Realizing the Transition to IPv6

Daniel G. Waddington and Fangzhe Chang, Bell Research Laboratories

ABSTRACT

While the details of the next-generation IPv6 protocol are now reaching maturity, the next hurdle in realizing the promises of IPv6 is the need for deployment on a wider scale. Until recently, the migration from IPv4 to IPv6 has been considered nontrivial, a factor generally attributed to thwarting its success. However, with the advent of a number of new transitioning techniques the evolution to IPv6 is now becoming increasingly feasible. These transitioning techniques offer tunneling and translation solutions that enable the gradual introduction of IPv6 support into an existing IPv4 infrastructure. Nevertheless, it is not yet clear what form this evolution is likely to take, which phases are likely to exist, and how the transition process will proceed. This article briefly examines existing IETF IPv6 transitioning mechanisms and discusses the key issues involved in IPv6 deployment. We examine those aspects that potentially affect choice of transition mechanisms and look at what factors are likely to mould the evolutionary path.

INTRODUCTION

As many are already aware, an increasingly likely candidate for the next-generation Internet Protocol is version 6 (IPv6), defined by Internet Engineering Task Force (IETF) RFC 2373 [1]. The proponents of IPv6 do not consider it a revolutionary protocol, designed to replace the existing IPv4, but more a long awaited improvement on the original IETF designs founded back in 1981. Much of its development has been influenced by lessons learned in the existing Internet. As a technology it promises a number of advances, including:

- A larger address space and flexible addressing scheme
- More efficient packet forwarding
- Inherent support for secure communications
- The ability to allow differentiated services
- Better support for mobility
- Ease of management

Deployment of IPv6 is not going to happen overnight. Instead, the Internet will evolve toward IPv6, initially through isolated "islands" and then gradual global saturation. One might envisage this evolution process to take the form of dual-stacked nodes, where every node in the Internet is both IPv4 and IPv6 capable. However, this would cause unnecessary complexity as functionality is replicated both in the network and the end systems.

The transition to IPv6 is also not entirely transparent to the networking layers above IP. IPv6 addresses are longer than IPv4 addresses, requiring a change in application data structures that embed IP addresses. Consequently, application programming interfaces (APIs) (e.g., sockets) must be extended to support both IPv4 and IPv6, as well as the ability to select the appropriate protocol for each interhost application communication. In general, legacy applications written for IPv4 need to be either rewritten or bridged to support IPv6. For example, FTP embeds IP addresses in its protocol, thus requiring changes to both the client and server applications.

In reality, the Internet is likely to become a complex conglomeration of different protocols. IPv4 will exist with IPv6 and other globally standardized protocols. The likelihood that IPv6 will someday grow to be as prevalent as its predecessor is certainly increasing. The main reasons for this are twofold. First, the escalating number of IETF proposed transitioning mechanisms are giving network administrators an easier path to migration by permitting network nodes, and more specifically applications on these nodes, to communicate with each other over a mix of end system and network device (e.g., switches and routers) capabilities. Second, specialized application domains with a respective market interest, particularly the mobile domain [2], are demanding IP features that cannot be fulfilled by IPv4, such as wider address space availability and ease of configuration.

Assuming the imminent success of IPv6, the next likely hurdle is in realizing and managing the transitional process. Even the wide array of available IETF transitioning mechanisms unfortunately do not mean that the move to IPv6 is going to be simple. To date, a large amount of effort has focused on specific techniques for tunneling, translation, and other transitional approaches. Nevertheless, the real issues now concern what form the evolutionary process will

0163-6804/02/$17.00 © 2002 IEEE

take and how the whole process can be managed in order to achieve a smooth and seamless transition.

In this article we discuss issues surrounding actual realization of IPv6. We present a consolidated view of IPv6 transitioning and the issues involved. To begin, we briefly review existing IETF proposals for IPv6 transitioning mechanisms and provide an overall view of their roles and individual features. We discuss attributes that are likely to change throughout the migrational process and economic factors that might influence decisions on choosing transitional solutions. We then review practical and technical issues concerning the deployment of IPv6 and offer some discussion of the less obvious obstacles that are likely to arise. Finally, we round up and offer our conclusions.

Existing Transition Mechanisms and Approaches

Transitioning mechanisms generally come in one of three forms: *dual stacks*, *tunneling,* and *translation*.

The principal building block for transitioning is the *dual stack*. Dual stacks, as the name suggests, literally maintain two protocol stacks that operate in parallel and thus allow the device to operate via either protocol. Dual stacks can be implemented in both end systems and network devices. In the end system they enable both IPv4- and IPv6-capable applications to operate on the same node. Dual-stacked capabilities in the network device allow handling of both IPv4 and IPv6 packet types.

Dual-stack mechanisms do not, by themselves, solve IPv4 and IPv6 interworking problems; the second building block, *translation*, is required for this. Translation refers to the direct conversion of protocols (e.g., between IPv4 and IPv6) and may include transformation of both the protocol header and the protocol payload (Fig. 1). Translation can occur at several layers in the protocol stack, including network, transport, and application layers. Protocol translation often results in feature loss, where there is no clear mapping between the features provided by translated protocols. For instance, translation of an IPv6 header into an IPv4 header will lead to the loss of the IPv6 flow label.

Translation mechanisms are either *stateless* or *stateful*. A stateless translator is able to process each conversion individually without any reference to previously translated packets; a stateful translator needs to maintain some form of state with respect to previous translations. For instance, in IPv6-to-IPv4 address translation, the translator must maintain a mapping between the two types of IP addresses.

Both end systems and network devices can be used to perform the translation process. Translation is considered *transparent* when traffic is inherently routed through a translator in the network (i.e., routing to the translator is not explicitly enforced by the end system itself).

The final building block for transitioning is *tunneling*. Tunneling is used to bridge compatible networking nodes across incompatible networks. It can be viewed technically as the transfer of a payload protocol by an encapsulating carrier protocol between two nodes and/or end systems. Encapsulation of the payload protocol is performed at the tunnel entrance and de-encapsulation is performed at the tunnel exit point (there is therefore a logical direction to a given tunnel). This logical association between tunnel entry and exit end-points is what defines the tunnel.

