

ADVANCED SOFTWARE PAVES PATH TO IPv6

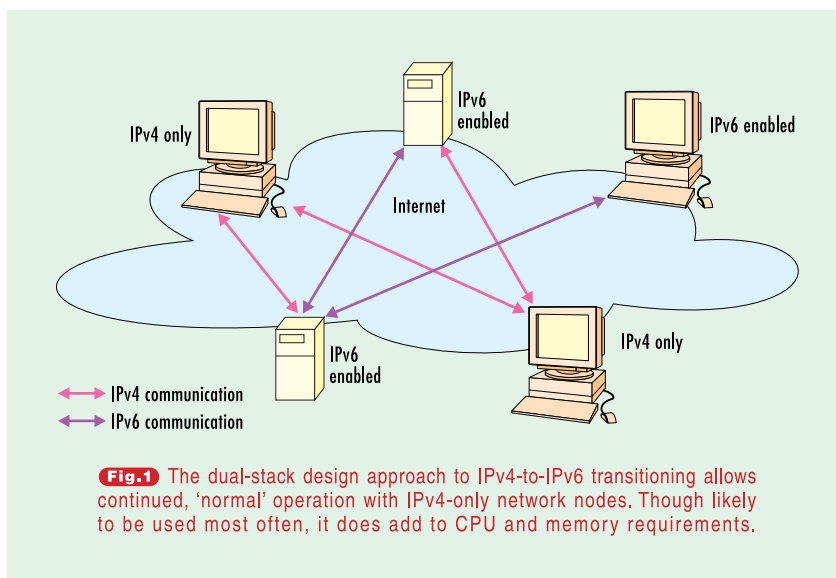
The advent of IPv4-to-IPv6 transition tools will allow designers to take advantage of the addressability and efficiencies of IPv6, thereby opening the floodgates to applications that can drive enhanced network services and, eventually, revenue.

The next two years will see a significant ramp of Internet Protocol Version 6 (IPv6) deployments. While an industrywide transition to the next-generation protocol will not be entirely complete for about a decade, 2003 to 2005 will represent the upward slope of the IPv6 implementation bell curve. So the requirement for system designers and programmers to begin enabling IP Version 4 (IPv4)-to-IPv6 transitions in their products is imminent, especially with IPv6 already making its way into infrastructure components. Operating systems supporting IPv6 are shipping from Microsoft, Sun, Hewlett-Packard/Compaq, IBM, Linux suppliers, Apple and others. In addition, IPv6-enabled file- and print-sharing applications in Microsoft Windows 2003 Server, peer-to-peer games from Microsoft and Sony and Sun Solaris applications are rolling out this year. Clearly, the industry is moving out of the early-adopter phase into commercial adoption. But, as with any core network infrastructure transition, developing the specifications for IPv6's many protocol components has been detailed and lengthy. Now with a decade of development under its belt, the IPv6 pro-

BY TONY HAIN



COVER STORY: IPv6 TRANSITION TOOLS



tool has grown mature and stable. While specifying the base protocol, Internet Engineering Task Force (IETF) working groups also defined transition tools that will enable dual-protocol coexistence—described in this article—and are currently developing guidelines for deploying and operating shared IPv4/IPv6 networks.

■ Opportunities, challenges

IPv6's 128-bit addressing space opens up new application and network service opportunities not possible with IPv4's more-limited 32-bit addressing. The expanded addressing also simplifies network operation for users of IPv6-enabled applications and network devices by alleviating the need for translation between private and public IP addresses. The translation process is often too complex for customers to configure and manage (see accompanying story, "Early Candidates for IPv6 Deployment").

And while IPv6 addresses are four times larger than IPv4's 32-bit addresses, the IPv6 header is only twice as big, and it compresses much more efficiently than IPv4. This is good news for system designers, who do not have to face a directly proportional increase in overhead when building applications and network devices to run over limited-speed network links.

Communication systems designers, in the short term, will often choose to enable peaceful coexistence of the two

protocols. This is because many of their customers and users will require legacy IPv4 infrastructures and applications to interoperate with the newer network elements. Occasionally, for installations where new capabilities have no requirement to communicate with legacy network components, building IPv6-only products will be appropriate.

IPv6's 128-bit addressing space opens new service opportunities and simplifies network operation.

