The Internet Protocol Journal

March 2000 Volume 3, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Work on a new version of the Internet Protocol, known as IPv6, has been under way for several years in the IETF. There is still some debate about when and how IPv6 will be deployed. Proponents of IPv6 argue that the demand for new IP addresses will continue to rise to a point where we will simply run out of available IPv4 addresses and that we should, therefore, start deploying IPv6 *today*. Opponents argue that such a protocol transition will be too costly and painful for most organizations. They also argue that careful address management and the use of *Network Address Translation* (NAT) will allow continued use of the IPv4 address space for a very long time. Regardless of the timeframe, a major factor in the deployment of IPv6 is an appropriate transition strategy that allows existing IPv4 systems to communicate with new IPv6 systems. A transition mechanism, known as "6to4," is described in our first article by Brian Carpenter, Keith Moore, and Bob Fink.

In previous editions of this journal, we have looked at various security technologies for use in the Internet. Security mechanisms have been added at every layer of the protocol stack, and IP itself is no exception. IP Security, commonly known as "IPSec," is being deployed in many public and private networks. In our second article, William Stallings describes the main features of IPSec and looks at how IPSec can be used to build Virtual Private Networks.

Our final article is a critical look at *Quality of Service* (QoS) in the Internet. The need to provide different priorities to different kinds of traffic in a network is well understood and the technical community has been hard at work developing numerous systems to address this need. Geoff Huston looks at the prospects of deploying QoS solutions that will operate across the Internet as a whole.

The Y2K transition has been described as a "nonevent" by many. However, the lessons learned and the collaborative coordination efforts that were put in place for this transition can hopefully be used in the future. A colleague of mine had to call a plumber to his house on New Year's Eve. When he tried to pay for the repair with a credit card which had "00" as the expiration year, the plumber insisted that this meant the card was invalid. So while most systems were "Y2K compliant," this particular plumber was clearly not. Do you have a Y2K story to share? Drop us a line at **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Connecting IPv6 Routing Domains Over the IPv4 Internet

by Brian E. Carpenter, IBM & iCAIR
Keith Moore, University of Tennessee
Bob Fink, Energy Sciences Network

A next-generation Internet Protocol[1], known first as IPng and then as IPv6, has been under development by the *Internet Engineering Task Force* (IETF) for several years to replace the current Internet Protocol known as IPv4. The reasons behind the need for IPv6 are not covered here, but interested readers are encouraged to read "The Case for IPv6"[2] for this background.

Of major importance during the development of IPv6 has been how to do the transition away from IPv4, and towards IPv6. The work on transition strategies, tools, and mechanisms has been part of the basic IPv6 design effort from the beginning. The current transition efforts, taking place at the *IETF IPng Transition Working Group* (ngtrans)[3], will continue until it is clear that the transition will be successful.

These transition design efforts resulted in a basic Transition Mechanisms specification for IPv6 hosts and routers[4] that specifies the use of a Dual IP layer providing complete support for both IPv4 and IPv6 in hosts and routers, and IPv6-over-IPv4 *tunneling*, encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures.

These concepts are heavily relied on for transition from the traditional IPv4-based Internet as we know it today, to an IPv6-based Internet. It is expected that IPv4 and IPv6 will coexist for many years during this transition.

Of great concern to transition strategy planners is how to provide connectivity between IPv6-enabled end-user sites (also known as *routing domains*) when they do not yet have a reasonable (or any) choice of *Internet Service Provider* (ISP) that provides native IPv6 transport services. One way to provide IPv6 connectivity between end-user sites (when native IPv6 service does not exist) is to use IPv6-over-IPv4 encapsulation (tunneling) between them, similar to the technique currently used in the 6bone[5] IPv6 testbed network. This requires complexity for both end-user sites, and the networks providing the tunneling service (for instance, the 6bone backbone ISPs), in creating, managing, and operating manually configured tunnels.

The "6to4" transition mechanism, "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels"[6], provides a solution to the complexity problem of using manually configured tunnels by specifying a unique routing prefix for each end-user site that carries an IPv4 tunnel endpoint address.