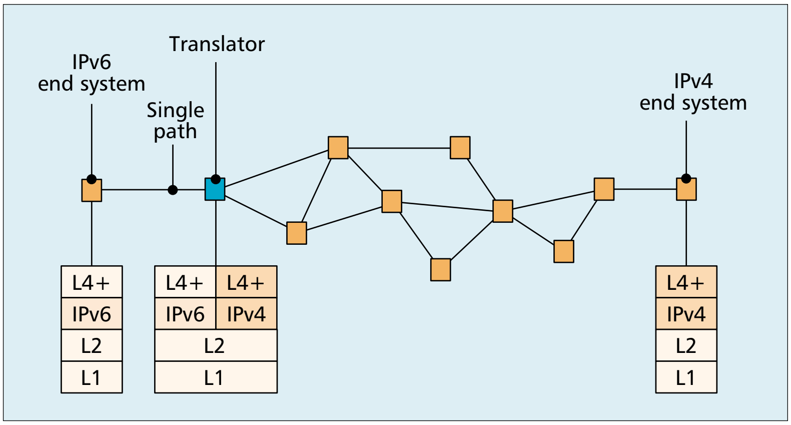


Figure 1. *Two-way IPv6/IPv4 translation at the network edge.*

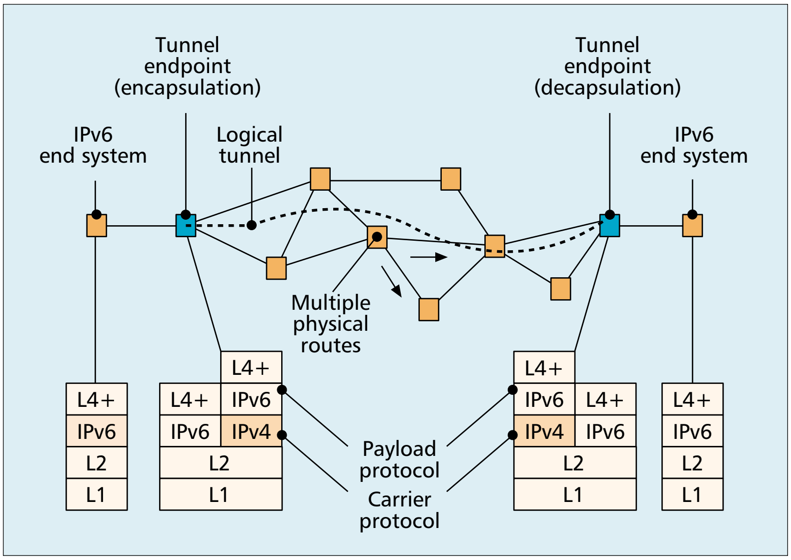


Figure 2. *Transparent IPv6 in IPv4 tunneling.*

The principal problem in tunnel deployment is the configuration of the tunnel endpoints, defining where encapsulation should be applied and to what packets it should be applied. Tunnel endpoint addresses are generally attained:

- By manual or tool-based parameter entry (e.g., tunnel broker, IETF RFC 3053)
- Through existing services, such as a well-known DNS service name or DHCP options
- By embedding information in the link layer addresses or IP addresses (e.g., IPv6 prefix or interface identifier parts)
- By using an IPv6 anycast address

From the perspective of IPv4/IPv6 transitioning, tunneling is in most cases used to bridge incompatible IP segments: an IPv6 payload over an IPv4 carrier, or an IPv4 payload over an IPv6

Sequential tunneling can be used to tunnel across finer grained segments in an end-to-end path. The use of hierarchical and sequential tunnel configurations does inevitably lead to increased processing requirements and packet delay.

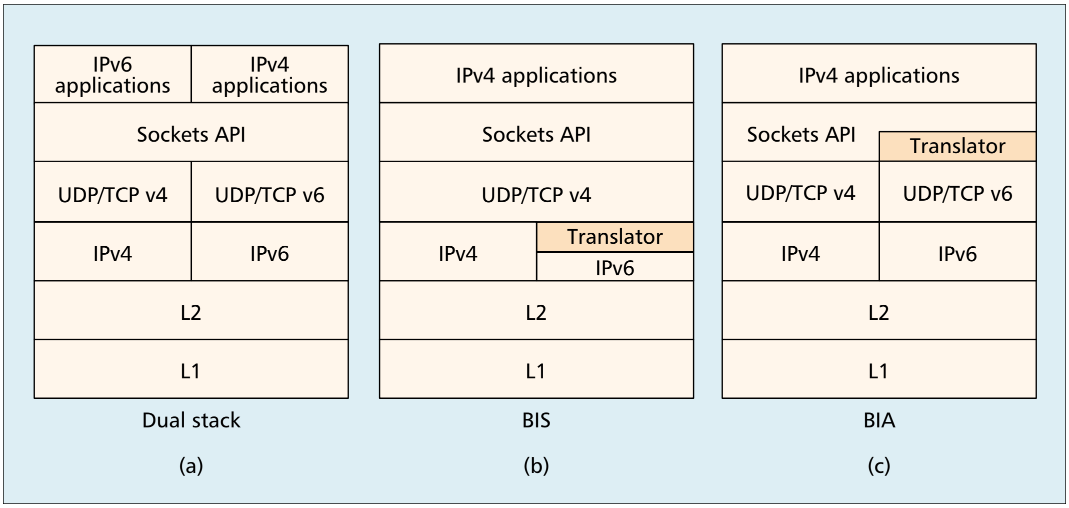


Figure 3. *End system protocol stack transition mechanisms.*

carrier. Figure 2 illustrates tunneling of IPv6 packets in IPv4 packets.

End systems and network devices may act as tunnel endpoints, performing encapsulation or decapsulation. In most cases, tunneling is deployed in a simple point-to-point configuration. However, tunnels can also exist both hierarchically (i.e., a tunnel within a tunnel) and sequentially (i.e., concatenated tunnels). Hierarchical configurations are often used where tunnels for the purpose of transitioning exist with tunnels for the purpose of security and QoS provisioning. For example, an IPSec tunnel (IETF RFC 1825) providing security may itself be tunneled inside a transitioning tunnel that provides tunneling across incompatible networks. Sequential tunneling can be used to tunnel across finer-grained segments in an end-to-end path (e.g., from the end system to the local gateway, and from the gateway and beyond). The use of hierarchical and sequential tunnel configurations inevitably leads to increased processing requirements and packet delay.

Table 1 overviews the currently proposed IETF transition mechanisms, how they can be classified with respect to connectivity, and the elements required for their deployment. Connectivity refers to the relationship between the session-instantiating node and the corresponding node. For example, 6-to-4 means that a node supporting IPv6 only is able to correspond with a node supporting IPv4 only. This might be between single or cooperating end systems (ESs) and network devices (NDs). Communication is generally bidirectional; however, the ordering 6-to-4 infers that the IPv6 node is responsible for session instantiation.

Each solution has individual characteristics, and each plays a specific role in the transitioning problem. The rest of this section reviews the IETF transitioning mechanisms shown in Table 1. In-depth details can be gained from the respective IETF drafts or RFCs.

IPv6/IPv4 Dual-stack

In the dual-stack scheme (IETF RFC 2893) a network node installs both IPv4 and IPv6 stacks in parallel (Fig. 3a). IPv4 applications use the IPv4 stack, and IPv6 applications use the IPv6 stack. Flow decisions are based on the IP header version field for receiving, and on the destination address type for sending. The address types typically come from DNS lookups; the appropriate stack is chosen in response to returned DNS record types.

Many off-the-shelf commercial operating systems already provide dual IP protocol stacks. Consequently, the dual-stack mechanism is the most widely deployed transitioning solution. However, note that dual stacks only enable like nodes to communicate (e.g., IPv6-IPv6 and IPv4-IPv4). Much more is required for a complete solution that enables IPv6-IPv4 and IPv4-IPv6 communications.

IPv4/IPv6 Translation Mechanisms

The basic role of translation in IPv4/IPv6 transitioning is the conversion of IP and ICMP packets. Many translation algorithms are based on the algorithm known as SIIT.

SIIT: The Stateless IP/ICMP Translation algorithm (SIIT) (IETF RFC 2765) specifies a bidirectional translation algorithm between IPv4 and IPv6 packet headers, as well as between ICMPv4 and ICMPv6 messages. SIIT ignores many IPv6 extension headers (except fragment headers) and IPv4 options. The translation has been designed so that UDP and TCP pseudo header checksums are not affected by the translation process. SIIT is used as the basis for BIS and NAT-PT, which are discussed below.

