary collection of nodes may be designated as an anycast group [29]. A packet addressed to the group's anycast address is delivered to only one of the nodes in the group, typically the node with the "nearest" interface in the group, according to current routing protocol metrics. This is in contrast with multicast services, which deliver packets to all members of the multicast group. Nodes in an anycast group are specially configured to recognize anycast addresses, which are drawn from the unicast address space [21].

Anycasting is a new service, and its applications have not been fully developed. Using anycast, an enterprise could forward packets to exactly one of the routers on its ISP's backbone (see Figure 3). If all of a provider's routers have the same anycast address, traffic from the enterprise will have several redundant access points to the Internet. And if one of the backbone routers goes down, the next nearest device automatically will receive the traffic.

In figure 3, suppose some hosts Q, X, Y, and Z in an Enterprise Network send data to the anycast address served by the backbone routers in the Anycast Group of the ISP Transit Domain. The border routers in the Enterprise Network forward the data just as they would for data sent to a unicast address. Then, any one of the backbone routers in the Anycast Group may receive the data, eliminating the overhead which would have been incurred if the backbone routers were instead configured to form a multicast group. If there are multiple home agents for mobile nodes on a single home network, they also join a anycast group. In that way, a mobile node can register with exactly one home agent even in the case when it doesn't know the address of any specific one.

Anycast is hoped to become an important method for allowing endstations to efficiently access well-known services, mirrored databases, Web sites, and message servers. It provides a versatile and cost-effective model for enabling application robustness and load balancing. For instance, anycast could provide enterprise robustness by assigning all the DNS servers in an enterprise the same anycast address.

2.3.5. Quality of Service

IPv4 carries a "differentiated services" byte and IPv6 carries an equivalent "traffic class" byte, intended for support of simple differentiated services. Both IPv4 and IPv6 can support the RSVP protocol for more complex quality of service implementations. Additionally, the IPv6 packet format contains a new 20-bit traffic-flow identification field that will be of great value to vendors who implement quality-of-service (QoS) network functions. Such QoS products are still in the planning stage, but IPv6 lays the foundation so that a wide range of QoS functions (including bandwidth reservation and delay bounds) may be made available in a open and interoperable manner.

2.3.6. The Transition to IPv6

The transition from IPv4 to IPv6 could take one of several paths. Some are lobbying for rapid adoption of IPv6 as soon as possible.

Others prefer to defer IPv6 deployment until the IPv4 address space is exhausted, or until other issues leave no other choice. Either way, given the millions of existing IPv4 network nodes, IPv4 and IPv6 will coexist for an extended period of time.

Therefore, IETF protocol designers have gone to great lengths to ensure that hosts and routers can be upgraded to IPv6 in a graceful, incremental manner. The transition will prevent isolation of IPv4 nodes, and also prevent "fork-lift" upgrades for entire user populations. Transition mechanisms have been engineered to allow network administrators flexibility in how and when they upgrade hosts and intermediate nodes. IPv6 can be deployed in hosts first, in routers first, or, alternatively, in a limited number of adjacent or remote hosts and routers. The nodes that are upgraded initially do not have to be colocated in the same local area network or campus.

Many upgraded hosts and routers will need to retain downward compatibility with IPv4 devices for an extended time period (possibly years or even indefinitely). It was also assumed that upgraded devices should have the option of retaining their IPv4 addresses. To accomplish these goals, IPv6 transition relies on several special functions that have been specified by the ``ngtrans'' working group of the IETF, including dual-stack hosts, routers, and tunneling IPv6 via IPv4.

2.3.7. IPv6 DNS

Domain Name Service (DNS) is something that administrators must consider before deploying IPv6 or dual-stack hosts. The current 32-bit name servers cannot handle name-resolution requests for 128-bit addresses used by IPv6 devices. In response to this issue, IETF designers have defined "DNS Extensions to Support IP Version 6" [35]. This specification creates a new "AAAA" (quad A) DNS record type that will map domain names to an IPv6 address. Domain name lookups (reverse lookups) based on 128-bit addresses also are defined. Once an IPv6-capable DNS is in place, dual-stack hosts can interact interchangeably with IPv6 nodes. If a dual-stack host queries DNS and receives back a 32-bit address, IPv4 is used; if a 128-bit address is received, then IPv6 is used. Where the DNS has not been upgraded to IPv6, hosts can resolve name-to-IPv6-address mappings through the use of manually configured local name tables.

IPv6 autoconfiguration and IPv6 DNS can be linked by using dynamic DNS updates, coupled with secure DNS. By these means DNS servers can be securely and automatically updated whenever an IPv6 node acquires a new address, enabling an additional measure of convenience compared with renumbering in IPv4 today.

2.3.8. Application Modification for IPv6

Applications that do not directly access network functions (i.e. do not call a socket or DNS API and do not handle numeric IP addresses in any way) need no modifications to run in the dual-stack environment. Applications that use certain interface APIs to communicate with the network stack will require updating before using IPv6. For example, applications that access DNS or use sockets must be enhanced with the capability to handle AAAA records and 128-bit addresses. Applications which are expected to run both IPv4 and IPv6, as well as using IPv6 security, quality of service, and other features, will need more extensive updating.

Adding such a dual-stack architecture to all the existing hosts is, in fact, a significant effort. This effort has to be balanced against the benefits of IPv6, and against the effort to renumber the existing hosts if the network deployment grows past the restrictions resulting from insufficient address space.

2.3.9. Routing in IPv6/IPv4 Networks

Routers running both IPv6 and IPv4 can be administered in much the same fashion that IPv4-only networks are currently administered. IPv6 versions of popular routing protocols, such as Open Shortest Path First (OSPF) and Routing Information Protocol (RIP), are already running. Administrators may choose to keep the IPv6 topology logically separate from the IPv4 network, even though both run on the same physical infrastructure, allowing the two to be administered separately. Alternatively, it may be advantageous to align the two architectures by using the same domain boundaries, areas, and subnet organization. Both approaches have their advantages. A separate IPv6 architecture can be used to replace the inefficient IPv4 topologies burdening many of today's enterprises. An independent IPv6 architecture presents the opportunity to build a fresh, hierarchical network address plan that will facilitate connection to one or more ISPs. This simplifies renumbering, route aggregation (summarization), and other goals of a routing hierarchy.

Initially, many IPv6 hosts may have direct connectivity to each other only via IPv4 routers. Such hosts will exist in islands of IPv6 topology surrounded by an ocean of IPv4. So, there are transition mechanisms that allow IPv6 hosts to communicate over intervening IPv4 networks. The essential technique of these mechanisms is IPv6 over IPv4 tunneling, which carries IPv6 packets within IPv4 packets (see Figure 4). Tunneling allows early IPv6 implementations to take advantage of existing IPv4 infrastructure without any change to IPv4 components. A dual-stack router or host on the "edge" of the IPv6 topology simply inserts an IPv4 header in front of ("encapsulates")

```
                     +--------------------+
 +-----------+       |   IPv4 Network     |    +-----------+
 | Dual-stack|       |                    |    | Dual-stack|
 | IPv4/IPv6 ========tunnel through======== IPv4/IPv6 |
 | router    |       |                    |    | router    |
 +-----------+       |                    |    +-----------+
     / | \           +--------------------+        / | \
    /  |  \                                       /  |  \
   /   |   \                                     /   |   \
+--+  +--+  +--+                             +--+  +--+  +--+
|  |  |  |  |  |                             |  |  |  |  |  |
+--+  +--+  +--+                             +--+  +--+  +--+
IPv6 endstations                             IPv6 endstations
```

Figure 4: IPv6 over IPv4 Tunneling

each IPv6 packet and sends it as native IPv4 traffic through existing links. IPv4 routers forward this traffic without knowledge that IPv6 is involved. On the other side of the tunnel, another dual-stack router or host "decapsulates" (removes the extra IP header from) the IPv6 packet and routes it to the ultimate destination using standard IPv6.

To accommodate different administrative needs, IPv6 transition mechanisms include two types of tunneling: automatic and configured. To build configured tunnels, administrators manually define IPv6-to-IPv4 address mappings at tunnel endpoints. Outside of the tunnel, traffic is forwarded with full 128-bit addresses. At the tunnel entry point, a manually configured router table entry dictates which IPv4 address is used to traverse the tunnel. This requires a certain amount of manual administration at the tunnel endpoints, but traffic is routed through the IPv4 topology dynamically, without the knowledge of IPv4 routers. The 128-bit addresses do not have to align with 32-bit addresses in any way.