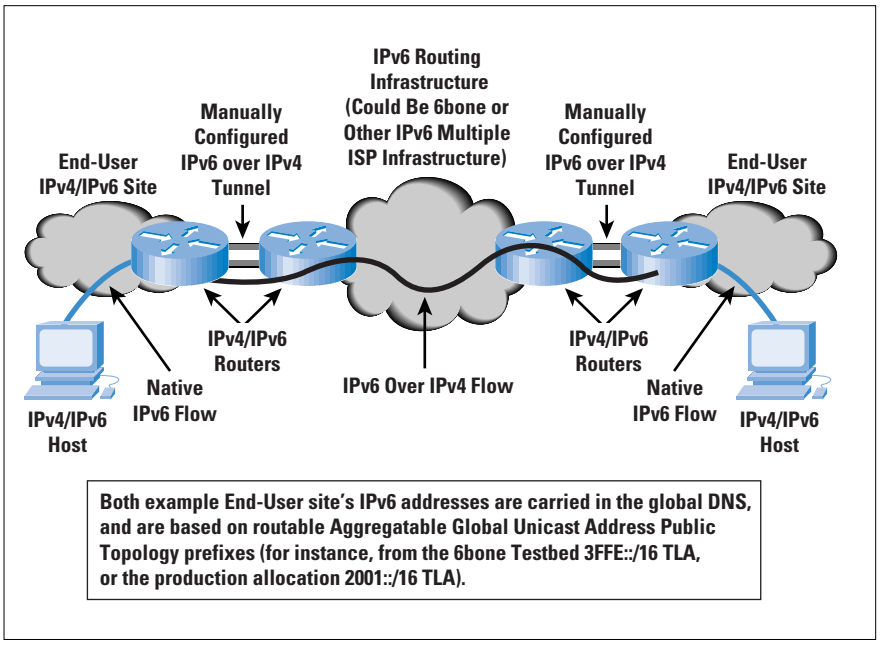
It should also be noted that each end-user site with as little as a single IPv4 address has a unique, routable, IPv6 site routing prefix thanks to the 6to4 transition mechanism.

Connecting IPv6 Routing Domains

When end-user site networks enable IPv6 in their local host and router systems, but have no native IPv6 Internet service, connectivity to other IPv6 routing domains across a worldwide Internet must be accomplished another way, or the value of a connected Internet is lost. Prior to the 6to4 transition mechanism, a site's network staff would have to rely on the manual configuration of IPv6-over-IPv4 tunnels to accomplish this connectivity.

This connectivity could be accomplished by arranging tunnels directly with each IPv6 site to which connectivity is needed, but more typically is done by arranging a tunnel into a larger IPv6 routing infrastructure that could guarantee connectivity to all IPv6 end-user site networks. (See Figure 1.) The 6bone IPv6 testbed was the first IPv6 routing infrastructure to provide worldwide IPv6 connectivity (starting in 1996), while more recently (late 1999) networks providing production IPv6 Internet service have also interconnected to provide this connectivity. In fact, the 6bone and production IPv6 routing infrastructures are well interconnected to guarantee worldwide IPv6 connectivity.

Figure 1: Configured Tunnel Overview



However, even given a solid, reliable, worldwide IPv6 routing infrastructure (similar to the IPv4-based Internet today), if an end-user site does not have a reasonable (or any) local choice for native IPv6 Internet service, a tunnel must be used.