Translators in the end systems can solve application to network interoperability problems. They are relatively easy to implement, but are often more difficult to manage on a larger scale. The term *bump* is used to denote additional processing modules in a conventional TCP/IP stack. The two end system translators currently proposed by the IETF are BIS and BIA. Both of these are aimed at allowing IPv4 applications to operate over an IPv6 network in order to meet legacy application requirements.

BIS: The Bump-In-the-Stack (BIS) (IETF RFC 2767) solution comprises a TCP/IPv4 module and a translator module, which consists of three bump components and is layered above an

Name	Connectivity	Type	Location
Dual stack	4-to-4 over 4, 6-to-6 over 6	Dual stack	In single ES or ND
SIIT	6-to-4, 4-to-6	Translator	In single ES or ND
Bump-in-Stack (BIS)	4-to-6	Translator	In single ES
Bump-in-API (BIA)	4-to-6	Translator	In single ES
NAT-PT	6-to-4, 4-to-6	Translator	In single ND
MTP	4-to-6,4-to-6 (multicast)	Translator	In single ND
TRT	6-to-4	Translator	In single ND
SOCKS64	4-to-6, 4-to-6	Translator	Between ES and ND
6over4	6-to-6 over 4	Tunnel	Between ES and ND
ISATAP	6-to-6 over 4	Tunnel	Between ES and ND
DSTM	4-to-4 over 6	Tunnel	Between ES and ND
Configured IP-in-IP	6-to-6 over 4, 4-to-4 over 6	Tunnel	Between ES and ND, two NDs or two ESs
6to4	6-to-6 over 4	Tunnel	Between two NDs

Table 1. *Classifying IETF transition mechanisms.*

The dual-stack mechanism is the most widely deployed transitioning solution. However, note that dual stacks only enable like nodes to communicate. Much more is required for a complete solution that enables IPv6-IPv4 and IPv4-IPv6 communications.

IPv6 module (Fig. 3b). Packets from IPv4 applications flow into the TCP/IPv4 module. Here, identified packets are translated into IPv6 and in turn forwarded into the IPv6 module, and so forth. The three bump components include the *extension name resolver*, which snoops DNS lookups to decide whether the peer node is IPv6-only; the *address mapper*; which allocates a temporary IPv4 address for the IPv6 peer and caches the address mapping; and finally, the *translator*, which translates packets between IPv4 and IPv6.

Temporary IPv4 addresses are only visible within the end system and are therefore typically from a private address space. As a result, BIS is only suited to applications that don't exchange address-dependent fields in their application layer protocols. For example, nonpassive FTP will not interoperate with BIS.

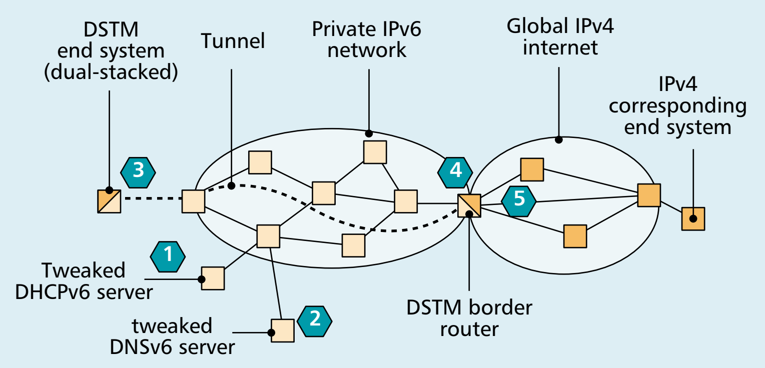
BIA: Bump-In-the-API (BIA) [3], like BIS, also permits IPv4 applications to communicate with peers over the IPv6 network. The difference is that the bump layer is inserted higher up, as part of the socket layer, enabling the interception of Socket API calls (Fig. 3c). The location of the BIA module avoids the translation of IP packets (thus allowing IP-level security) and, unlike BIS, avoids modifications to the operating system kernel. BIA implementations consist of three bump components: a *name resolver*, an *address mapper*, and a *function mapper*. The first two behave in a manner similar to BIS. The function mapper intercepts IPv4 socket function calls and translates them to the equivalent IPv6 socket calls. As with BIS, BIA can also use any temporary IPv4 address, and again suffers from the inability to handle embedded addresses in the application layer protocol.

To avoid increased complexity in the end system, which often leads to scalability problems in larger deployments, translators can alternatively be deployed within the network. However, translation processing is relatively heavyweight and therefore is generally only feasible at the network edge rather than within the core. Proposed mechanisms for translation in network devices operate at either the network or transport layer. NAT-PT and MTP are IETF proposed network layer translators. The former translates unicast packets, whilst the latter translates multicast packets. TRT and SOCKS64, on the other hand, are transport layer translators.

NAT-PT: The Network Address Translation-Protocol Translation (NAT-PT) (IETF RFC 2766) is a stateful IPv4/IPv6 translator that uses the SIIT algorithm previously mentioned. The NAT-PT device serves multiple IPv6 nodes, allocates a temporary IPv4 address to each, and acts as a communication proxy with IPv4 peers. Allocation is triggered either by the first outbound IPv6 packet (using an IPv4-compatible IPv6 destination address) or by the inbound IPv4 DNS lookup (from the peer) arriving at a co-located Application Level Gateway (ALG). Because NAT-PT maintains translation state, each session must be routed via the same NAT-PT device (unless state information is exchangeable across a load-balancing cluster).

MTP: The Multicast Translator based on IGMP/MLD Proxying (MTP) [4] proposes an architecture for translating multicast packets between IPv4 and IPv6. The translator is located at the site boundary between IPv4 and IPv6, and comprises an address mapper and a multicast translator. The address mapper translates between IPv4 multicast addresses and IPv4-compatible IPv6 multicast addresses (represented by prefix FFxx::/96 followed by the 32 bit IPv4 multicast address).

The multicast translator consists of an *IPv4 multicast proxy* that joins IPv4 multicast groups on behalf of IPv6 receivers, an *IPv6 multicast proxy* that joins IPv6 multicast groups on behalf of IPv4 receivers, and a *translator* that gets multicast packets from the proxies, obtains address mappings from the address mapper, and trans-



1. Sessions initiated by the DSTM host cause the DHCPv6 server to allocate a temporary IPv4 address and specify the address of the tunnel endpoint (i.e., DSTM border router).
2. Sessions initiated by an external IPv4 host trigger a DNS lookup which in turn triggers IPv4 address allocation.
3. The DSTM host encapsulates outgoing IPv4 packets in IPv6 packets and forwards to the offlink DSTM border router.
4. Incoming packets are tunneled from the border router to the DSTM host which decapsulates them on receipt.
5. The DSTM border router forwards and receives IPv4 packets from the global Internet.

Figure 4. *The Dual Stack Transition Mechanism (DSTM).*

lates, then forwards the multicast packets over a different IP. Automated techniques for proxy deployment are still undefined and still considered an administrative task; hence, the scalability of this approach is still under scrutiny.