To create applications supporting IPv6, host operating systems must be IPv6-enabled. System designers should be aware that today, the IPv6 capability might be available in separate IPv4 and IPv6 protocol stacks or in an integrated IPv4/IPv6 stack, depending on the host operating system. For example, at this juncture, Microsoft Windows XP and 2003 Server include separate parallel stacks, while recent versions of FreeBSD Unix, HP's True64 Unix, Linux and Sun Solaris support integrated stacks.

To minimize disruption during transitions from IPv4 to IPv6, the IETF's Next-Generation Transition (NGTRANS) Working Group has specified tools that enable designers and developers to begin moving network infrastructure segments and applications—or both—to the newer

protocol. The primary transition tool is called dual stack, which enables both protocols to run alongside one another simultaneously. Other tools include several network-tunneling options and translation.

The primary goal of these options is to decouple the deployment dependencies between the applications and the network infrastructure.

■ Dual stack

The dual-stack approach is likely to be used most often. Simply described, it involves running both IPv4 and IPv6, together, in part or all of a network. Network components can continue to talk to IPv4-only devices (see Fig. 1), but gain new functions with IPv6.

Dual stack can be compared with building an eight-lane freeway alongside an older two-lane highway that has been in place for decades. Adding a new transport capability to run in parallel to the one already there without ripping anything out avoids disruption. Over time, the "vehicles" (packets) can move from one infrastructure to another, until it makes economic sense to discontinue maintaining the older, two-lane highway (the IPv4 infrastructure).

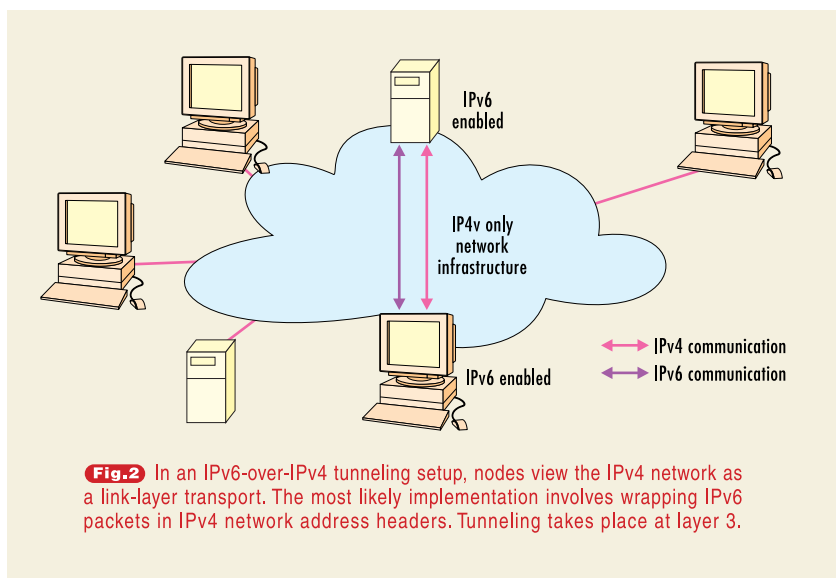
Running two routing and communication infrastructures does carry a price in terms of additional CPU and memory consumption. Designers creating products targeting dual-stack environments must keep the resource requirement issue in mind. However, the payoff of providing sufficient CPU and memory to operate both protocols concurrently will be consumer acceptance of operationally simple devices.

■ Tunneling

This is a good point at which to clarify the difference between tunneling and encapsulation. The industry has developed a terminology distinction between tunneling and encapsulation. Tunneling takes place at Layer 3 (the routing layer) and above, while encapsulation occurs at Layer 2 (the link layer). The mechanisms are architecturally the same, however. They involve "wrapping" one type of packet inside another to enable a transmission stream to traverse a dissimilar network infrastructure.

There are several types of tunneling, but each allows the network imple-

COVER STORY: IPv6 TRANSITION TOOLS



menter to decouple the deployment dependencies between the applications and the network infrastructure by running one network “over” another. Tunneling techniques can be used alone or in conjunction with dual stack. Regardless of the tunneling method used, this approach requires at least one pair of dual-stack nodes somewhere in the network.