Mbone deployment using IP-within-IP tunneling has been quite successful, and validates this design approach as well as supporting the likelihood of smooth transition.

2.3.10. The Dual-Stack Transition Method

Initial users of IPv6 machines will require continued interaction with existing IPv4 nodes. This is accomplished with the dual-stack IPv4/IPv6 approach. Many hosts and routers in today's multivendor, multiplatform networking environment already support multiple network stacks. For instance, the majority of routers in enterprise networks are multiprotocol routers. Many workstations run some combination of IPv4, IPX, AppleTalk, NetBIOS, SNA, DECnet, or other protocols. The inclusion of one additional protocol (IPv6) on an endstation or router is a well-understood problem. When running a dual IPv4/IPv6 stack, a host has access to both IPv4 and IPv6 resources. Routers running both protocols can forward traffic for both IPv4 and IPv6 end nodes.

Dual-stack machines can use totally independent IPv4 and IPv6 addresses, or they can be configured with an IPv6 address that is IPv4-compatible. Dual-stack nodes can use conventional IPv4 autoconfiguration services (DHCP) to obtain their IPv4 addresses. IPv6 addresses can be manually configured in the 128-bit local host tables, or preferably obtained via IPv6 autoconfiguration mechanisms. Major servers will run in dual-stack mode until all active nodes are converted to IPv6.

2.3.11. Automatic Tunneling

Automatic tunnels use "IPv4-compatible" addresses, which are hybrid IPv4/IPv6 addresses. A compatible address is created by adding leading zeros to a 32-bit IPv4 address to pad it out to 128 bits. When traffic is forwarded with a compatible address, the device at the tunnel entry point can automatically address encapsulated traffic by simply converting the IPv4-compatible 128-bit address to a 32-bit IPv4 address. On the other side of the tunnel, the IPv4 header is removed to reveal the original IPv6 address. Automatic tunneling allows IPv6 hosts to dynamically exploit IPv4 networks, but it does require the use of IPv4-compatible addresses, which do not bring the benefits of the 128-bit address space.

IPv6 nodes using IPv4-compatible addresses cannot take advantage of the extended address space, but they can exploit the other IPv6 enhancements, including flow labels, authentication, encryption, multicast, and anycast. Once a node is migrated to IPv6 with IPv4 compatibility, the door is open for a fairly painless move to the full IPv6 address space. IPv4-compatible addressing means that administrators can add IPv6 nodes while initially preserving their basic address and subnet architecture. Automatic tunnels are available when needed, but they may not be necessary when major backbone routers are upgraded to include the IPv6 stack. Upgrades can be achieved quickly and efficiently when backbone routers support full remote configuration and upgrade capabilities.

3. Part II: The Technical Case for IPv6

In this section, the technical aspects of IPv6 are discussed. In many cases, the technical details illustrate the concepts of the previous section. Other features are introduced as needed to help provide a fuller understanding of the protocol.

3.1. IPv6 Headers vs. IPv4 Headers

To start the technical look at IPv6, we compare the IPv6 header with the IPv4 header. Both headers carry version numbers and source/destination addresses, but as Figure 6 shows, the IPv6 header is considerably simplified, which makes for more efficient processing by routing nodes. Whereas IPv4 headers are variable in length, IPv6 headers have a fixed length of 40 bytes. This allows router software designers to optimize the parsing of IPv6 headers along fixed boundaries. Additional processing efficiencies have been realized by reducing the number of required header fields in IPv6. The classic IPv4 header contains 14 fields, whereas IPv6 only uses 8 fields.

```
+--------+-------+----------------+--------------------------------+
|Version| 4 bits|    8 bits      |         16 bits                |
| == 4  |  IHL  |Type of Service|       Total Length             |
+--------+-------+----------------+--------------------------------+
|          16 bits                | 4 bits|       12 bits          |
|        Identification           | Flags |    Fragment Offset     |
+---------------------------------+--------------------------------+
|     8 bits     |     8 bits     |         16 bits                |
| Time to Live   |    Protocol    |       Header Checksum          |
+---------------------------------+--------------------------------+
|                              32 bits                             |
|                           Source Address                         |
+------------------------------------------------------------------+
|                              32 bits                             |
|                        Destination Address                       |
+------------------------------------------------------------------+
```

Figure 5: IPv4 Header Format

One of the first IPv4 components to be discarded was the header length field, which is clearly no longer required due to the fixed header length of all IPv6 packets. The total length field of IPv4 has been retained in the guise of the IPv6 payload length field. But this field does not include the length of the IPv6 header, which is always assumed to be 40 bytes. The new payload length field can

```
+--------+---------------+----------------------------------------+
|Version|    8 bits     |             20 bits                     |
| == 6  | Traffic Class |            Flow Label                   |
+--------+---------------+-------+---------------+----------------+
|           16 bits              |    8 bits     |    8 bits      |
|        Payload Length          |  Next Header  |   Hop Limit    |
+--------------------------------+---------------+----------------+
|                            128 bits                             |
|                                                                 |
|                         Source Address                          |
+-----------------------------------------------------------------+
|                            128 bits                             |
|                                                                 |
|                       Destination Address                       |
+-----------------------------------------------------------------+
```

Figure 6: IPv6 Header Format

accommodate packets up to 64 KB in length. Even larger packets, called "jumbograms", can be passed between IPv6 nodes if the payload length field is set to zero and a special extension header is added, as discussed below.

The time-to-live (TTL) field of IPv4 has been renamed the IPv6 ``hop limit'' field, to describe more accurately its actual function. The field is used by routers to detect and break loops, by decrementing a maximum hop value by 1 for each hop of the end-to-end route. The hop-limit field is set to the appropriate value by the source node. When the value in the hop limit field is decremented to zero, the packet is discarded. The IPv6 hop-count field allows up to 255 hops, which exceeds the needs for even the largest of networks, as best we can calculate today.

In addition to the header length field, a number of basic IPv4 fields were eliminated from the IPv6 header: fragment offset, identification, flags, checksum. The IPv4 type-of-service field is replaced by the IPv4 traffic class field, plus the all-new flow label field. The IPv4 fragmentation fields (offset, identification, and flags) have been moved to optional headers in IPv6, as discussed in section 3.6. Finally, the IPv4 checksum field has been abandoned in IPv6, since error checking typically is duplicated at other levels of the protocol stack. Bad packets will be detected below, at the link-layer, or above, at the transport layer. Requiring routers to perform error checking has caused reduced performance in today's Internet.

3.2. Extension Headers

IPv4 headers include an options field, which conveys information about security, source routing, and other optional parameters. Unfortunately, options are poorly utilized because routers typically offer degraded performance to packets that contained options.

The IPv4 options field has been replaced in IPv6 by extension headers that are located after the primary IPv6 header and before the transport header and application payload. IPv6 extension headers provide security, fragmentation, source routing, and other functions. There is no set limit on the number of extension headers between the initial header and the higher layer payload. Since IPv6 separates options into modular headers, processing should be simpler and thus can remain on the fast path as needed. Figure 7 shows encryption and fragmentation headers occurring after the primary IPv6 header and before the Transmission Control Protocol (TCP) header.

```
+----------+----------------+-------------------+-----------------
| IPv6 Hdr | Encryption Hdr | Fragmentation Hdr | Transport, etc
+----------+----------------+-------------------+-----------------
```

Figure 7: IPv6 Extension Headers

The protocol type field (e.g., TCP or User Datagram Protocol (UDP)), is no longer needed, since each header field indicates the type of the next header, which can be a TCP/UDP header, or another IPv6 extension header. IETF working groups have already defined a number of extension headers for IPv6 and have suggested guidelines for the order of header insertion. The suggested order for extension headers, if any are present, is as follows:

- (Primary IPv6 header)
- Hop-by-Hop options header
- Destination options header-1
- Source Routing header
- Fragmentation header
- Authentication header
- IPv6 Encryption header
- Destination options header-2

followed by the upper layer headers and payload.

Each extension header typically occurs only once within a given packet, except for the destination options header (as explained in Section 3.4).

3.3. Hop-by-Hop Options Header

When present, this header carries options that are examined by intermediate nodes along the forwarding path. It must be the first extension header after the initial IPv6 header. Since this header is read by all routers along the path, it is useful for transmitting management information or debugging commands to routers. One currently defined application of the hop-by-hop extension header is the Router Alert option, which informs routers that the packet should be processed completely by a router before it is forwarded to the next hop. An example of such a packet is an RSVP [3] resource reservation message for QoS.