Transport layer relays, such as those found in firewall products, can also be extended into IPv6/IPv4 translators. A relay process on the border router partitions the transport layer path into two "terminated" segments, where each segment supports different IP versions. Packets traversing the relay pass up through to the transport layer and then get sent out in the adjacent segment. Translation only occurs at layer 4 and above; therefore, IP layer conversion is avoided. However, processing packets at higher layers often leads to poor performance and, as with translation techniques, end-to-end security often cannot be attained. As with all stateful translators, traffic between given peers should pass through the same relay router. The most widely used relay techniques available today are TRT and SOCKS64.

TRT: The Transport Relay Translator (TRT) [5] translates between TCP/UDPv6 and TCP/UDPv4 sessions. Communication is initiated from the IPv6 side via a special destination address type (a 64-bit prefix followed by the IPv4 address of the destination node). The routing information is configured to route this prefix toward the dual-stacked TRT router, which terminates the IPv6 session and initiates IPv4 communication to the final destination.

SOCKS64: SOCKS64 [6] uses a dual-stacked SOCKS64 router and "socksified" applications to enable communication between IPv4 and IPv6 nodes. Applications are socksified by using a special SOCKS64 library that replaces Socket and DNS APIs. The SOCKS64 library intercepts session-initiating DNS name lookups from the end system application and responds with "fake" IP addresses mapped for the given session. The SOCKS64 library also issues session control calls (e.g., TCP connect) to the local SOCKS64 router, which in turn uses the real IP address to establish a session with the final destination via a different IP version.

IPv4/IPv6 Tunneling Mechanisms

Tunneling, from the perspective of transitioning, enables incompatible networks to be bridged and is typically deployed in a point-to-point or sequential fashion. Two common scenarios are:

- To allow end systems to use offlink transitional devices (e.g., dual-stacked routers) in a sparsely distributed transitioning network
- To enable network edge devices to interconnect over incompatible networks

Three IETF mechanisms are proposed for the former scenario: 6over4, ISATAP, and DSTM.

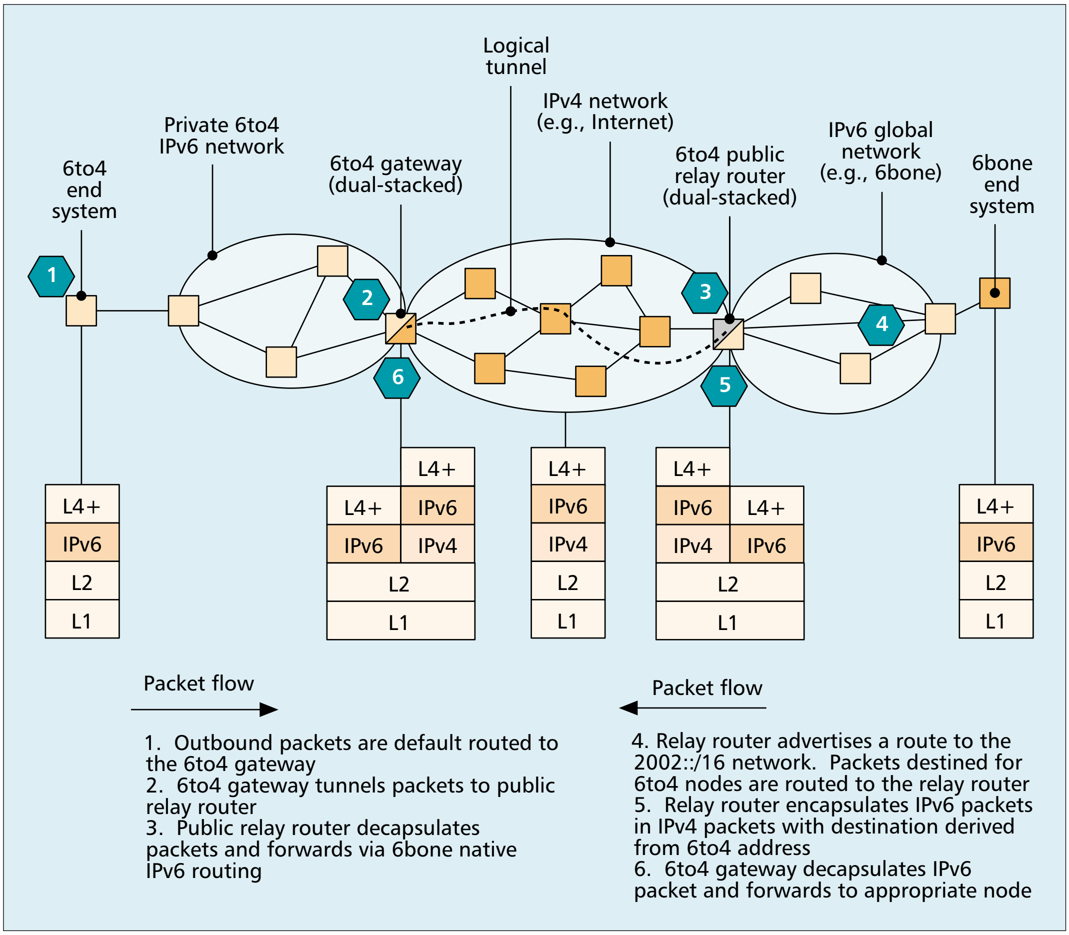
6over4: 6over4 (IETF RFC 2529) embeds IPv4 addresses in the IPv6 address link layer identifier part (i.e., last 64 bits) and defines Neighbor Discovery (ND) (IETF RFC 2461) over IPv4 by using organization-local multicast (IETF RFC 2365). This use of multicast means that the IPv4 network effectively behaves as a virtual LAN. A sender resolves the IPv6 target address (i.e., that of the offlink router or isolated end system) on the virtual LAN via ND. The resulting address bears the destination tunnel endpoint's IPv4 address.

6over4 maintains all of the features of IPv6, including end-to-end security and stateless autoconfiguration, and supports multicast by defining a mapping between IPv6 multicast addresses and IPv4 organization-local multicast addresses. Because the multicast is scoped, the isolated end systems can also use private IPv4 address space.

ISATAP: The Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [7] is similar to 6over4, enabling dual-stack end systems to reach IPv6 routers that are not directly connected, via tunneling. Tunneling to the end system is automated using IPv6-IPv4 compatibility addresses (IETF RFC 2373). These embed an IPv4 address in the interface identifier part (i.e., prefix 0x02005EFE followed by the IPv4 address). Tunneling to the offlink router is achieved by either establishing a new DNS well-known service name for the offlink routers, or assigning the offlink routers a well-known anycast address.

DSTM: The Dual Stack Transition Mechanism (DSTM) [8] enables allocation of temporary IPv4 addresses to dual-stacked end systems that are connected to an IPv6 only network. The scheme tunnels IPv4 packets across the IPv6 network to the global IPv4 Internet.

When sessions are initiated by the DSTM end system, a "tweaked" DHCPv6 server is used to obtain both a temporary IPv4 address and the address of the offlink DSTM border router, to which packets are later tunneled. Alternatively, when sessions are initiated by an IPv4-only node, the DNS lookup request is directed to a tweaked DNS server in the DSTM domain. This server assigns a temporary IPv4 address to the end system. Thus, incoming packets are tunneled to this IPv4 address. Figure 4 below illustrates the DSTM architecture.


