



WHATEVER HAPPENED TO THE *Next-Generation Internet?*

Despite its performance benefits, long history of development, and well-heeled advocates, IPv6 may never replace IPv4.

Mark Weiser

★ A NEW INTERNET PROTOCOL (IP) IS OSTENSIBLY ON THE HORIZON — where it's been for the past decade. Among the rapidly developing hardware and software technology around it, the current protocol is very much an anomaly. IP version 4 (v4), the 1981 U.S. Department of Defense standard defined by RFC791 [8], was originally developed to support a limited backbone linking at most dozens of networks in an open government and academic research environment. It now supports more than 60 million nodes on the global Internet.

In the early 1990s, users, vendors, and researchers became concerned that a major overhaul would be needed to accommodate the Internet's expected growth. The harshest critics forecast that the 32-bit v4 address space would be exhausted by 1996, compelling an immediate move to the "next generation" of the Internet Protocol (IPng). With over 70% of the traffic generated by commercial applications on today's privatized Internet, reports calling for guaranteed or improved quality of service and security, as well as a range of other enhancements, have made the move to IPng seem an absolute imperative for the

Internet's survival. Thus motivated, the Internet Engineering Task Force (IETF, www.ietf.org) has been moving to develop IP version 6 (v6) [6] while continuing to refine both the basic protocol and a suite of companion specifications. Earlier this year, Cisco Systems, a communications gear vendor that plays a leading role in protocol development, even released v6 as a free upgrade for most of its routers.

Despite apparently compelling arguments for protocol improvement and the increasing availability of v6, the world is still nicely served by v4. There is no apocalypse looming on the immediate horizon

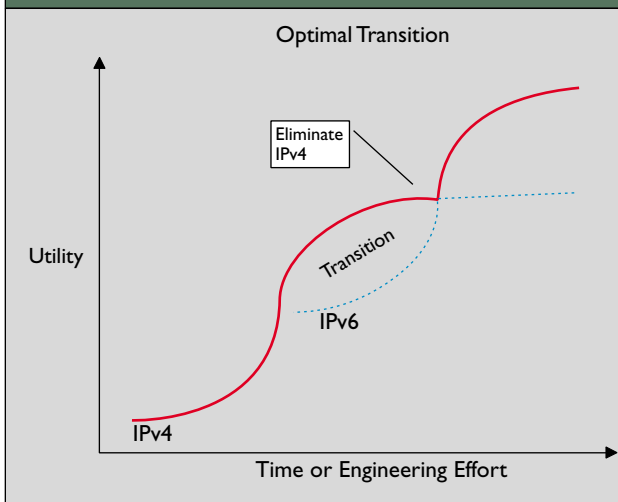
threatening to bring down the Internet. In fact, due to past and current allocation schemes and to the distributed nature of address assignment, there is no way to accurately determine how many addresses have been allocated (reserved) or assigned or when addresses will actually be exhausted. Many experts now conservatively forecast the absolute barrier for address exhaustion sometime after 2010. Meanwhile, effective network-transmission speeds have increased, and products with enhanced quality of service and security over v4 have been introduced by a number of vendors.

Few would argue that the features of v6 are not superior to v4, though most current transition proposals include a lengthy and expensive conversion that would increase network overhead and complexity. Many feel that extensions to the current Internet's

Despite apparently compelling arguments for protocol improvement and the increasing availability of v6, THE WORLD IS STILL NICELY SERVED BY V4.

functionality have raised the network's performance capability to a point where it supports expected applications almost as well as the apparent successor protocol, making a difficult transition that much more difficult to justify. Even John Klensin, chairman of the Internet Architecture Board (www.iab.org), a technical advisory group of the Internet Society, has said that all v6 really solves, relative to the augmented and improved v4, is address exhaustion. However, the life expectancy of v4 and its ability to support the global Internet's future requirements remain in doubt. Technologists must therefore face the transition issue quickly and either plan for a major protocol conver-

Figure 1. The expected optimum transition is when a radical new technology (IPv6) can be developed and transitioned to, prior to the collapse of the original technology (IPv4) [5].