3.4. Destination Options Headers

There are two variations of this header, each with a different position in the packet. The first incidence of this field is for carrying information to the first destination listed in the IPv6 address field. This header can also be read by a subsequent destination listed in the source routing header address fields. The second incidence of this header is used for optional information that is only to be read by the final destination. For efficiency, the first variation is typically located towards the front of the header chain, directly after the hop-by-hop header (if any). The second variation is relegated to a position at the end of the extension header chain, which is typically the last IPv6 optional header before transport and payload.

3.5. Source Routing Header

The IPv6 routing extension header subsumes the loose and strict source routing functions supported currently by IPv4. This optional header allows a source node to specify a list of IP addresses that determine which routing path a packet will traverse. IPv6, in [12], defines a "Type 0" (zero) routing header, which gives a sending node a great deal of control over each packet's route. Type 0 routing headers contain a 24-bit field that indicates how intermediate nodes may forward a packet to the next address in the routing header. This extended variety of routing header should provide sufficient routing flexibility for many future routing applications, for applications that need better routing control than is available today.

IPv6's loose source routing (LSR) (analogous to IPv4's LSR option) is illustrated in Figure 8. In "loose" forwarding, unlisted routers can be visited by a packet. So, for example in figure 8 the packet could be routed from router 3 through router 4 and then to router 5, even though router 4 was not specified in the routing information field of the routing header. If, instead, "strict" source routing were selected, then the packet would have to be dropped after it arrived at router 3, since router 3 does not have a direct connection to router 5. The source routing feature works in conjunction with another routing header field that contains a value equal to the total number of segments remaining in the source route. Each time a hop is made, this "segments left" field is decremented.

IPv6 corrects another deficiency in the specification of IPv4 source routing options, by relaxing the requirement that destination nodes reverse the source route for transmitting packets back to the node originating the source route. This requirement is among the reasons that IPv4 source routing has almost entirely fallen out of use, because it opens up a big security hole. If a source route were to be reversed, without being sure that the source route was in fact originated by the indicated source node, then any other node within the Internet could easily masquerade as that indicated source node. IPv6 source routes, on the other hand, do not carry with them the same security exposure, since the recipient of such a routing header is not required to use the information for sending packets back to the source.

```
                     IPv6 Packet
   +----------+-----+-------------------------------+- -- -- -- -- --
   | IPv6 Hdr | ... | Route Information: 1, 2, 3, 5 |  ...
   +----------+-----+-------------------------------+- -- -- -- -- --


     +---+
     | X |                 +--------+            +--------+     +---+
     +---+              ---| rtr 4 |------------| rtr 5 |------| Y |
         \             /   +--------+            +--------+     +---+
          \           /           \
           +--------+              \   +--------+
           | rtr 1 |               \--| rtr 3 |
           +--------+                  +--------+
                     \                /
                      \              /
                       +--------+   /
                       | rtr 2 |--
                       +--------+
```

Figure 8: Source Routing Extension Header

When Type 0 routing headers are used, the initial IPv6 header contains the destination addresses of the first router in the source route, not the final destination address. At each hop, the intermediate node replaces this destination address with the address of the next routing node, and the "segments left" field is decremented.

3.6. Fragmentation Header

IPv4 has the ability to fragment packets at any point in the path, depending on the transmission capabilities of the links involved. This feature has been dropped in IPv6 in favor of end-to-end fragmentation/reassembly, which is executed only by IPv6 source and destination nodes. Packet fragmentation is not permitted in intermediate IPv6 nodes. The elimination of the fragmentation field allows a simplified packet header design and better router performance for the great majority of cases where fragmentation is not required. Today's networks generally support frame sizes that are large enough to carry typical IP packets without fragmentation. In the event that fragmentation is required, IPv6 provides an optional extension header that is used by source nodes to divide packets into smaller units. If higher level protocols are using larger payloads, the source node can make use of the IPv6 fragmentation extension header to divide large packets into 1500-byte units for network transmission. The IPv6 destination node will reassemble these fragments in a manner that is transparent to upper layer protocols and applications.

The IPv6 fragmentation header contains fields that identify a group of fragments as a packet and assigns them sequence numbers. The source node is responsible for sizing packets correctly, so it has to determine the Maximum Transmission Unit (MTU) of the links in the end-to-end path. For instance, if two FDDI networks with 4500-byte MTUs are connected by an Ethernet with an MTU of 1500, then the source node must send packets that are no larger than 1500.

End nodes can determine the smallest MTU of a path with the MTU path discovery process [25]. Typically, with this technique, the source node probes the MTU by transmitting a packet with an MTU as large as the local interface can handle (see Figure 9). If this MTU is too large for some link along the path, an ICMP "Datagram too big" message will be sent back to the source. This message will contain a packet-too-big indicator and the MTU of the affected link. The source can then adjust the packet size downward (fragment) and retransmit another packet. This process is repeated until a packet gets all the way to the destination node. The discovered MTU is then used for fragmentation purposes. Although source-based fragmentation is fully supported in IPv6, it is recommended that

```
     +--ICMP Datagram Too Big--<--+
     v                            |
   +---+  FDDI  +-----+  FDDI  +-----+   Ethernet   +-----+  FDDI  +---+
   | X |--------| rtr |--------| rtr |--------------| rtr |--------| Y |
   +---+        +-----+        +-----+  MTU = 1500  +-----+        +---+
     |                            |
     +-->-MTU Discovery Message->-+
```

Figure 9: MTU Discovery Process

network applications adjust packet size to accommodate the smallest MTU of the path. This will avoid the overhead associated with fragmentation/reassembly on source and destination nodes.

3.7. IPv6 Security

IPv6 has two security extension headers, one that enables the authentication of IP traffic for security purposes, and another that fully or partially encrypts IP packets. Implementation of security at the IP level can benefit "security aware" applications, as well as "security ignorant" applications that don't take explicit advantage of security features.

3.8. IPv6 Authentication Header

With IPv6 authentication headers, hosts establish a standards-based security association that is based on the exchange of algorithm-independent secret keys (e.g., MD5 [23]). In a client/server session, for instance, both the client and the server need to have knowledge of the key. Before each packet is sent, IPv6 authentication creates a checksum based on the key combined with the entire contents of the packet. This checksum is then re-run on the receiving side and compared. This approach provides authentication of the sender and guarantees that data within the packet has not been modified or replayed by an intervening party. Authentication can take place between clients, or clients and servers on the corporate backbone. It can also be deployed between remote nodes and corporate dial-in servers to ensure that the perimeter of the corporate security is not breached.

3.9. IPv6 Encryption Header

```
<-------------- Unencrypted ---------------> <----- Encrypted ----...
+-------------+----------------+------------+-----------------------
| IPv6 Header | Extension Hdrs | ESP Header | Transport Hdr & Payload
+-------------+----------------+------------+-----------------------
```

Figure 10: Transport Mode of IPv6 Encryption

```
<-----Unencrypted--------> <--------- Encrypted ----------------...
+--------+--------+-------+--------+--------+-------+---------------
|IPv6 Hdr|Ext.Hdrs|ESP Hdr|IPv6 Hdr|Ext.Hdrs|ESP Hdr|Transpt/Payload
+--------+--------+-------+--------+--------+-------+---------------
<-Encapsulating Headers--> <--------- Original Packet -------.......
```

Figure 11: Tunnel Mode of IPv6 Encryption

Authentication headers eliminate a number of host spoofing and packet modification attacks, but they do not prevent passively reading of data traversing the Internet and corporate backbone networks. This protection is offered by the Encapsulating Security Payload (ESP) service of IPv6 -- another optional extension header. Packets protected by the ESP encryption techniques can have very high levels of privacy and integrity -- something that is not widely available with the current Internet, except with certain secure applications (e.g., private electronic mail and secure HTTP Web servers). ESP provides encryption at the network layer, making it available to all applications in a standardized fashion.

IPv6 ESP is used to encrypt the transport-layer header and payload (e.g., TCP, UDP), or the entire IP datagram. Both these methods are accomplished with an ESP extension header that carries encryption parameters end-to-end. When just the transport payload is to be encrypted, the ESP header is inserted in the packet directly before the TCP or other transport header. In this case, the headers before the ESP header are not encrypted and the headers and payload after the ESP header are encrypted. This is referred to as "transport-mode" encryption, and is illustrated in figure 10. If it is desirable to encrypt the entire IP datagram, a new IPv6 and an ESP header are wrapped around all the fields (including the initial address fields) of the packet. Full datagram encryption is sometimes called "tunnel-mode" encryption because the payload of the datagram is unintelligible except at the endpoints of the security tunnel (see Figure 11).