Logical tunnel
IPv4 network (e.g., Internet)
Private 6to4 IPv6 network
6to4 end system
6to4 gateway (dual-stacked)
6to4 public relay router (dual-stacked)
IPv6 global network (e.g., 6bone)
6bone end system
1
2
3
4
5
6
L4+
IPv6
L2
L1
L4+
IPv6
IPv6
IPv4
L2
L1
L4+
IPv6
IPv4
L2
L1
L4+
IPv6
IPv4
IPv6
L2
L1
L4+
IPv6
L2
L1
Packet flow
Packet flow

1. Outbound packets are default routed to the 6to4 gateway
2. 6to4 gateway tunnels packets to public relay router
3. Public relay router decapsulates packets and forwards via 6bone native IPv6 routing

4. Relay router advertises a route to the 2002::/16 network. Packets destined for 6to4 nodes are routed to the relay router
5. Relay router encapsulates IPv6 packets in IPv4 packets with destination derived from 6to4 address
6. 6to4 gateway decapsulates IPv6 packet and forwards to appropriate node

Figure 5. *6to4 automatic tunneling.*

Dual-stack, tunneling and translation mechanisms are only the basic building blocks for transitioning. These individual mechanisms do not provide a complete transitioning solution. Both infrastructural and economic factors also play an important part in forming a complete solution.

The two main IETF tunneling solutions for interconnection across incompatible networks are configured IP-in-IP tunneling and 6to4 automatic tunneling.

Configured IP-in-IP Tunneling: Nodes within the network are statically configured to perform tunneling. Tunneling parameters are managed either through manual data entry or via some automated service provided by a tunnel broker (IETF RFC 3053). Tunnel brokers alleviate the management effort required. Their services are generally provided through Web-based applications that allocate IPv6 address prefixes and return the appropriate tunnel configuration scripts and parameters. Tunnel brokers often periodically check the status of IPv4 tunneling clients. Unreachable clients are generally removed from the tunnel database, and the respective resources are reclaimed.

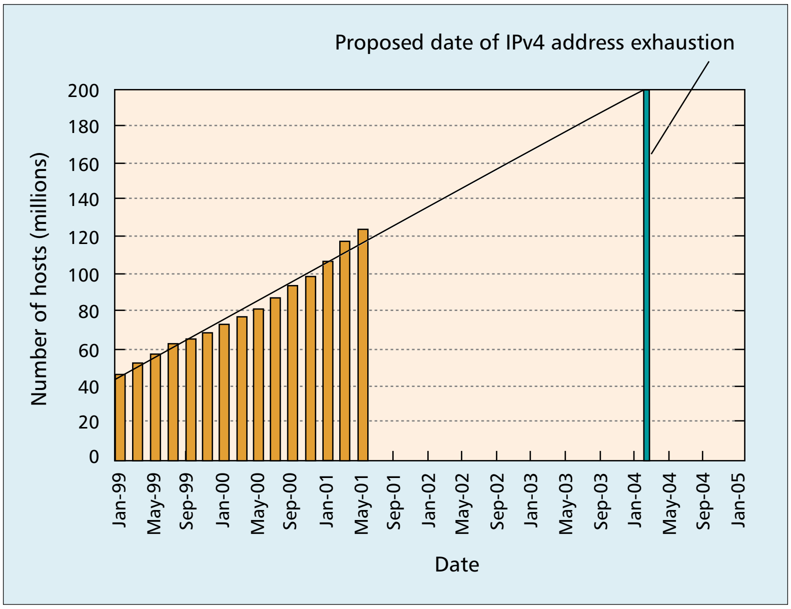
6to4 Automatic Tunneling: Automatic tunneling infers that tunnel configuration is performed without the need for explicit management. 6to4 is the most widely used automatic tunneling technique (IETF RFC 3056). The 6to4 mechanism tunnels IPv6 traffic over IPv4 networks among isolated 6to4 networks. Each 6to4 network assumes a special prefix that embeds the IPv4 address of its 6to4 gateway (2002:V4ADDR::/48). This means that tunnel endpoint addresses are easily obtained and do not need involvement of any IPv6 administrative body. Figure 5 illustrates a typical 6to4 deployment scenario connecting an isolated 6to4 network to the 6Bone via the IPv4 Internet.

As depicted in Fig. 5, each 6to4 network is connected to the rest of the IPv6 network through a local 6to4 gateway and a remote relay router (both are dual-stacked). All IPv6 packets, except for those destined to local addresses, are directed to the gateway. Traffic in the reverse direction, destined for the 6to4 network, is first forwarded to a nearby relay router (advertising the 2002::/16 prefix). This then tunnels the traffic to the appropriate 6to4 gateway using the embedded IPv4 address. In this direction, any relay router can be used since IPv4 routing is ultimately used to locate the 6to4 network.

To support multicast, relay routers may also act as multicast proxies, exchanging group membership information and forwarding multicast packets over the tunnel.

Economic Factors Affecting IPv4 to IPv6 Evolution

Dual stack, tunneling, and translation mechanisms are only the basic building blocks for transitioning. These individual mechanisms do not provide a complete transitioning solution. Both infrastructural and economic factors also play an important part in forming a complete solution. In this section we review some general economic



■ **Figure 6.** *Predicted Internet growth and IPv4 address exhaustion.*

factors that are likely to affect the transition to IPv6. These include end-user demand for IPv6, the need to maintain support for legacy IPv4 applications, issues in upgrading existing network infrastructure, and the availability of IPv6-capable devices in the marketplace.

USER DEMAND FOR IPV6

Before the ubiquity of IPv6, the most compelling reason to deploy IPv6 is to take advantage of its new features and large address space. The principal gains to end users are improved quality of service (QoS) handling through the use of IPv6 flow labels and the knowledge that communications can be secured. Other advantages, such as increased address space, ability to perform autoconfiguration of end systems, standardized security, and efficient header processing, are all generally transparent to end users. They are nevertheless a blessing for the network administrators. Some "specialist" end users (e.g., users of mobile IP) are increasingly in favor of IPv6, its features, and its extensibility. We believe that the demand for IPv6 will be primarily driven by network administrators and specialist end users; therefore, transitioning is likely to start within these domains.

Many proponents of IPv6 believe its wider address space will be the key factor to its success. IPv6 skeptics often take the viewpoint that IPv4 address space is plentiful; currently there are 122 million end systems and a massive 3584 million addresses still available (there are 4294 million theoretical IPv4 addresses, 586 million of which are experimental, multicast, and private). However, the critical point is that the remaining IPv4 address space cannot be allocated with 100 percent efficiency. Huitema (IETF RFC 1715) proposes the H-Ratio which defines a logarithmic ratio of end systems to available address bits. From this, the H-Ratio predicts that only 200 million end systems are likely to consume the rest of the address space. Marrying this analysis, with the extrapolated growth of the Internet, it would seem that IPv4 address exhaustion could become a real problem by 2004 (Fig. 6). Of course, predictions are difficult to make. Factors including demand for new technology (e.g., mobile phones) may influence the Internet's rate of growth.

Another aspect driving demand for IPv6 is the adoption of IPv6 by international standards and specification bodies. For example, the Universal Mobile Telecommunications System (UMTS) Forum (www.umts-forum.org) and Third Generation Partnership Project (www.3gpp.org) both mandate IPv6 in their reference architectures. Thus, IPv6 is becoming a critical piece required for implementation conformance.