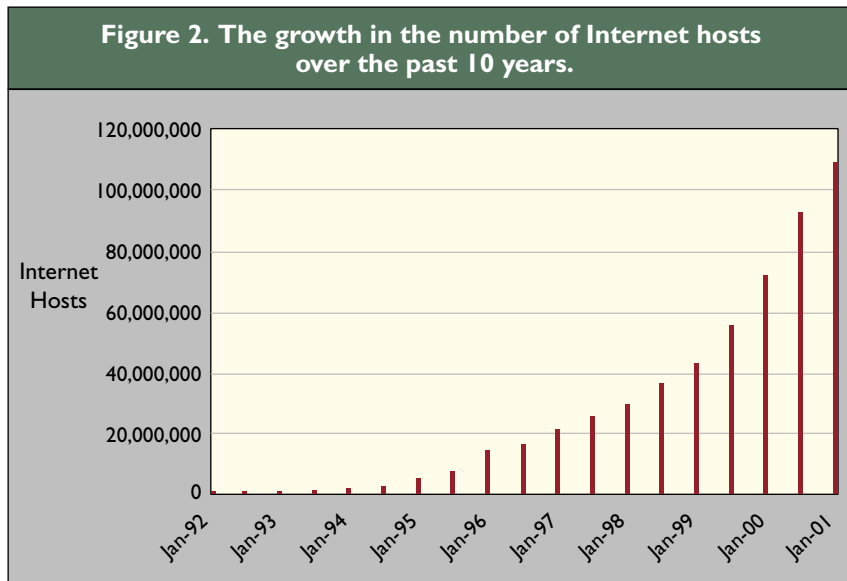
sion or develop and acquire the necessary suite of additions to v4 to deliver the functionality and longevity needed for stable worldwide data communications past the next decade.

Here, I outline the major forces influencing transition from v4 to v6. I then describe the three most critical enhancements in v6 that purportedly make it far superior to v4, as well as the incremental changes to v4 that bring its applied potential close to that of v6. Lastly, I explore modes of transition, including those allowing a lengthy extension to v4 until full transition to v6, or even to some successor, becomes necessary.

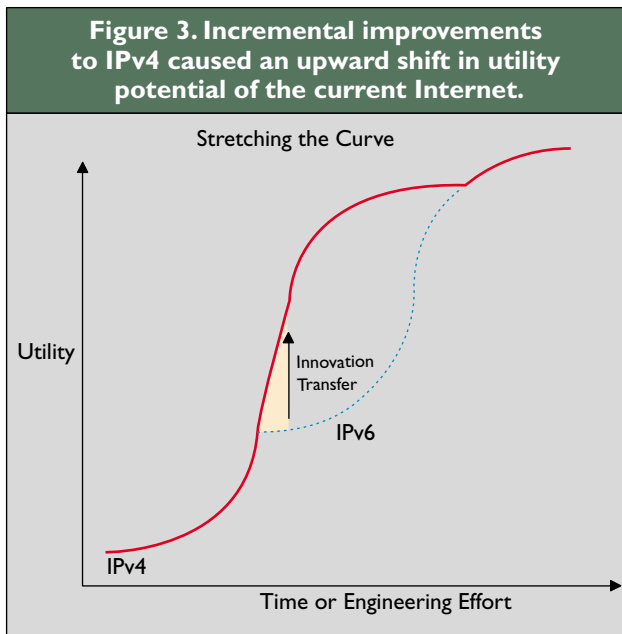
Forces of Transition

The “S-curve” concept is frequently used to explain technology life cycles in a competitive environment, including introduction of new, and retirement of obsolete, items. Rather than weighing the high costs of research and development against the expected revenues new products may bring through this model, I compare the high cost of an IP transition against the expected increase in utility the new implemented protocol would bring to our Internet-based applications (see Figure 1).