Fully encrypted datagrams are somewhat more secure than transport mode encryption because the headers of the fully encrypted packet are not available for traffic analysis.

For instance, full tunnel-mode encryption allows the addresses contained in IPv6 source routing headers to be hidden from packet sniffing devices for the public portion of a path. There is a considerable performance penalty for full encryption, due to the overhead and processing cost of adding an additional IPv6 header to each datagram. In spite of its cost, full ESP encryption is particularly valuable to create a security tunnel (steel pipe) between the firewalls of two remote sites (see Figure 12). The full datagram encryption in the tunnel ensures that the various headers and address fields of encrypted packets will not be visible as traffic traverses the public Internet. Within the tunnel, only the temporary encapsulating address header is visible. Once through the tunnel and safely within a firewall, the leading ESP headers are stripped off and the packet is again visible, including any source routing headers required to finish the path.

```
                      ~~                            ~~
                      F~                            ~F
    +--------+        i~   +--------------------+   ~i       +--------+
    |        |        r~   |                    |   ~r       |        |
    | Site 1 |        e~   |   Public Internet  |   ~e       | Site 2 |
    |        |   ----------------------------------------    |        |
    |   <-------( - - - - - - ESP Steel Pipe - - - - - -()<-----<--   |
    |        |   ----------------------------------------    |        |
    |        |        w~   |                    |   ~w       |        |
    |        |        a~   |                    |   ~a       |        |
    |        |        l~   +--------------------+   ~l       |        |
    +--------+        l~                            ~l       +--------+
                      ~~                            ~~
```

Figure 12: Firewalls and Steel Pipe

The encryption and authentication services of IPv6 together create the security solution often needed by business and military applications. In some cases an authentication header will be carried inside an encrypted datagram, providing an additional layer of data integrity and verification of the sender's identification. In other cases, the authentication header may be placed in front of the encrypted transport-mode portion of the packet. This approach is desirable when the authentication takes place before decryption on the receiving end, which is the logical order in many cases. Taken together, the authentication and encryption services of IPv6 provide a robust, standards-based security mechanism that will play a decisive role in the continuing expansion of commerce and corporate operations onto IP-based network fabrics.

3.10. The IPv6 Address Architecture

Much of the discussion of IPv4 versus IPv6 focuses on the relative size of the address fields of the two protocols (32 bits versus 128 bits). But an equally important difference is the relative abilities of IPv6 and IPv4 to provide a hierarchical address space that facilitates efficient routing architectures. IPv4 was initially designed with class A, class B, and class C addresses, which divided address bits between network and host but did not create a hierarchy that would allow a single high-level address to represent many lower-level addresses. Hierarchical address systems work in much the same way as telephony country codes or area codes, which allow long-haul phone switches to route calls efficiently to the correct country or region using only a portion of the full phone number.

As the Internet grew, the non-hierarchical nature of the original IPv4 address space proved inadequate. This problem has been improved by use of CIDR 2.2.1, but legacy address assignments still hamper routing within the Internet. These legacy assignments limit both local and global levels of internetworking. To combat IPv4 deficiencies at the local area network level, the subnetting technique has been developed to create a more manageable division of large networks. Using subnets, a single network address can stand for a number of physical networks, a technique that conserves address space considerably. For example, a single Class B address can be used to access hundreds of physical networks, each of which itself could have dozens or hundreds of individual hosts.

At the level of large internet backbones and global routing, IPv4 addresses can be more efficiently aggregated with supernetting, a form of hierarchical addressing. With supernetting, backbone routers store a single address that represents the path to a number of lower level networks. This can considerably reduce the size of routing tables in backbone routers, which increases backbone performance and lowers the amount of memory and number of route processors required. Subnetting and supernetting have been particularly useful in extending the viability of the IPv4 Class C addresses. Both of these techniques are made possible by associating addresses stored in routers to bit masks that indicate which bits in an address are valid at the various levels of the hierarchy.

The process of creating an IPv4 routing hierarchy was formalized in CIDR, as discussed in Section 2.2.1. For instance, CIDR allows a number of (plentiful) Class C addresses to be summarized by a single prefix address, allowing Class C addresses to function in a similar way to hard-to-get Class A and Class B addresses. CIDR has extended the life of IPv4 and helped the Internet scale to its current size, but it has not been implemented in a consistent way across the Internet and enterprise networks. Consequently, the route table efficiencies and address space conservation advantages of CIDR are not today fully realized, nor will they ever be fully realized, due to the legacy nature of IPv4 networks and the difficulty of restructuring them. IPv4 will continue to waste its address space, and to burden routers with inefficient routes and excessively large routing tables.

At the departmental and workgroup level of internetworking, IPv4 engenders a high administrative workload associated with maintaining subnet bit masks and host addresses within the subnet structure, particularly where there are large, dynamic populations of end users. When an end user is moved in the subnetting environment, careful attention must be paid to ensure that the host renumbering process does not disrupt the ability of the user to make effective use of the network. The complexities and pitfalls of current subnetting methods can eventually make IPv4 less than viable in large organizations that experience growth of internetwork user populations (especially at current rates of growth).

3.11. The IPv6 Address Hierarchy

Motivated by the experience gained from IPv4, IPv6 designers made sure from the very beginning to provide a scalable address space that can be partitioned into a efficient global routing hierarchy. At the top of this hierarchy, several international registries assign blocks of addresses to top level aggregators (TLA). TLAs allocate blocks of addresses to Next Level Aggregators (NLA), which represent large providers and global corporate networks. When an NLA is a provider, it further allocates its addresses to its subscribers. Routing is efficient because NLAs that are under the same TLA will have addresses with a common TLA prefix. Subscribers with the same provider have IP addresses with an NLA common prefix. See Figure 13 for an example of Aggregation-based Allocation Structures. Although a number of allocation schemes are possible within IPv6's huge address space, an aggregation-based hierarchy is favored by IETF designers because it allows a choice between various allocation approaches. Provider allocation divides the hierarchy along lines of large service providers, regardless of their location. Geographic allocation divides the hierarchy strictly on the basis of the location of providers/subscribers (as does the telephony system

```
   +-----------+                          +-----------+
   | Long-Haul | - - - - - - - - - - - - -| Long-Haul |
   |  Provider |                          |  Provider |
   +-----------+                          +-----------+
     |   \                                      /
          \--------\        /------------------/
     |          +---------------+
                | Interexchange | - - - - - - ---> To other
     |          |     (TLA)     |                    interexchanges
                +---------------+
     |    /--------/   |    \  \---------------\
     |   /             |     \                  \
   +-----------+  +---------+ \           +-----------+
   | Long-Haul |  |Provider|  \          | Long-Haul |
   |  Provider |  +---------+   |         |  Provider |
   +-----------+          |     |         +-----------+
       |                  |     |              |    |
     +----------+         |     |    +----------+  +----------+
     |Subscriber|         |     |    |Subscriber|  | Provider |
     +----------+         |      \   +----------+  +----------+
                   +----------+   \                      |
                   |Subscriber|    \                     |
                   +----------+   +----------+   +----------+
                                  |Subscriber|   |Subscriber|
                                  +----------+   +----------+
```

Figure 13: Aggregation-based Allocation Structures

of country and area codes). Both of these approaches have their drawbacks because large backbone networks often don't conform strictly to geographic or provider boundaries. Some large networks, for instance, may connect to several ISPs; many large networks span numerous countries and geographical regions.

Aggregation-based allocation is based on the existence today of a limited number of high-level exchange points, where large long-haul service providers and telephone networks interconnect. The use of these exchange points to divide the IPv6 address hierarchy has a geographical component because exchanges are distributed around the globe. It also has a provider orientation because all large providers are represented at one or more exchange points.

As shown in Figure 14, the first 3 address bits indicate what type of address follows (unicast, multicast, etc.). The next 13 bits are allocated to the various TLAs around the world. Eight bits are

```
+--------+---------+-----------+-----------+---------------------+
| 3 bits | 13 bits |  32 bits  |  16 bits  |      64 bits        |
|  001   |   TLA   |    NLA    |    SLA    |    Interface ID     |
+--------+---------+-----------+-----------+---------------------+
<------ Public Topology ------> <- Site --> <--Local Interface-->
```

Figure 14: Aggregation-based IPv6 Addresses

reserved for future use, and the following 24 bits are allocated to the next lower level of providers and subscribers.