MAINTAINING LEGACY IPV4 APPLICATIONS

An important element in the progression of IPv6 is the availability of IPv6 applications. Although the porting process from an IPv4 application to IPv6 is relatively simple, the vast number of legacy IP applications will still take considerable time to move to IPv6. It is likely that some IPv4 applications will never be ported to IPv6. In some cases, the end user will see the migration to IPv6 as a downgrade in their services, whereby their legacy IPv4 applications will not work on an IPv6 only network.

Nevertheless, although a wide array of IPv4 applications exist, a small group of these make up a significant portion overall. This small group includes Web browsers, email readers, news readers, distributed file systems, and the corresponding servers. The availability of IPv6 versions of these applications is already happening. The limited group of commonly used applications are likely to satisfy the requirements of most Internet users, judging from the experience with firewall devices. Many existing firewall devices only allow a constrained set of application protocols to get through, and their use in large corporations is still considered acceptable. Hence, with respect to migration to IPv6, the impact of legacy IPv4 applications may be less significant than initially expected. Meanwhile, the need for IPv6 features will drive the replacement of legacy applications. For example, a user who wishes to have QoS support for multimedia streaming in their Web browser is likely to require an IPv6-enabled version that takes full advantage of the newly defined flow labels.

UPGRADING NETWORK INFRASTRUCTURE

In general, IPv4 infrastructure cannot handle IPv6. On the control plane, routing software is often unable to support IPv6-compatible protocols such as BGP4+ (IETF RFC 2545). On the data plane, the majority of IP network devices use IPv4-based application-specific integrated circuit (ASIC) hardware to achieve better packet processing performance. Obviously, ASICs cannot easily be upgraded. The majority of today's off-the-shelf IPv6-capable routers provide revisions for routing software and only software packet forwarding in the data plane. This approach by router vendors has led to the availability of IPv6 software upgrades for a number of existing IPv4 products. Devices that can be upgraded to IPv6, via software upgrades, can be useful in migrating to the new protocol. However, the principal problem with software upgrades is that of performance degradation. Because IPv6 packets cannot be handled by ASICs on the data path, performance is generally degraded. In experiments we conducted at Bell Labs, we measured the packet handling of a leading commercially available router running software-based IPv6 forwarding. We measured the maximum throughput of two loops (one ingress and one egress) across two pairs of 10 Mb/s Ethernet ports, and examined the effect of loading the two simultaneously with IPv4 and IPv6 traffic. The results showed a performance degradation of up to 36 percent of the normal IPv4 throughput, which is a significant drop in capability. This drop in performance is not surprising, considering that the router uses a relatively low-performance and low-cost processor designed for control-based processing.

Beyond possible degradation in performance, there may also be issues of cost. Software upgrades often entail main memory and flash memory upgrades, primarily because of increased size in the code base. Necessary hardware upgrades can be quite costly; therefore, upgrading a complete network may become an expensive proposition.

Software upgrades often entail main memory and flash memory upgrades, primarily because of an increased size in the code base. Necessary hardware upgrades can be quite costly and therefore upgrading a complete network may become an expensive proposition.

Market Availability of IPv6 Infrastructure

An important factor in the migration to an all-IPv6 network, is the availability of IPv6 infrastructure. Currently, support for IPv6 is available in end system protocol stacks, routers, and some other IP devices. Protocol stacks exist for a wide number of commercial operating systems, including Microsoft Windows, Sun Microsystems Solaris, Linux, IBM AIX, HP UX, OpenBSD, FreeBSD, NetBSD, SCO UNIX, SGI Irix, and MAC OS.

Router support for IPv6 is also becoming increasingly widespread. Current known router vendors that offer IPv6 capabilities in their products include 3Com, Cisco, Hitachi, Juniper, NEC, Nokia, Nortel, Sumitomo, Teledat, and Telebit Communications. At the time of writing, of these vendors, only Hitachi, Juniper, and Nortel have products available that incorporate native IPv6 support in hardware. Common off-the-shelf PCs can also act as low-end IPv6 routers by installing IPv6-capable routing daemons, such as GNU's Zebra (http://www.gnu.org/software/zebra/zebra.html) and the GateD Consortium's Protocol suite (http://www.gated.org/).

Moving beyond end system protocol stacks, the availability of IPv6 support is likely to appear in programmable devices, such as field programmable gate arrays (FPGAs) and network processors (e.g., Intel IXP1200). These devices provide a hybrid solution, striding the flexibility of software and the performance of hardware. Because these devices are more general purpose, the financial risk involved in supporting IPv6 is considerably less. Other IPv6-capable semiconductors are beginning to appear in the marketplace; for example, Dallas Semiconductors has recently extended its product offerings with a number of IPv6-capable network ASICs.

Technical Issues in Deploying IPv6 Networks

We now discuss technical issues that affect the transition process from an administrative point of view and give some insight into aspects that must be considered in practical deployment. These issues include upgrading and supporting IP domain name services, deploying IPv6-capable routing protocols and address management protocols, using IPSec-based security, and achieving connectivity to wide-area IPv6 networks.

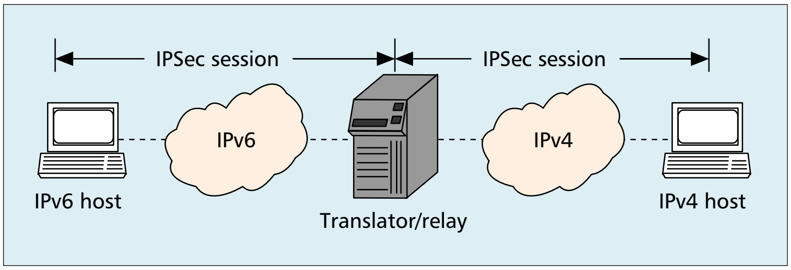
Domain Name Services

IP applications generally do not directly use IP addresses, but more user-friendly domain names. Because of the increased address size of IPv6, the use of a Domain Name Service (DNS) is even more crucial. DNS servers provide a service for remote end systems to resolve domain names to IP addresses. DNS request messages, as defined in (IETF RFC 1035), are sent via either UDP datagrams or within a TCP session. The difference between IPv6- and IPv4-capable DNS is that the former is able to handle requests from IPv6 transport layers.

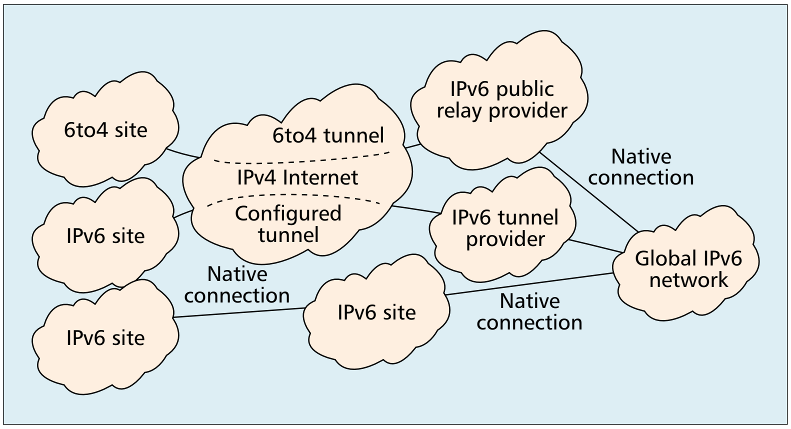
DNS servers maintain a directory of resource records (RR) mapping IP addresses to their respective domain names. 32-bit IPv4 addresses are mapped to domain names through what are known as A-records. However, because of the extended length of IPv6 addresses, the older A-records are not suitable. Instead, IETF DNS specifications define two new record types known as AAAA and A6 records (IETF RFC 2874). AAAA records simply map a domain name to a larger 128-bit address. A6 records allow the mapping of IPv6 addresses to domain names, and also the mapping of IPv6 address prefixes to partial domain names. Thus, to obtain the IPv6 address or addresses of a given name, the DNS server must obtain a complete chain of A6 records (each segment of the chain may be given by a different DNS server). The purpose of this feature is to provide the ability to change the address prefix of a given domain or subdomain by adjusting only a single record. Many issues still surround DNS for IPv6, and the currently proposed new record types are not yet completely agreed on.