In 1988, when government restrictions were loosened, the Internet became accessible for limited commercial use. The resulting financial implications encouraged greater emphasis on application development. The Web was introduced in 1991, followed by National Science Foundation contracts with private organizations to provide access points to the T3 links of the 1992 Internet. Mosaic introduced the first graphical browsing capability in 1993. We can view these steps as incremental engineering inputs to the



Source: ISC Internet Domain Survey www.isc.org/ds/



Internet, though the basic underlying protocol did not change substantially. The increasing body of Internet-related information brought extensive potential benefits to all kinds of businesses worldwide, as well as to individual users, as indicated in Figure 1 by the rapid increase in the v4 curve. In 1993 and 1994, the number of hosts (most in the U.S.) began increasing rapidly as a result of the network's expected utility (see Figure 2). By 1994, however, concern that the entire address space would be exhausted in two to six years raised expectations in the minds of network planners, developers, and researchers that the inflection point of the S-curve had already been passed and v4's technical limit was fast approaching. More than 20 years ago, economic theorist Devendra Sahal suggested that the only way to maintain the pace of progress in develop-

ing any product or technology (in the face of minimal incremental returns on time or engineering effort) is through a radical redefinition to better match the application scope [9]. This concept helped prompt development of the next-generation Internet, well before v4 would have been defunct.

Since 1991, the IETF has intended v6 to be a revolutionary, rather than evolutionary, change incorporating v4 strengths, eliminating major v4 weaknesses in address size, and adding enhancements to facilitate secure business transactions on the public network, as well as quality-critical

multimedia applications.

The result has been a series of interim solutions to the absolute problem of address exhaustion, stretching the utility of v4 until v6 is widely available. But many of the revolutionary developments in v6 security and quality of service have been found to be applicable directly as powerful evolutionary changes to v4. Thus, the v4-to-v6 technology transfer, coupled with more permanent addressing solutions, has caused an apparent shift in the v4 S-curve, further delaying the transition to the new protocol by increasing v4's potential utility (see Figure 3).

Debatable is the placement of the v6 curve with respect to the inflection of the v4 curve and the relationship between the apexes of the curves. The three major recognized critical areas for which improvements were built into the new IP are scalability, security, and quality of service. Depending on how important each is to a particular user, users may assess their positions on the technology S-curve much differently from other users with other priorities. Analyzing each of these areas and the level at which v4 and v6 meet users' needs should help guide individual v6 adoption schedules.

Scalability. Each Internet address must uniquely identify a particular host interface and provide enough information to route a packet from anywhere on the network to that interface. The v4 packet header includes a 32-bit address field providing approximately 4.3 billion unique bit combinations with useable addresses. These addresses are divided into three classifications: 126 class-A networks, each including about 16.8 million unique addresses; 16,382 class-B networks, each supporting 56,535 addresses; and two million class-C networks, each supporting 254 hosts. The remaining addresses are

reserved for other uses, thus decreasing the theoretical limit to about 3.7 billion addresses.

The division of networks into classes originally constrained the size of core router tables, allowing faster packet forwarding. Unfortunately, once a network is allocated to an organization, the entire address space is unavailable for assignment anywhere else. For instance, as one of the first universities on the Internet, Stanford University was allocated a 16.8-million-host class-A network to support a population that today includes fewer than 70,000 hosts—an efficiency of less than 0.4%. A sample of other networks in 1994 [10], estimated that only 2.46% of the addresses in allocated class-B spaces and only 10.9% of the class-C addresses were assigned. At these efficiency rates, the fully allocated address space would support only slightly more than 100 million hosts worldwide—a number that was expected to be reached between 1996 and 2005, and was, in fact, surpassed by the end of 2000.

The principal v6 enhancement is a 128-bit addressing scheme; a total of $3.4 \times 1,038$ unique bit combinations can be made with this size field. However, just as in v4, not all of the 2,128 bit combinations are designed to be available as point-to-point addresses. Of these bits, 11 are either reserved or used to identify the packet as unicast. Individual organizations control 80 bits for creating their own addressing hierarchies and identifying individual interfaces. A 12-bit Top-Level Aggregation Identifier (TLA ID) field identifies individual core providers and limits the backbone router tables to 8,192 entries. These segmentations reduce the theoretical address space to approximately $1.16 \times 1,038$ addresses. Each of the core providers can then assign the Next-Level Aggregation Identifier (NLA ID), creating their own hierarchies and identifying individual networks served by their access points. Even assuming complete assignment of TLA IDs, a 75% efficient allocation of NLA IDs, and a reasonable 0.5% assignment efficiency by organizations, v6 would still support more than $6 \times 1,032$ unique hosts—far more than anyone expects for worldwide deployment of the new protocol.