IPv6-over-IPv4 tunnels: The most common implementation of tunneling is likely to involve wrapping IPv6 packets in IPv4 network address headers, as many network implementers will start with pockets of IPv6 and leave the majority of the IPv4 network infrastructure in place (see Fig. 2).

Depending on the situation, however,

the opposite could happen: the majority of a network could migrate to IPv6 with a few legacy IPv4 nodes retained. In this case, IPv4-over-IPv6 tunneling would be more appropriate. To accommodate customers planning on this course of action, system designers will want to provide IPv6 support in intelligent devices at the very edge of the network, such as in user devices and applications.

Connection of IPv6 domains via IPv4 clouds (6 to 4): Targeted at deployments between isolated sites, this mechanism allows an IPv6 router to automatically tunnel packets across the IPv4 network using a single IPv4 address to create an IPv6 prefix.

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP): ISATAP targets

campus deployments and allows dual-stack nodes that do not share a physical link with an IPv6 router to automatically tunnel packets to the IPv6 next-hop address through IPv4.

Teredo: This option encapsulates IPv6 in the User Datagram Protocol (UDP) over IPv4. Its objective is to traverse IPv4 Network Address Translation (NAT). The UDP transport protocol establishes a “hole” in NAT to enable the tunneling. This tool is best applied to the development of consumer applications that require high levels of simplicity, such as multiplayer gaming.

IPv6 over Multiprotocol Label Switching (MPLS) backbones: Similar to IPv6-over-IPv4 tunneling, this tool allows isolated IPv6 domains to communicate with each other over an MPLS backbone. The impact here is for designers building provider edge (PE) routers at the perimeter of service-provider networks. These devices will have to recognize IPv6 addresses and formats and put the appropriate MPLS label on this new address format for proper switching across the MPLS backbone.

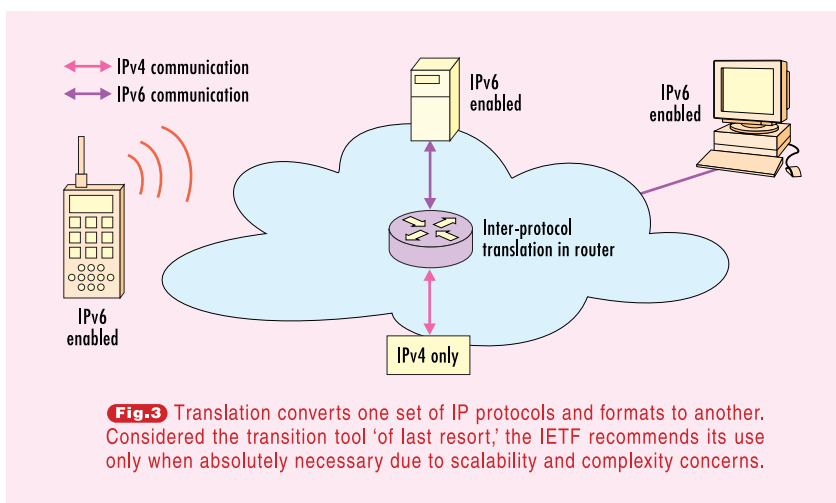
Translation

This transition “tool of last resort” converts one set of IP protocols and formats to another. The IETF recommendation is to use translation only when absolutely necessary; in most cases, it works no better than IPv4 with NAT.

The reason that translation is not recommended for frequent use is that it raises scaling concerns and causes complexity in applications. At the same time, translation tools are important in networks where some devices run IPv6 only while others support IPv4 only—leaving, in effect, islands of functionality. For example, if a new portable device includes only IPv6 but shared devices such as printers have not been upgraded, translation will be required to enable them to interoperate (see Fig. 3).

There are several versions of translators, which work at different layers. Specifically, these include API translators, transport-layer translators and IP packet translators. The IP packet translators can be stateful or stateless, depending on the nature of the session being translated.

As mentioned, masking complexity with a plug-and-play transition is a design



goal of IPv6. Ideally, using the tools described, any device should be able to plug into the network and, with minimal configuration, begin sending packets.

Subnet services: In the IPv4 environment, a service provider's ability to offer enhanced services was limited to business customers with technically astute IT and network staffs. This was due to the challenges associated with configuring and managing the limited-address space or the NAT process.