Routing

Global coordination of routing across the Internet is performed by routing protocols. Autonomous systems (ASs) define the organizational breakup of infrastructural ownership. Within a given AS (intradomain), routing tables are managed through interior gateway protocols, such as RIP and Open Shortest Path First (OSPF). Outside the AS (interdomain), routing information is exchanged via exterior gateway protocols, such as EGP and BGP. Unlike the approach to combining IPv4 and IPv6 DNS services as previously discussed,



■ **Figure 7.** *Supporting IPSec across relays and translators.*



■ **Figure 8.** *Global connectivity configurations.*

IPv4 and IPv6 routing is kept separate and managed in parallel. The main reason for this is to keep the IPv6 routing tables as efficient as possible by avoiding pollution from existing IPv4 routing tables that may be poorly aggregated. We now discuss interior and exterior IP routing protocols that are typically used in IPv6 networks.

Interior Routing — Most IPv6 interior routing protocols are direct extensions from their IPv4 counterparts. In terms of suitability to IPv6 and implementation availability, OSPF (v. 3.0) (IETF RFC 2740), RIPng (IETF RFC 2080) (the name given to RIP for IPv6) and IS-IS (IETF RFC 1142) are the de facto standards. Many other proprietary routing protocols, such as Cisco's IGRP, also support IPv6, but are less widely used. Table 2 highlights new support for IPv6 in OSPF, RIP, and IS-IS.

Exterior Routing — At the conception of IPv6, it was proposed that the most widely used exterior routing protocol, the Border Gateway Protocol (BGP) (IETF RFC 1771), was too tightly coupled to IPv4 addressing and therefore would not be appropriate for the new protocol. As a result, IPv6 was positioned to use the Inter-Domain Routing Protocol (IDRP), an ISO defined standard for multiprotocol exterior routing. However, IDRP for IPv6 has never been widely deployed. The majority of router vendors preferred to use BGP (IETF RFC 1771) because of its wide recognition and extensive testing in the Internet. As a result, at the time of writing, BGP4 is the protocol of choice for IPv6 exterior routing. BGP4 includes support for multiprotocol extensions (also known as BGP4+) that enable it to directly support the extended address length and scoping requirements of IPv6 (IETF RFC 2545). Table 3 compares aspects of BGP4+ and IDRP.

DHCP and Address Configuration

A key issue in network deployment is end system address assignment. IPv6 is able to perform address auto-configuration, whereby end systems configure their own link local addresses by appending the link local prefix (FE80::/64) to their 64-bit link layer address (e.g., Ethernet MAC address). Once end systems have link local addresses, they can then configure global unicast addresses through either stateless or stateful configuration. Stateless configuration combines network prefixes advertised in router advertisements (RAs) with the 64-bit link layer addresses to form global addresses. Alternatively, stateful configuration may be used to manage address allocation via the Dynamic Host Configuration Protocol (DHCP) (IETF RFC 1541). DHCP was originally specified for IPv4 automatic address configuration. DHCPv6 defines the protocol for IPv6 [9]. The DHCP protocol can also be used to configure end system parameters other than addresses, including default gateways, name servers, and proxies.

In the absence of DHCP, configuration parameters can be inherently assumed using predefined IPv6 multicast and anycast addresses. However, this approach does not allow control of address allocation (thus permitting arbitrary end system connection) and furthermore does not support the allocation of multiple address spaces to the same physical link.

IP Security

Another important issue in network deployment is security. The IPv6 specification mandates that end systems be able to support a basic level of security conforming to the IPSec Internet security architecture (IETF RFC 2401). The hope is that by defining security support at the IP level, as opposed to application level, secure communications can be used for all applications, including nonsecure legacy applications. In brief, IPSec defines two services: the authentication header (AH) and the encapsulating security payload (ESP) for connectionless integrity, data origin authentication, and confidentiality. In addition to ESP and AH, the IPSec architecture also defines techniques for security requirements configuration, key management, and particular algorithms for authentication and encryption. One should note, however, that IPSec is not specific to IPv6 and is in fact intended for IPv4 also.

In terms of deploying IPSec in conjunction with IPv6 transitioning solutions, mechanisms that directly modify a packet will render the packet inauthentic. This problem principally arises from the use of translators and relays, such as NAT-PT and TRT. In such scenarios, IPSec cannot easily be deployed end to end. However, in theory, IPSec could be deployed between end systems and relay/translation devices, resulting in the concatenation of multiple IPSec sessions providing end-to-end security (Fig. 7). This would, however, require additional processing on the translator/relay together with an appropriate key management solution.

Alternatively, hierarchical tunnel-based transitioning solutions may be used when security is a requisite. However, encapsulating IPv6 packets within IPSec payloads, and then within IPv4 packets, does result in significant overhead, and therefore performance is likely to become degraded.

Achieving Global IPv6 Connectivity

Today, the Internet is predominantly based on IPv4. As a result, any IPv4 end system that is connected to the Internet is able to exchange packets with any other connected IPv4 end system; the network is truly global. However, there is no obvious single global IPv6 Internet on the same scale. Presently, the most globally connected IPv6 network is the 6Bone (www.6bone.net). The 6Bone to date (March 2002) interconnects over 1000 sites across the world and is rapidly becoming the de facto IPv6 Internet. Nevertheless, a large number of smaller IPv6-capable networks exist, including private research networks and commercial testbeds.

There are three preferred approaches to providing wide-area IPv6 connectivity: 6to4 automatic tunnels, configured tunnels, and native connections (Fig. 8). Because of their inherent ease of deployment, 6to4 tunnels are likely to be the first solution of choice. Address allocation is simple, and only one tunnel endpoint need be configured (i.e., through choice of relay router). However, 6to4 tunnels are often unreliable, since no contract exists between the public relay and the site's 6to4 gateway (hence, the relay provider may arbitrarily terminate its services). Furthermore, the use of 6to4 public relays often results in poor network QoS, due to uncontrolled load on a single route, and also generally inhibits the use of multicast and anycast features.