Although address exhaustion is not a near-term issue with v6, its implementation schedule is still uncertain, creating a need for interim solutions extending the lifetime of the fixed 32-bit v4 address space. Over the past decade, the IETF has applied a series of solutions, including tighter policies on network allocation, reclaiming allocated but unassigned addresses, Classless Interdomain Routing (CIDR), and Network Address Translation (NAT).

In 1990, the Internet's Network Information Center (InterNIC, www.internic.org), which then adminis-

tered network number allocation, changed its policies to preserve the existing address space by allocating more appropriately sized networks. Since then, these policies have greatly increased the assignment efficiency of all assigned networks. Because a large portion of the address space was allocated inefficiently before 1990, much of it has had to be reclaimed in exchange for networks of the next-smaller class. For example, Stanford University relinquished its Class-A network in 2000, consolidating to four Class-B networks.

CIDR links contiguous networks of a single class, allowing for routing multiple bit-wise adjacent networks through a single network prefix. The reverse is also true; a large network, such as a class-A, could be split, with a portion assigned to multiple entities, each using the final bits of the network to uniquely identify its subnetwork. Unfortunately, this classless assignment complicates packet routing and increases routing

A single node that
migrates entirely to
the new protocol
**REMOVES
ITSELF FROM
CONNECTIVITY**
with the rest of
the network
running v4.

table size, with core nodes already handling in excess of 50,000 entries. The network portion of the address, which used to be known implicitly by its class, now has to be specified explicitly as a network mask.

These v4-enhancing policies and protocols collectively push the practical limit of the address space much higher. However, NAT breaks even the theoretical limit of 3.7 billion addresses by allowing a single public address to be used by multiple hosts. Although RFC1631 in 1994 proposed the NAT idea, implementation only recently became prevalent as a result of the address crunch and an improved protocol version,

defined by RFC3022 [11]. The technique allows a Network Address Translator to act as a router between addresses known only to the internal organization and to valid public addresses. The dynamic allocation of Internet addresses to internal host packets makes it possible for an entire network to be supported by a handful of addresses or potentially even by a single address. By using unique transport layer identities, each session between an internal and an external host can be properly identified and routed, without additional configuration on either end of the communication. Wide NAT implementation virtually removes the theoretical bounds of the 32-bit v4 address space.

But NAT is not without its own inherent limitations. Many IP researchers and developers view address translation as a poor solution to the addressing problem because IP was designed back in 1981 to carry packets from source to destination; ever since then, protocols, application software, and routers have been designed around this fundamental principal. Some applications prior to NAT may thus require an Application Layer Gateway (ALG) for processing. Some security models require that addresses and ports remain unmodified, while others, such as some IP Security Architecture (IPsec) models and upper-layer encryption, explicitly prevent further modification. One or more translation servers between the source and the destination host could create problems whose scope is not yet fully known.

These NAT-related difficulties spawned Realm Specific IP (RSIP) [2], introduced by 3Com to the IETF in 1998, as a potential replacement for NAT; it allows end-to-end IPsec through an RSIP/NAT gateway—something impossible with NAT alone. RSIP provides more end-to-end transparency for packets in transit but requires the configuration of hosts, unlike NAT. When a client needs to create a connection, the RSIP server allocates logical resources, such as addresses and ports, to the client. These resources may be used for the duration of a connection with an external host and then returned to the pool of available RSIP resources. NAT difficulties are avoided because the address and the ports are valid from source to destination.

An RSIP server also provides basic NAT functionality for any hosts not configured for RSIP. In 1991, the IETF decided to develop RSIP as a migration path to v6, as well as a backup plan, should transition to v6 fail [7]. This plan was the IETF's first indication that complete migration to v6 might never actually take place or may be delayed for an extended period.