Next level aggregators can divide the NLA address field to create their own hierarchy, one that maps well to the current ISP industry, in which smaller ISPs subscribe to higher level ISPs, and so on. This is accomplished by the further subdivision of the 32-bit NLA field (see Figure 15). Following the NLA ID are fields for

```
<------------ 32 bits -----------> <--16 bits-> <---- 64 bits ---->
+-------+-------------------------+------------+-------------------+
| NLA 1 |          Site           |    SLA     |   Interface ID    |
+-------+-------------------------+------------+-------------------+
        +-------+-----------------+------------+-------------------+
        | NLA 2 |       Site      |    SLA     |   Interface ID    |
        +-------+-----------------+------------+-------------------+
                +-----------------+------------+-------------------+
                | NLA 3 |   Site  |    SLA     |   Interface ID    |
                +-----------------+------------+-------------------+
```

Figure 15: Subdividing the NLA Address Space

subscriber site networking information: Site Level Aggregator (SLA) and Interface ID. Typically, service providers supply subscribers with blocks of contiguous addresses, which are then used by individual organizations to create their own local address hierarchy and identify subnets and hosts. The 16-bit SLA field supports up to 65,535 individual subnets. The 64-bit Interface ID, which is used to identify an IPv6 interface on a network link, will typically be derived from the installed MAC address.

Internet backbone routers must maintain 40,000 or more routes. As the Internet continues to grow in size, IPv6's uniform application of hierarchical routing will likely be the only viable method for keeping the size of backbone router tables under control. With an aggregator-based address hierarchy, all of a subscriber's internal network segments can be reached through one or more high- level aggregation points. This allows backbone routers around the globe to efficiently summarize the routes to a customer's networks with high-level TLA address prefixes. Forwarding routes in the highest level backbones can be quickly calculated by looking only at the TLA portion of the address. IPv6's large hierarchical address space also allows a more decentralized approach to IP address allocation. Service providers can allocate addresses independently from central authorities, encouraging global network growth and eliminating bureaucratic bottlenecks in the growth process.

Aggregation-based addresses are just part of the total address space that has been defined for IPv6. Other address ranges have been assigned to multicasting and to nodes that only require unique addresses within a limited area (site-local and link-local addresses).

Site-local and link-local addresses are available for private, internal use by all enterprises, and are not allocated by public registry authorities. Site-local addresses enable networks to use non-unique local addresses that are later made globally unique by adding a prefix. This has an advantage: if an ISP changes, site local addresses can remain the same because they do not directly connect to the outside world. Link local addresses operate only over a single link, and can be used for temporary "bootstrapping" of network nodes before they receive a globally unique address (more on this in section 3.12).

3.12. Host Address Autoconfiguration

IPv6 has a large enough address architecture [19] to accommodate Internet expansion for many decades to come. Furthermore, IPv6 hosts can have their addresses automatically configured and reconfigured in a cost-effective and manageable way. Automatic address configuration is necessary in hierarchical routing because it supports scalable (and thus cost-effective) numbering and renumbering of large populations of IP hosts. Even a small renumbering cost, if incurred tens of thousands of times for every ISP connection, adds up to a major administrative headache. Conversely, scalable renumbering techniques will enable business enterprises to shop for the best connectivity solutions without worrying about the renumbering costs of reconnection to a new provider.

Autoconfiguration capabilities are important regardless of which style of address allocation is in effect. Occasionally, it may be necessary to renumber every host within an organization, as would be the case with a company that relocated its operations (with geographic addressing) or changed to another service provider (with provider-based addressing). Configuration of IP addresses is a fact of life at the workgroup and department levels of large networked organizations. IP addresses need to be configured for new hosts, for hosts that change location, and for hosts connected to physical networks that receive address modification (e.g., a new prefix). In addition to these traditional requirements for configuration, new requirements are emerging as large numbers of hosts become mobile. These requirements are basically not met in any meaningful way for use with the existing IPv4 installed base.

The process of autoconfiguration under IPv6 starts with the Neighbor Discovery (ND) protocol [28]. ND combines and refines the services provided in the IPv4 environment by Address Resolution Protocol (ARP) [31], Internet Control Message Protocol (ICMP) [32], and Router Advertisement [14]. Although it has a new name, ND is actually just a set of complementary ICMPv6 [8] messages that allow IPv6 nodes on the same link to discover link-layer addresses and to obtain and advertise various network parameters and reachability information. In a typical scenario, a host starts the process of autoconfiguration by creating a link-local address [37]. This address can be formed by adding a generic local address prefix to a unique token (typically the host's IEEE LAN interface address [20]). Once this address is formed, the host sends out an ND message to the address to ensure that it is unique. If no ICMP message comes back, the address is unique. If a message comes back indicating that the link-local address is already in use, then a different token is used (e.g., an administrative token or a randomly generated token).

Using the new link local address as a source address, the host then sends out an ND router solicitation request. The solicitation is sent out using the IPv6 multicast service. Unlike the broadcast ARPs of IPv4, IPv6 ND multicast solicitations are not necessarily processed by all nodes on the link, which can conserve processing resources in hosts. (IPv6 currently defines several permanent multicast groups for finding resources on the local node or link, including an all-routers group, an all-hosts group, and a DHCP server group). Routers respond to solicitation messages from hosts with a unicast router advertisement that contains, among other things, prefix information that indicates a valid range of addresses for the subnet. The ND message exchange is shown in Figure 16. Routers also send unsolicited advertisements periodically to local multicast groups.

The router advertisement message controls whether hosts use stateless or stateful autoconfiguration methods. In the case of stateful autoconfiguration, the host will contact a stateful address server, which will assign an address from a manually administered list.

```
              +---+                     +---+
              | Y |---------------------| Z |
              +---+                     +---+
               /                          \
          ----/                            \-----
         /                                       \
      +---+   ----- Router Solicitation ------> +------+
      | X |                                     | rtr |====To Internet
      +---+  <----- Router Advertisement -----  +------+
         \                                       /
          ----                              -----
              \                            /
               \                          /
              +---+                     +---+
              | W |---------------------| V |
              +---+                     +---+
```

Figure 16: Neighbor Discovery (ND) Router Message Exchange

DHCP [16] is the protocol of choice for autoconfiguration in IPv4 networks and has been reformulated for the IPv6 environment [2, 30].

With the stateless approach [37], a host can automatically configure its own IPv6 address without the help of a stateful address server or any human intervention. The host uses the globally valid address prefix information in the router advertisement message to create its own IPv6 address. This process involves the concatenation of a valid prefix with the host's link-layer address or a similar unique token. As long as the token is unique on the link and the prefix received from the router is correct, the newly configured IP address should provide reachability for the host extending to the entire enterprise and the Internet at large.

The advantages of stateless autoconfiguration are many. For instance, if an enterprise changes service providers, the prefix information from the new provider can be propagated to routers throughout the enterprise, and hence to all stateless autoconfiguring hosts. Hypothetically, if all hosts in the enterprise use IPv6 stateless autoconfiguration, the entire enterprise could be renumbered without the manual configuration of a single non-router host. At a more modest level, workgroups with substantial move/change activity also benefit from stateless autoconfiguration because hosts can receive a freshly configured and valid IP number each time they connect and reconnect to the network.

```
          +-------+
          | Home  |
          | Agent |\
          +-------+ \        +---------------------+
                     \       |                     |
                      ----------+                  |       +---+
                             |  |       /------------------| X |
                      ----------+ <----/           |       +---+
                     /       |                     |
                    /        +---------------------+
                   /
        +--------+/
        | Mobile |
        |  Node  |
        +--------+
```

Figure 17: Forwarding IP Traffic for Mobile IPv6 Nodes

Address autoconfiguration plays an essential role in the support for mobile nodes within IPv6. Each mobile node can configure an appropriate address, no matter which network it is attached to; it uses this address as a kind of forwarding address (or, as it is called, a "care-of address"). Then, the mobile node can receive all of its data from its home network by asking a router (called a "home agent") to forward packets to it at its care-of address. This process is illustrated in figure 17. Better yet, the mobile node can also instruct any other node (e.g., node 'X' in the figure) to forward data to its care-of address, so that the data never traverses the home network. Although not shown by the figure, the mobile node is identified by its home address, even though it is receiving packets sent to its care-of address. This is important so that the mobile node can maintain its connections even when it is wireless and undergoing handoff operations during continued operation of its network applications.