■ Service provisioning

While IPv6 enables this type of service without asking much from the customer, the development community would need to include several characteristics in service-provider and customer devices. For example, the network service provider's premises equipment router must be able to do the following:

- Allocate network prefixes to customers using Dynamic Host Control Protocol Version 6 (DHCPv6), which is currently an IETF draft; and
 - Install routes to those prefixes.
- At the other end, the customer's network must be able to do the following:
- Acquire the prefix;
 - Announce the prefix to downstream devices; and
 - Install a default route.

Address Privacy: A feature that mobile users will find valuable is the privacy address capability defined in RFC 3041. Because interfaces in IPv6 networks support multiple IP addresses, user devices can use different interface identifiers over time on a visited subnet. This greatly reduces the ability of contacted nodes to profile the subnets to which a specific device attaches—in other words, it makes mobile devices less traceable. The configuration underpinnings of this capability are masked from the consumer.

From a design perspective, IPv6 nodes will have multiple entries in the neighbor cache, which will require more memory.

■ Other considerations

As indicated in the section above, the typical interface in an IPv4 network is assigned only one address, but in IPv6 it must support multiple addresses. While this adds some complexity to the system designer's task, it greatly simplifies the operational characteristics of devices.

Something else designers should keep in mind is that IPv4 can pass around a 32-bit address on a 32-bit bus in one clock cycle, while two clock cycles will be required for distributing a 128-bit IPv6 address on a system with a 64-bit bus.

With the large address space per subnet, there is no need to continually re-size pools to match current demand, reducing operational complexity for the network implementer. Hotels and wireless hotspots, for example, won't run short of space in their DHCP pools, which now happens when a large number of devices vie for temporary addresses.

Routing protocol considerations: Border Gateway Protocol Version 4 and Intermediate System-to-Intermediate System support both versions of the IP protocol in a common instance. However, the routing protocol Open Shortest Path First (OSPF) Versions 2 and 3 run in ships-in-the-night mode, in that each operates independently and is unaware of each other. So, in OSPF networks, designers should consider that every event will cause routing devices to run two calculations instead of one. This means routing convergence time is likely to increase unless additional CPU resources are added. ■

For more on IPv6 transitioning, see: "MPLS, Ipv6 Bring Integrated Services Closer"; www.commsdesign.com/story/OEG20011121S0041

"Tunneling Solves IP Traffic Snarls"; www.commsdesign.com/story/OEG20021101S0054

Further reading:

- IETF IPv6 Working Group: www.ietf.org/html.charters/ipv6-charter.html

- IPv6 Resources: <http://playground.sun.com/ipng/ipng-main.html>; www.ipv6forum.com: www.v6pc.jp/en/index.html

Tony Hain (thain@cisco.com) is a technical leader for Cisco Systems Advanced Architecture Group, where he specializes in IPv6 technology and product development. He is technology director of the IPv6 Forum's North American Task Force Steering Committee and co-chairman of the IETF NGTRANS Working Group that developed the IPv6 transition tools discussed in this article.

Applications to drive IPv6, not mobility

Heavy marketing by the mobile-device community has led to the belief that the proliferation of 3G mobile devices will drive the use of Internet Protocol version 6, for several reasons. These include predictions that mobile-phone deployments will reach more than 1.5 billion worldwide by 2005, that most of those phones will be data-enabled and that Third-Generation Partnership Project-2 (3GPP2) standards will mandate the use of IPv6 in 3G devices.

3G will certainly be a large IPv6 driver, but its near-term impact is in question due to cost and spectrum concerns that have led to scaled-back deployment timetables.

Instead, near-term IPv6 proliferation will be fueled by applications that do not perform well or are too difficult for users to manage when deployed over IPv4 in combination with network address translation (NAT) and network address port translation (NAPT).

Managing the translation tables in NAT/NAPT is common in large enterprises among IP network service providers with network-savvy staffs. But such complexity is generally beyond the skill set of most consumers.

Continued use of IPv4 with NAT/NAPT also enforces the client/server model of computing, in which a server in the public network hosts applications such as e-mail and Web browsing. But music sharing, gaming, voice-over-IP conferencing and other emerging peer-to-peer applications require higher performance.

IPv6 communication allows any node to contact any other node directly, without having to traverse a central server, thereby reducing latency and unpredictability. ■