Because automatic configuration is necessary for managing a potentially large address space, it can also be viewed as a scale issue. There are some proprietary

operating system-specific solutions for v4 in Windows98 and Macintosh systems, as well as in the standard Dynamic Host Configuration Protocol (DHCP). V6 natively allows for stateless autoconfiguration; a v6 node can combine a network prefix it learns from a local router with its layer-2 Media Access Control (MAC) address. This built-in functionality greatly simplifies assignment with a complex address space and is touted by transition proponents as a major v6 advantage. However, because the MAC address is typically 48 or 64 bits, up to 264 addresses in each autoconfigured network must be reserved to ensure uniqueness for possible future nodes. The loss of large areas of v6 address space is a high cost for simplified configuration. As such, the IETF developed a v6 version of DHCP to allow stateful autoconfiguration in a manner similar to v4.

Security. The v4 specification does not explicitly include security. The early Internet was intended for information sharing between the few connected parties, and network-layer security was not required for most related communication. Any necessary security was applied at the upper layers, prior to transporting packets.

Increasing commercial use and a heightened awareness of electronic transmission threats have prompted development of security mechanisms that could be applied universally to all Internet transmissions. The IPsec, required by the v6 specification, meets many basic security needs at the network layer, allowing for virtual private networks and other applications. Authenticated and encrypted connections are made possible by IPsec's three principal components: Authentication Header (AH), Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE).

AH provides integrity and authentication to IP datagrams, authenticating as much of an IP datagram as possible, including the source, destination addresses, and payload. ESP includes authentication of the payload information and extension headers, but can also encrypt the payload data. AH and ESP can each be used in either transport or tunnel mode. Transport mode is used only by source and destination hosts, not intermediate routers. The appropriate additional header is computed and placed immediately after the original IP header data. With ESP, a trailer is also added at the end of the payload. An intermediate router may use tunnel mode, concealing the actual source and destination information from the original packet, further strengthening security. AH-ESP combinations are also possible, employing both transport and tunnel modes. Moreover, different endpoints can be used for each association, creating a range of security combinations meeting most users'

Because Internet providers are likely to be slow to make the transition, a mechanism is needed to allow islands of v6 hosts to **COMMUNICATE ACROSS OCEANS** of v4 connectivity.

implementation needs.

Although v6 requires support of IPsec by all related network nodes, each of the mechanisms just described may also be employed by v4 systems. If a given application requires network-layer security, an IPsec implementation in v4 can provide it in the same manner as v6. Despite its availability in v4, few networks have adopted IPsec, opting instead for existing application-layer security mechanisms and raising doubts about the market's need for this feature.

Service quality. As the user population and number of applications on the Internet have increased, transmission traffic has increasingly involved multimedia file types. Unfortunately, congestion at different points in the network creates variable packet delays. Sufficient buffering at destination hosts can alleviate many of the problems of traffic rate differences that would otherwise interrupt a video or audio stream. But real-time applications cannot handle a delay, so the network has to provide a method for delivering a near-constant-rate stream of packets as needed for any application.

The v6 header contains two fields supporting priority routing of special traffic types. A 4-bit priority field allows applications to specify a certain priority for the traffic they generate, thus introducing a class-of-service concept to the Internet by labeling different packets with their priority levels. Routers are allowed to drop packets bearing priorities 0–7 during periods

of congestion but always try to forward packets with priorities 8–15. Higher-priority traffic is also forwarded ahead of lower-priority traffic, thereby decreasing network delay for time-critical applications. A flow-label field uniquely identifies a sequence of packets from a specific host. After interpreting the first packet in a message, intermediate routers provide similar handling for each packet in the session; time stamping can be added to generate almost constant delays at each router, resulting in almost constant end-to-end delays.

Although v6 has priorities built into the header, v4 has always had an 8-bit service-type field supporting six precedence levels, including “routine” to “critical” and two control priorities. Limited demand for packet prioritization and a lack of mechanisms to prevent each host from indicating that every packet is of the highest priority have prevented widespread implementation of packet priorities. Most routers, including those at the Internet core, simply ignore this field's value, and there is no reason to believe that implementation of v6 will, by itself, stimulate additional demand for quality of service.

When needed, sites with stringent prioritization requirements can turn to other existing protocols running in concert with IP, including Differentiated Services (DiffServ), Multi-Protocol label switching (MPLS), and Real-Time Transport Protocol (RTP). Some of them can further exploit priority mechanisms in link protocols, including ATM and Priority Ethernet.