To facilitate dynamic host renumbering, IPv6 has a built-in mechanism to create a graceful transition from old to new addresses. Fundamental to this mechanism is the ability of IPv6 nodes to support multiple addresses per interface. IPv6 addresses assigned to an interface can be identified as valid, deprecated, or invalid. In the renumbering process, an interface's IPv6 address would become deprecated when a new address was automatically assigned (e.g., in the case of network renumbering). For a period of time after the new (valid) address is configured, the deprecated address continues to send and receive traffic. This allows sessions and communications based on the older address to be finished gracefully. Eventually the deprecated address becomes invalid and the valid address is used exclusively. Issuing multiple IP addresses allows renumbering to occur dynamically and transparently to end users and applications. Besides simplifying host renumbering, IPv6 has work underway to help with reconfiguring routers [9].

The above described stateless autoconfiguration process is particularly suited to conventional IP/LAN environments with 48-bit or 64-bit addressing [20] and native multicast services. Other network environments with different link characteristics may require modified or alternative configuration techniques. For instance, current ATM networks do not inherently support multicast services or IEEE MAC addresses, due to the use of virtual circuits and telephony-style calling numbers. Multicasting solutions for ATM are seen in the emerging Multicast Address Resolution Server (MARS) [34] that is being developed for IPv4 multicast over ATM. Plans are being devised to use MARS-style functionality to extend the IPv6 Neighbor Discovery protocol across ATM networks. This would allow network renumbering and stateless autoconfiguration to take place seamlessly in hybrid ATM/IPv6 fabrics.

3.13. Other Protocols and Services

The preceding discussion focuses on some of the more innovative and radical changes that IPv6 brings to internetworking. In many other areas, protocols and services will operate much the same as they do in the current IPv4 regime. As the industry moves to IPv6, PPP, DHCP and DNS servers are being modified to accommodate 128-bit addresses, but in terms of basic functionality, there will be little change. This is also generally true for interior and exterior routing protocols.

For example, OSPF is being updated with full support for IPv6 [6], allowing routers to be addressed with 128-bit addresses. The 32-bit link-state records of current OSFP will be replaced by 128-bit records. In general, the OSPF IPv6 link-state database of backbone routers will run in parallel with the database for IPv4 topologies. In this sense, the two versions of OSPF will operate as "ships in the night," just as the routing engines for IPv4, OSI and proprietary protocols may coexist in the same router without major interaction. Given the limited nature of the OSPF IPv6 upgrade, those engineers and administrators who are proficient in OSPF for IPv4 should have no problems adapting to the new version. An updated version of RIP is also available [24].

As with the interior gateway protocols, work is underway to create IPv6-compatible versions of the exterior gateway protocols that are used by routers to establish reachability across the Internet backbone between large enterprises, providers, and other autonomous systems. Today's backbone routers use the Border Gateway Protocol (BGP) to distribute CIDR-based routing information throughout the Internet. BGP is known by providers and enterprises and has a large installed base. Currently, work is underway to define BGP extensions to exchange reachability information based on the new IPv6 hierarchical address space.

4. Part III: Transition Scenarios

Part I of this paper provided an overview of the major transition mechanisms that are integral to the IPv6 design effort. These techniques include dual-stack IPv4 /IPv6 hosts and routers, tunneling of IPv6 via IPv4, and a number of IPv6 services, including IPv6 DNS, DHCP, MIBs, and so on. The flexibility and usefulness of the IPv6 transition mechanisms are best gauged through scenarios that address real-world networking requirements.

4.1. First Scenario: No Need to NAT

```
       --------------                              --------------
      /              \                            /              \
     |   Enterprise   |       +----------+       |   Enterprise   |
     |       A        |-------| IPv6 rtr |-------|       B        |
      \              /        +----------+        \              /
       --------------                              --------------
             ^                                           |
             |                                           |
             |                                           v
         +-------+                                   +-------+
         |IPv4 + |        IPv6 communication         |IPv4 + |
         |   IPv6|    - - - - - - - - - - - - - >    |   IPv6|
         | Host  |                                   | Host  |
         +-------+                                   +-------+
```

Figure 18: IPv6 Unites Private Address Spaces

Take, for instance, the case of two large, network-dependent organizations that must interface operations due to a merger and acquisition (M&A), or a new business partnership. Suppose both of the enterprises have large IPv4-based networks that have grown from small beginnings. Both of the original enterprises have a substantial number of private IPv4 addresses that are not necessarily unique within the current global IPv4 address space. Combining these two non-unique address spaces could require costly renumbering and restructuring of routers, host addresses, domains, areas, exterior routing protocols, and so on. This scenario is common in the current business climate, not only for Merger and Acquisition (M&A) projects, but also for large outsourcing and customer/supplier networking relationships, where many hosts from the parent, outsourcer, supplier, or partner must be integrated into one existing enterprise address structure. For these situations, IPv6 offers a convenient solution.

The task of logically merging two enterprise networks into a single autonomous domain can be expensive and disruptive. To avoid the cost and disruption of comprehensive renumbering, enterprises may be tempted to opt for the stopgap solution of a network address translator (NAT). In the M&A scenario, a NAT could allow the two enterprises to maintain their private addresses more or less unchanged. To accomplish this, a NAT must conduct address translation in real time for all packets that move between the two organizations. Unfortunately, this solution introduces all the problems associated with NATs that were discussed in Part I, section 2.2.2, including performance bottlenecks, lack of scalability, lack of standards, and lack of universal connectivity among all the nodes in the new enterprise and the Internet.

In contrast with NAT, IPv6 seamlessly integrates the two physical networks (see Figure 18). Suppose the two originally independent enterprises are known as Enterprise A and Enterprise B. The first step is to determine which hosts need access to both sides of the new organization. These hosts are outfitted with dual IPv4/IPv6 stacks, which allow them to maintain connectivity to their original IPv4 network while also participating in a new IPv6 logical network that will be created "on top" of the existing IPv4 physical infrastructure.

The accounting department of the combined enterprise will often have financial applications on servers that will need to be accessed by accounting employees in both Enterprise A and Enterprise B. Both servers and clients will run IPv6, but they will also retain their IPv4 stacks. The IPv6 sessions of the accounting department will traverse the existing local and remote links as "just another protocol," requiring no changes to the physical network. The only requirement for IPv6 connectivity is that routers that are adjacent to accounting department users must be upgraded to run IPv6. Where end-to-end IPv6 connectivity can't be achieved, one of the IPv4/IPv6 tunneling techniques can be employed.

As integration continues, other departments in the newly merged enterprises will also be given IPv4/IPv6 hosts. As new departments and workgroups are added, they may be given dual-stack hosts, or in some cases, IPv6-only hosts. Hosts that require communications to the outside world via the Internet will likely receive dual stacks to maintain compatibility with IPv4 nodes exterior to the enterprise. But in some cases, hosts that only require access to internal servers and specific outside partners may be able to achieve connectivity with IPv6-only hosts. A migration to IPv6 presents the opportunity for a fresh start in terms of address allocation and routing protocol structure. IPv6 hosts and routers can immediately take advantage of IPv6 features such as stateless autoconfiguration, encryption, authentication, and so on.

4.2. Second Scenario: IPv6 from the Edges to the Core

For corporate users, connectivity requirements typically focus primarily on access to local e-mail, WWW, database, and applications servers. In this case, it may be best to initially upgrade only isolated workgroups and departments to IPv6, with backbone router upgrades implemented at a slower rate. IPv6 protocol development is more complete for "edge" routing than for high-level backbone routing, so this is an excellent way for enterprises to gracefully transition into IPv6. As shown in Figure 19, independent workgroups can upgrade their clients and servers to dual-stack IPv4/IPv6 hosts or IPv6-only hosts. This creates "islands" of IPv6 functionality.

As enterprise-scale routing protocols such as OSPF and BGP for IPv6 mature, the core backbone IPv6 connections can be deployed. After the first few IPv6 routers are in place, it may be desirable to connect IPv6 islands together with router-to-router tunnels. In this case, one or more routers in each island would be configured as tunnel endpoints. As illustrated in Part I, in figure 4, when hosts use full IPv6 128-bit addressing, tunnels are manually configured so that the routers participating in tunnels know the address of the endpoints of the tunnel. With IPv4-compatible IPv6 addresses, automatic, nonconfigured tunneling is possible.