Configured tunnels offer an alternative solution. These are more difficult to manage, particularly with a view to initial deployment, but do in most cases provide better network QoS and support multicast and anycast. Configured tunnels require configuration of both end-points, i.e., between the client site and the remote tunnel provider. Once a tunnel has been set up between the provider and the client, the provider will advertise the appropriate routing information into the client's network. Contracts for configured tunnels are more feasible and because their use is more strictly controlled, QoS can be more easily assured. Nevertheless, free configured tunnel providers do exist, including Viagenie's Freenet6 service, which provides tunneling services and the allocation of up to a 48-bit network prefix (http://www.freenet6.net). Of course, free services are best effort, but load is often controlled through registration admission. Tunnels by their nature fixate a portion of the routing path between communicating IPv6 nodes. Consequently, global routing often becomes less than optimal. To avoid excess redirection, tunnels should be made as short as possible (in terms of number of hops). In deploying tunnels, both configured and automatic, the nearest adjacent tunnel provider generally provides least interference on efficient routing. In some cases multiple tunnels may be deployed within the same network, each providing an external route for a specific destination prefix (route selection is managed by an exterior routing protocol). This technique can also be used when an IPv6 site requires both global connectivity and connectivity to another "private" site across some incompatible network (i.e., an extranet).

The ultimate solution for global connectivity is a native IPv6 connection that provides a direct link to an adjacent IPv6 network. Because tunneling is not required, this solution generally leads to better performance.

Protocol	Notes
RIPng	Simple distance vector routing protocol suited to smaller networks. Routing entries consist of an IPv6 destination prefix, a metric, and an IPv6 address of the next-hop router. Existing RIPv2 protocols for IPv4 are easy to port to RIPng.
OSPF v. 3	More sophisticated link state protocol suited to larger networks. Protocol processing is per-link, not per-subnet as with IPv4, since IPv6 allows multiple IPv6 subnets to be assigned to a single link. OSPF messages are addressed through IPv6 link local address and are sent directly over IPv6 in the form of link state advertisements.
IS-IS	An ISO defined extensible intradomain routing protocol. Routes signaled through dissemination of variable-length messages. IPv6 extensions in IS-IS define two new type-length-values (TLVs) for IPv6 reachability and IPv6 interface addresses.

Table 2. *IPv6 interior routing protocols.*

Protocol	Notes
BGP4+	BGP uses both global and link local IPv6 address to announce a next hop (use of link local address conforms to the IPv6 ICMP specification). BGP messages can be transported via TCP over either IPv4 or IPv6. A BGP4 router must have at least one IPv4 address [RFC, 2283]. Implementations available in most router software suites.
IDRP	Messages exchanged directly over IPv4 or IPv6 datagrams. Designed as a multiprotocol routing protocol from the ground up and therefore has no dependencies on IPv4. Not as widely available as BGP.

Table 3. *IPv6 exterior routing protocols.*

Conclusions

It is now fairly well accepted that the arrival of IPv6 in the Internet will actually happen. Proponents admit that the progress in taking up this new protocol has been slower than was initially hoped. We believe the key reason for this is that IPv6 is *evolutionary*, not *revolutionary*. Until the Internet actually runs out of address space, or demand for security and QoS becomes more significant, IPv6 technology will be considered a luxury. Nevertheless, acceptance of IPv6 is consistently on the increase, primarily due to the realization that the problems arising in the current IPv4 Internet will need to be solved sooner or later, and that addressing these problems sooner is likely to result in less overall expense. Furthermore, IPv6 is the only real solution we have to this impending problem, gaining maturity and expanding understanding.

Much of the current drive for IPv6 is centered on Europe and Asia, while the principal IPv6 skeptic is North America. There are two likely reasons for this. The first is that Europe and Asia are the biggest sufferers of insufficient address space. North America is allocated some 70 percent of the worldwide IPv4 address space, while the massive technology-hungry populations of Europe and Asia are left with severely dwindling network addresses. The second is with respect to 3G wireless technology. Europe and Asia maintain a large market demand for mobile technologies that will likely result in increased address demands. Consequently, European and Asian 3G vendors and their respective standards bodies are significantly committed to the resolution of the address shortage problem and thus IPv6. Recently, Japan's government has mandated the incorporation of IPv6 for ISPs and has set a deadline of 2005 in which to upgrade their systems. In addition to political influences, large commercial companies are now supporting IPv6 in their products. Already millions of IPv6-capable operating systems are connected to the Internet.

Irrespective of the apparently increasing enthusiasm for IPv6, the shift to this new protocol is not going to happen overnight. Migration must be phased in order to adapt to the changing demand for IPv6 and to allow a gradual transition.

Irrespective of the apparently increasing enthusiasm for IPv6, the shift to this new protocol is not going to happen overnight. Migration must be phased in order to adapt to the changing demand for IPv6 and to allow a gradual transition. As discussed in this article, many technical issues exist in deploying IPv6. However, results from work in the IETF are now providing more feasible solutions for migration toward use of IPv6 in the Internet. From this work, we believe that the pivotal mechanisms in the transition are NAT-PT, 6to4 tunnels, and configured tunnels. These are already the most widely deployed transitioning technologies and fulfill the basic requirements for interoperability within the existing Internet. Maybe now IPv6 can move from a pipe dream to reality.

References

[1] C. Huitema, *IPv6 — The New Internet Protocol*, 2nd ed., Prentice Hall, 1997.

[2] C. Perkins and D. Johnson, "Mobile Support in IPv6," *Proc. MobiCom '96*, Nov. 1996.

[3] S. Lee *et al.*, "Dual Stack Hosts Using Bump-in-the-API (BIA)," IETF draft, draft-sylee-bia-00.txt, Feb. 2001.

[4] H. Tsuchiya *et al.*, "An IPv6/IPv4 Multicast Translator based on IGMP/MLD Proxying," IETF draft, draft-ietf-ngtrans-mtp-00.txt, May 2001.

[5] J. Hagino and K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator (TRT)," IETF draft, draft-ietf-ngtrans-tcpudp-relay-03.txt, Apr. 2001.

[6] H. Kitamura, "A SOCKS-based IPv6/IPv4 Gateway Mechanism, IETF draft, draft-ietf-ngtrans-socks-gateway-06.txt, Mar. 2001.

[7] F. Templin, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)," IETF draft, draft-ietf-ngtrans-isatap-00.txt, Mar. 2000.

[8] J. Bound *et al.*, "Dual Stack Transition Mechanism (DSTM)," IETF draft, draft-ietf-ngtrans-dstm-04.txt, Feb. 2001.

[9] J. Bound *et al.*, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," IETF draft, draft-ietf-dhc-dhcpv6-18.txt, Apr. 2001.

Note: For IETF RFCs references throughout this article see http://www.ietf.org/rfc/

Additional Reading

[1] C. Perkins and J. Bound, "DHCP for IPv6," *Proc. 3rd IEEE Symp. Comp. and Commun.*, Athens, Greece, June 1998.

Biographies

Daniel G. Waddington (dwaddington@lucent.com) received a B.Sc. Hons. computing degree in 1995 and a Ph.D. degree in 2000, both from Lancaster University, England. During his Ph.D. he worked as a research scientist in the British Telecom URI project on the Management of Multiservice Networks. In 2000 he joined Bell Laboratories, Lucent Technologies, Holmdel, New Jersey, where he is currently working in the field of IPv6 with a principal focus on network topology discovery and evolutionary analysis.

Fangzhe Chang (fangzhe@lucent.com) is a member of technical staff at Bell Laboratories, Lucent Technologies, Holmdel, New Jersey. His current research focuses on network services and management, especially discovery and analysis of network topology and issues in transitioning to IPv6 networks. He received his Bachelor's degree from the Changsha Institute of Technology, his M.Eng. from the Institute of Software, Academia Sinica, and his Ph.D. from the Courant Institute of Mathematical Sciences, New York University.