Migration Mechanisms

There is little doubt that if we could go home on a Friday and come back the following Monday to find the world switched to a fully functional implementation of v6 and a suite of compatible applications, we would be at least as well off as we are now, with greater upward limits, at least in terms of address space. Unfortunately, the move to v6 is a classic chicken-and-egg dilemma. A single node that migrates entirely to the new protocol removes itself from connectivity with the rest of the network running v4. Until enough other nodes have already made the switch, there is little motivation to migrate. A migration mechanism is therefore required to allow a staged transition.

Ideally, hosts and routers would add v6 support to their existing capabilities. This “dual stack” would allow communication over v6 if the sender, receiver, and connecting network supported the protocol, and otherwise use v4 or some combination of v4 and v6. Applications incompatible with v6 could continue to operate in this mixed environment until they were upgraded, providing a smoother pathway to the new

Figure 4. Critical Internet features by protocol type.

	V4	V6	V4 with Support
Scalability • Theoretical • Practical	3.7 billion hosts 100 million hosts	1.16×10^{38} hosts 6×10^{32} hosts	Unknown Unknown
Security	None <i>Handled by upper layers</i>	IPSec (RFC 2401) • Tunnel: Entire packet • Transport: Upper-layer information	IPSec (RFC 2401) • Tunnel: Entire packet • Transport: Upper-layer information
Priorities • Basic • Other Features	7 bits in packet header • 8 precedence levels • 5 type-of-service values • 11 unused values	4 bits in packet header • “Discard-eligible” • 8 priority levels 24-bit flow label • Routers maintain state info on flow in cache Hop-by-hop extension	7 bits in packet header • 8 precedence levels • 5 type-of-service values • 11 unused values Diffserv MPLS RSVP
Auto Configuration	None	• Automated • DHCP	DHCP
Efficiencies • Per-packet overhead • Minimum packet size • Maximum payload	20B 28B 65,535B	40B 1,280B 4 billionB	20B 28B 65,535B

protocol that allows the “bugs” to be recognized and resolved before v4 is abandoned. Recognition of a destination host’s version will likely be the responsibility of the Domain Name Service, a database system that translates textual network domain names into numeric IP addresses. Depending on the type of records returned through the resolution of a host name, the source host creates a packet using the appropriate version of the protocol.

Because Internet providers are likely to be slow to make the transition, a mechanism is needed to allow islands of v6 hosts to communicate across oceans of v4 connectivity. RFC2529 [3] and Internet Draft ngtrans-6to4 [4] define “IPv6 over IPv4” and “6to4” respectively. Each of these drafts proposes a mechanism to allow isolated groups of v6 domains or hosts, attached to v4 networks, to communicate with each other. Based on v6 destination address, a supporting server encapsulates the packet inside a v4 packet and addresses it to a v4 interface on a similar system, which then translates it for another v6 domain.

However, running multiple network layer stacks at end hosts increases the complexity of processing and routing. A tunneling mechanism allows for a single network protocol at the end hosts, at the expense of a complex server that encapsulates, readdresses, and forwards packets. Each of the 6/4 tunneling mechanisms

also triples the network layer overhead from the typical 20B to 60B, including a v4 header, as well as the encapsulated v6 header.

As provider networks increasingly support v6 routing, translation servers can be pushed toward the core of the Internet. Using v6, some routes may quickly support end-to-end transmission, while others take much longer. In some cases, legacy systems may never transition from v4 to v6 due to limitations inherent in hardware or software, thus requiring perpetual translations, proxying packets over v6 alias addresses, and the relaying of packets on internal networks via v4.

Address translation for migration. Even ardent v6 supporters realize that some form of address translation will likely be prevalent [1]—to ward off address exhaustion and provide an additional layer of protection from the proxy function. These servers, in addition to extending v4’s useful lifetime, may also offer an ideal transition mechanism.

NAT servers, including those with RSIP, are already implemented in v4-based networks to forward requests between, say, a private internal addressing scheme and one or more valid public addresses. The same server may offer v6 over v4 or 6to4 tunneling, providing virtual end-to-end links for internal v6 hosts to communicate with other migrated hosts at other locations. The primary NAT benefit is that it creates an environment in which the benefits of a major transition may be derived through a series of logical component upgrades.