Routing protocols treat tunnels as a single IPv6 hop, even if the tunnel is comprised of many IPv4 hops across a number of different media. IPv6 routers running OSPF can propagate link-state reachability advertisements through tunnels, just as they would across conventional point-to-point links. In the IPv6 environment, OSPF can ensure that each tunnel is weighted properly within the topology. Routers generally make packet-forwarding decisions in the tunneling environment in the same way as in the IPv6-only network. The underlying IPv4 connections are essentially transparent to IPv6 routing protocols.

```
       IPv6 "Island"                             IPv6 "Island"
   --------------------                        --------------------
   |                  |                        |                  |
   | Dual Stack Hosts |                        | Dual Stack Hosts |
   |  +---+   +---+   |                        |  +---+   +---+   |
   |  |   |   |   |   |                        |  |   |   |   |   |
   |  +---+   +---+   |                        |  +---+   +---+   |
   |    |       |     |                        |    |       |     |
   |     \     /      |                        |     \     /      |
   |    +-------+     |                        |    +-------+     |
   |    | Dual  |     |                        |    | Dual  |     |
   |    | Stack |     |                        |    | Stack |     |
   |    | Router|     |                        |    | Router|     |
   |    +-------+     |                        |    +-------+     |
   |                  |                        |                  |
   --------------------                        --------------------
                  \                               /
                   \                             /
                 +------+                   +------+
   IPv4          | IPv4 |-------------------| IPv4 |       IPv4
     Hosts       |  rtr |                   |  rtr |         Hosts
   +---+         +------+       IPv4        +------+        +---+
   | X |-\         /  \    infrastructure     / \         /-| W |
   +---+  \       /    \-------\    /--------/   \       /  +---+
           \     /              \  /              \     /
   +---+    \ +-----+          +-----+         +-----+ /    +---+
   | Y |------| rtr |----------| rtr |---------| rtr |------| Z |
   +---+      +-----+          +-----+         +-----+      +---+
```

Figure 19: Islands of IPv6

4.3. Other mechanisms

Additional mechanisms for transition or for IPv4/IPv6 coexistence are also under discussion. For example, IPv4 multicast can be used to support neighbor discovery by isolated IPv6 nodes [5]. There are several proposals on how to support transactions between IPv4-only nodes and IPv6 nodes that do not have IPv4-compatible addresses.

IETF members are putting intense effort into transition, as well as the basic IPv6 protocol specification. The combination of tunnels, compatible addresses, and dual-stack nodes gives network administrators the range of flexibility and interoperability they need to deploy IPv6. Transition services allow organizations depending upon current IPv4 networks to take advantage of the more technical IPv6 features.

5. Security Considerations

Sections 2.2.4, 3.8, and 3.9 of this paper emphasize the security benefits that IPv6 offers. By adopting IPv6, the Internet and the enterprise-specific applications will be much better able to satisfy their security needs by making use of standardized network features. Expediting the deployment for IPv6 will bring these security features into service sooner. Furthermore, the Internet will be able to avoid the security pitfalls made more likely by the deployment of NAT devices, as discussed in Section 2.2.2, and arising from any applications using IPv4 source routing (see section 3.5).

6. Acknowledgments

This work is derived from a Bay Networks white paper on IPv6 (published in 1997) that was co-authored by Steve King, Ruth Fax, Haskin, Wenken Ling, and Tom Meehan of Bay Networks. Thanks to Steve Deering and Bob Hinden for their many efforts as chairs of the IPng working group.

Full Copyright Statement

A. Myths

Because of its and the number of detailed technical choices that had to be made, the birth of IPv6 has been attended by some controversy, and by a number of somewhat misleading stories that can distract network owners who are in the process of crafting their forward-looking network strategy. Confusion is to be expected, considering the implications of migrating our global internetwork infrastructure to an updated protocol. But if the IPv6 myths are perpetuated indefinitely, there's a risk that the Internet will not be able to progress beyond a patched-up version of IPv4. In these appendices, we try to counteract some of these myths.

Myth #1: The only driving force behind IPv6 is address space depletion.

Many of the discussions about a new Internet protocol focus on the fact that we will sooner or later run out of globally unique network layer addresses, due to IPv4's fixed 32-bit address space. The various address registries that assign blocks of IP addresses to large network service providers and network operators have become cautious about the way these addresses are handed out, though most predictions for IPv4 address exhaustion target a time frame that starts well into the next decade.

With the long-haul in mind, IPv6 has been outfitted with a 128-bit address space that should guarantee globally unique addresses for every conceivable variety of network device for the foreseeable future (i.e., decades). IPv6 has 16 byte addresses, or

340,282,366,920,938,463,463,374,607,431,768,211,456

addresses (over a third of a duodecillion of them, in fact). The number of addresses gets a lot of attention but it is only one of many important issues that IPv6 designers have tackled. Other IPv6 capabilities have been developed in direct response to current business requirements for more scalable network architectures, mandatory security and data integrity, extended quality-of-service (QoS), autoconfiguration, and more efficient network route aggregation at the global backbone level. These features are all specified with IPv6 in a way that would be difficult to realize as effectively in IPv4.

Myth #2: Extensions to IPv4 can replicate IPv6 functionality.

There have been multiple efforts to extend the life of IPv4 incrementally with evolutionary changes to the protocol standards and various proprietary techniques. One such example is the development of network address translators (NAT) that preserve IPv4 address space by intercepting traffic and converting private intra-enterprise addresses into one or a few globally unique Internet addresses. Other examples include the various QoS and security enhancements to IPv4, which are in general scaled-back or identical to mechanisms specified in IPv6.

We do not know how long IPv4's life can be extended by these techniques. What is certain is that the widespread introduction of NAT devices negatively affects the end-to-end viability of emerging Internet applications; in practice only a limited set of well-known applications can be correctly handled by NAT devices or by application level gateways associated with them. In particular NAT devices prevent the deployment of end-to-end IPv4 security. Furthermore, the development of new and innovative Internet applications is burdened with the design constraints posed by NATs [18]. Since NAT is strictly unnecessary for IPv6, standard end-to-end IPv6 security can be deployed, and a future enlivened by new lightweight and more fully functional applications can be envisioned. NAT translation is also known to create great difficulty in the construction of Virtual Private Networks (VPNs), since it turns address space administration into a nightmare and interferes with standard security mechanisms.

NAT also only works in a "flat universe" for a site accessing the global Internet - even moderately-sized enterprises are not flat internally, with nested multi-party relationships. Realistic NAT deployment solutions would have to include routing via multiple ingress/egress NATs for load balancing, multi-NAT-hop routes and so on - all this would create in miniature the v4 (or in fact v6) architecture, since it is solving the same problem, but piecewise and badly.

It is hard to compare the costs of converting to IPv6 with those of remaining with IPv4 and its upgrades. Every network manager will have to make this comparison; but staying with IPv4 has been likened to the situation of a lobster in a pot of water, as the temperature slowly increases - at first, it feels comfortable.

Myth #3: IPv6 support for a large diversity of network devices is not an end-user or business concern.

Over the next few years, conventional computers on the Internet will be joined by a myriad of new devices, including palmtop personal data assistants (PDA), hybrid mobile phone technology with data processing capabilities, smart set-top boxes with integrated Web browsers, and embedded network components in equipment ranging from office copy machines to kitchen appliances. Some of the new devices requiring IP addresses and connectivity will be consumer-oriented, but many will become integral to the information management functions of corporations and institutions of all sizes. These new devices require features not fully understood by most protocol designers during the initial growth of the IPv4 Internet.

IPv6's 128-bit address space will allow businesses to deploy a huge array of new desktop, mobile, and embedded network devices in a cost-effective, manageable way. Further, IPv6's autoconfiguration features will make it feasible for large numbers of devices to attach dynamically to the network, without incurring unsupportable costs for the administration for an ever-increasing number of adds, moves, and changes.

The business requirement for IPv6 will be driven by end-user applications. Applications for mobile nodes, electronic commerce, and those needing specialized routing features will be easier to design and implement using IPv6, especially as compared to IPv4 patched by NAT. To remain competitive in the coming era of high-density networking, businesses should exploit IPv6 to create a highly scalable address space and robust autoconfiguration services that will remain viable in the face of an explosion of end-user networking needs.

Myth #4: IPv6 is primarily relevant to backbone routers, not end-user applications.