Conclusion

I’ve compared the benefits of v6 with both the classic v4 and the Internet’s current capabilities. I’ve outlined the distinctions, and more important, the similarities between the current Internet’s features and those expected in the proposed next-generation v6-based Internet as an aid to future research and protocol development. They represent a fairly comprehensive reference for making an informed decision about the needs of individual network systems and methods for transition (see Figure 4).

These comparisons make clear that the intended revolutionary development of v6 is only an incremen-

Coming Next Month in Communications

A special section on Aspect-Oriented Programming.

As an emerging branch of the post-object programming era, AOP is based on the notion that a computer system is better programmed by separately specifying the system's areas of interest. The articles in this section will explore all sides of AOP—its development, its applications, and its enhancement toward a complete AOP software engineering process.

AOP with adaptive methods ~
reshaping evolving software ~ using
composition filters ~ aspects in
software design ~ operating system
aspects ~ building an open AO sys-
tem ~ does AOP really work?

The October issue will also
feature an editorial discussion on
high-performance Java, including
the Ninja Project, enabling Java for
high-performance computing, and
multiparadigm communications for
grid computing.

tal improvement over the current v4-based Internet, raising questions about the viability of v6 for both the Internet's backbone and for individual organizations' intranets. A shrinking number of advocates, led by those individuals, companies, and industry consortia with major financial interests in the demise of v4 continue to proclaim the death of v4 and push for a quick change. However, there are clearly alternatives to a major transition, at least for the short-term. Many of them come at a substantially lower cost than direct implementation of v6 and fall along a likely transition path to the new protocol, should it be necessary.

Deciding whether the Internet should transition to v6 or stay with the 1981 standard is largely up to users and system managers. The market will ultimately decide when and if the expected utility of v6 (less the high transition cost) exceeds that of the current version. The Internet is not in immediate peril of collapse but may need further modification and upgrade to support novel applications. Meanwhile, a killer app may come along that will drive a transition to v6 or perhaps to an even later version of the protocol. **C**

REFERENCES

1. Barney, D. and Marsan C. Internet pioneer: IPv6 transition needed. *Network World* (Oct. 25, 1999).
2. Borella, M., Grabelsky, D., Lo, J., and Tuniguchi, K. *Realm-Specific IP: Protocol Specification*. Internet Draft (July 2000); see draft-ietf-natrsip-protocol-07.txt.
3. Carpenter, B. and Jung, C. *Transmission of IPv6 Over IPv4 Domains Without Explicit Tunnels*. RFC 2529, Mar. 1999.
4. Carpenter, B. and Moore, K. *Connection of IPv6 Domains via IPv4 Clouds Without Explicit Tunnels*. Internet Draft, Oct. 1999; see draft-ietf-ngtrans-6to4-03.txt.
5. Christensen, C. *Innovation and the General Manager*. Irwin McGraw-Hill, New York, 1999.
6. Deering, S. and Hinden, R. *Internet Protocol, Version 6 Specification*. RFC 1883, Dec. 1995.
7. Marsan, C. The next best thing to IPv6? *Network World* (Sept. 20, 1999).
8. Postel, J. *Internet Protocol: DARPA Internet Program Protocol Specification*. RFC 791, Sept. 1981.
9. Sahal, D. *Patterns of Technological Innovation*. Addison-Wesley, Reading, MA, 1981.
10. Solensky, F. *Minutes of the Address Lifetime Expectations Working Group*, July 1994; see ftp://ftp.ietf.cnri.reston.va.us/ietf-online-proceedings/94jul/area.and.wg.reports/ipng/ale/ale-minutes-94jul.txt.
11. Srisuresh, P. and Egavang, K. *Traditional IP Network Address Translator*. RFC 3022, Jan. 2001.

Current versions of all RFCs and Internet Drafts can be found on the IETF site www.ietf.org. All Internet Drafts are considered works-in-progress.

MARK WEISER (weiser@okstate.edu) is the director of the Telecommunications Management Program and the Fleming Professor of Technology Management at the College of Business Administration of Oklahoma State University, Stillwater, OK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 2001 ACM 0002-0782/01/0900 \$5.00