It is true that IPv6 address aggregation allows efficient multitiered routing hierarchies that limit the uncontrolled growth of backbone router tables. But many of the advanced features of IPv6 also bring direct benefits to end-user applications at the workgroup and departmental levels. For instance, applications will have available the mandatory IPv6 encryption and authentication services as an integral part of the IP stack. For mobile business users and changing organizations, IPv6 autoconfiguration will allow the efficient assignment of IP addresses without the delays and cost associated with manual address administration or even traditional DHCP, which takes place in many current IP networks. IPv6 is very much both an end-user concern and a business concern. This concern will become increasingly important as QoS flows and QoS routing become important architectural components of the Internet.

Myth #5: Asynchronous Transfer Mode (ATM) cell switching will negate the need for IPv6.

ATM and other switching methods offer interesting technology for present and future internetworks, but ATM is, by itself, not a replacement for packet routing Internet architecture. ATM is better understood as a link-layer technology over a non-broadcast multiple access (NBMA) medium. It gives some isolation properties, and offers the promise for offering improved Quality of Service (QoS) connections for applications that need it. Even these hypothetical advantages are not yet fully developed for ATM, and it is possible that these advantages will be equally well available in future IPv6 networks not running over ATM.

Fortunately, network owners do not have to make a choice between ATM or IPv6 because the two protocols will continue to serve different and complementary roles in corporate networking. Large networks will make use of both protocols. For many network designers, ATM is a useful transmission medium for high-speed IPv6 backbone networks. Standards and development work is being devoted to integrating ATM and IPv6 environments. IPv6, like its predecessor IPv4, provides network layer services over all major link types, including ATM, Ethernet, Token Ring, ISDN, Frame Relay, and T1.

Myth #6: IPv6 is something that only large telephone companies or the government should worry about.

Some Internet pundits have characterized IPv6 as a concern that's outside the corporate network and outside the current time frame. In reality, IPv6 is a standards track and mainstream solution for the operation and continued efficiency of day-to-day business activities. But the only way that IPv6 will take hold and succeed is if businesses and institutions of all types come to terms with the inadequacies of IPv4 and begin to lay plans for migration. In the past few years, Internet protocols have enabled a whole new style of distributed commerce that brings people together inside enterprises and gives enterprises access to the entire world. In fact, the sustained and impressive growth of the Internet, which has inspired the current engineering efforts for IPv6, is in large measure due to the penetration of the World Wide Web to business and consumer end users. Offering services to such end users is of interest to many more institutions than merely governments and telephone companies.

Myth #7: IPv6 requires extensive modifications to existing operating systems, applications, and programming techniques.

IPv6 obviously requires certain modifications to the network protocol handling modules installed on the relevant computers. However, this typically requires little or no change to the base operating system. Simple and natural modifications, typically confined to fewer than a dozen lines of the programs, can be made to enable applications to use IPv6 addresses directly. Since IPv6 reserves a part of its address space for compatibility with IPv4 addresses, applications modified to handle IPv6 addresses can still communicate with existing IPv4 clients and servers.

Moreover, the transition strategies defined for IPv6 deployment within the IPv4 Internet should make the gradual adoption of IPv6 a smooth process that allows existing applications to be converted for native IPv6 operation in a gradual, controlled manner.

References

[1] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions. RFC 2132, March 1997.

[2] J. Bound and C. Perkins. Dynamic Host Configuration Protocol for IPv6. draft-ietf-dhc-dhcpv6-14.txt, February 1999. (work in progress).

[3] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205: Resource ReSerVation Protocol (RSVP) --- version 1 functional specification, September 1997. Status: PROPOSED STANDARD.

[4] B. Carpenter. RFC 1958: Architectural principles of the Internet, June 1996. Status: INFORMATIONAL.

[5] B. Carpenter and C. Jung. Transmission of IPv6 over IPv4 Domains without Explicit Tunnels. draft-ietf-ipngwg-6over4-02.txt, January 1999. (work in progress).

[6] R. Coltun, D. Ferguson, and J. Moy. OSPF for IPv6. draft-ietf-ospf-ospfv6-05.txt, November 1997. (work in progress).

[7] A. Conta and S. Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6). RFC 1885, December 1995.

[8] A. Conta and S. Deering. RFC 2463: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, December 1998. Obsoletes RFC1885 [7]. Status: DRAFT STANDARD.

[9] Matt Crawford. Router Renumbering for IPv6. draft-ietf-ipngwg-router-renum-07.txt, January 1999. (work in progress).

[10] S. Deering. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. Request for Comments (Experimental) 2117, Internet Engineering Task Force, June 1997.

[11] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, December 1995.

[12] S. Deering and R. Hinden. RFC 2460: Internet Protocol, Version 6 (IPv6) specification, December 1998. Obsoletes RFC1883 [11]. Status: DRAFT STANDARD.

[13] S. Deering, C. Partridge, and D. Waitzman. Distance Vector Multicast Routing Protocol. Request for Comments (Experimental) 1075, Internet Engineering Task Force, November 1988.

[14] Stephen E. Deering, Editor. ICMP Router Discovery Messages. RFC 1256, September 1991.

[15] M. Degermark, B. Nordgren, and S. Pink. Header Compression for IPv6, February 1999. Status: PROPOSED STANDARD.

[16] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[17] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy. RFC 1519, September 1993.

[18] T. Hain. Architectural implications of NAT. draft-iab-nat-implications-02.txt, October 1998. (work in progress).

[19] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 1884, December 1995.

[20] IEEE. Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, March 1997. http://standards.ieee.org/db/oui/tutorials/EUI64.html.

[21] D. Johnson and S. Deering. Reserved IPv6 Subnet Anycast Addresses. draft-ietf-ipngwg-resv-anycast-00.txt, August 1998. (work in progress).

[22] D. Johnson and C. Perkins. Mobility Support in IPv6. draft-ietf-mobileip-ipv6-07.txt, November 1998. (work in progress).

[23] H. Krawczyk, M. Bellare, and R. Cannetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, February 1997.

[24] G. Malkin and R. Minnear. RFC 2080: RIPng for IPv6, January 1997. Status: PROPOSED STANDARD.

[25] J. McCann, S. Deering, and J. Mogul. Path MTU Discovery for IP version 6. RFC 1981, August 1996.

[26] J. Moy. Multicast extensions to OSPF. Request for Comments (Proposed Standard) 1584, Internet Engineering Task Force, March 1994.

[27] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP version 6 (IPv6). RFC 1970, August 1996.

[28] T. Narten, E. Nordmark, and W. Simpson. RFC 2461: Neighbor discovery for IP Version 6 (IPv6), December 1998. Obsoletes RFC1970 [27]. Status: DRAFT STANDARD.

[29] C. Partridge, T. Mendez, and W. Milliken. Host anycasting service. Request for Comments (Informational) 1546, Internet Engineering Task Force, November 1993.

[30] C. Perkins. Extensions for the Dynamic Host Configuration Protocol for IPv6. draft-ietf-dhc-dhcpv6ext-11.txt, February 1999. (work in progress).

[31] David C. Plummer. An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware. RFC 826, November 1982.

[32] J. B. Postel, Editor. Internet Control Message Protocol. RFC 792, September 1981.

[33] Y. Rekhter and T. Li. An Architecture for IP Address Allocation with CIDR. RFC 1518, September 1993.

[34] R. Talpade and M. Ammar. RFC 2149: Multicast server architectures for MARS-based ATM multicasting, May 1997. Status: INFORMATIONAL.

[35] S. Thomson and C. Huitema. DNS extensions to support IP version 6. Request for Comments (Proposed Standard) 1886, Internet Engineering Task Force, January 1996.

[36] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 1971, August 1996.

[37] S. Thomson and T. Narten. RFC 2462: IPv6 stateless address autoconfiguration, December 1998. Obsoletes RFC1971 [36]. Status: DRAFT STANDARD.

Authors' and Editors' Addresses

Original Authors:

- Steve King, Bay Networks
- Ruth Fax, Bay Networks
- Dimitry Haskin, Bay Networks
- Wenken Ling, Bay Networks
- Tom Meehan, Bay Networks

Questions about this memo can be directed to the editors:

Robert Fink
Esnet R&D
Lawrence Berkeley Nat'l Laboratory
1 Cyclotron Road
Bldg. 50A, Room 3139
Mail-Stop 50A-3111
Berkeley, CA 94720
USA

phone: +1 510 486-5692
fax: +1 510 486-4790
e-mail: rlfink@lbl.gov

Charles E. Perkins
Networking and Security Center
Sun Microsystems Laboratories
15 Network Circle
Room 2682
Mail Stop MPK15-214
Menlo Park, CA 94025
USA

+1-650-786-6464
+1-650-786-6445
cperkins@Eng.sun.com
http://www.svrloc.org/~charliep