The 6to4 mechanism addresses many of the practical difficulties with manually configured tunneling:

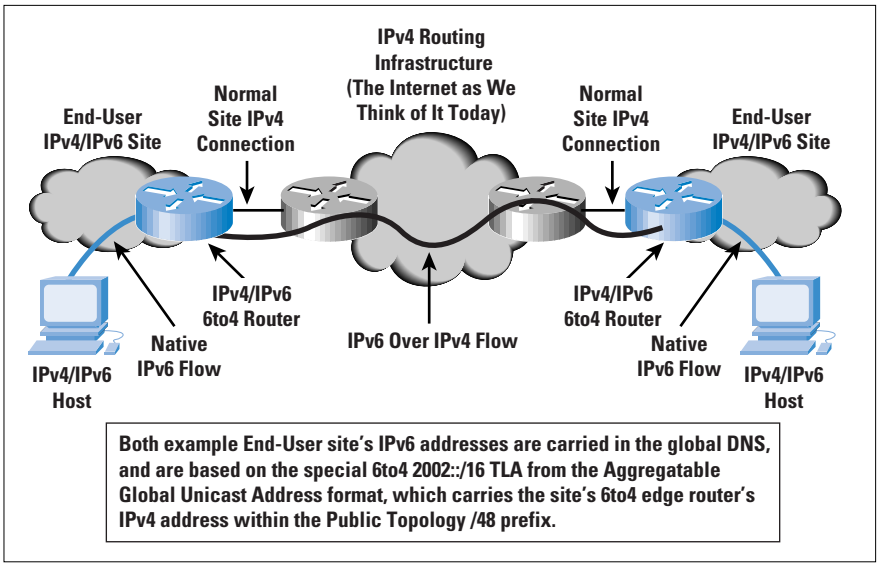
- The end-user site network staff must choose an IPv6 Internet service to tunnel to. This entails a process of at least three parts:
  - Finding candidate networks when the site's choice of IPv4 service does not provide IPv6 service (either tunneling or native),
  - Determining which ones are the best IPv4 path to use so that an IPv6-over-IPv4 tunnel doesn't inadvertently follow a very unreliable or low-performance path,
  - Making arrangements with the desired IPv6 service provider for tunneling service, a scenario that may at times be difficult if the selected provider is not willing to provide the service, or if for other administrative/cost reasons it is difficult to establish a business relationship.

Clearly it is easiest to use the site's own service provider, but in the early days of IPv6 transition this will often not be an option.

- An IPv6-over-IPv4 tunnel must be built to the selected provider, and a peering relationship must be established with the selected provider. This requires establishing a technical relationship with the provider and working through the various low-level details of how to configure tunnels between two routers, including answering the following questions:
  - Are the site and provider routers compatible early on in this process?
  - What peering protocol will be used (presumably an IPv6-capable version of the *Border Gateway Protocol Version 4* [BGP4]), and are the versions compatible and well debugged?
  - Have all the technical tunnel configuration issues between the site and provider been addressed?

Again, it is clearly easiest to perform all these steps if they are taken with the site's own IPv4 service provider.

Figure 2: 6to4 Tunnel Overview



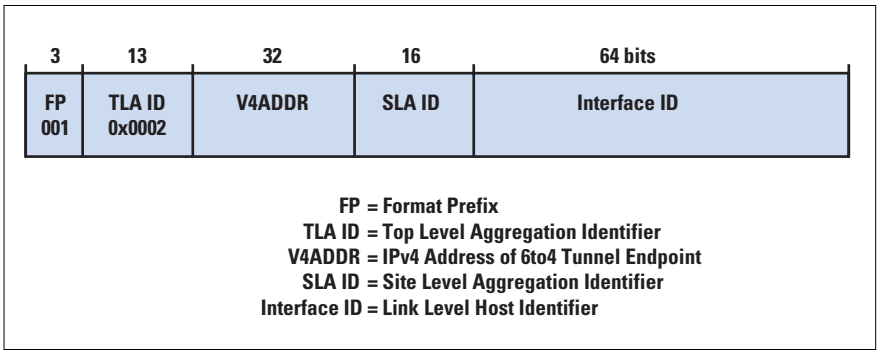
6to4 Eliminates Complex Tunnel Management

The 6to4 transition mechanism provides a solution to the complexity problem of building manually configured tunnels to an ISP by advertising a site's IPv4 tunnel endpoint (to be used for a dynamic tunnel) in a special external routing prefix for that site. Thus one site trying to reach another will discover the 6to4 tunnel endpoint from a *Domain Name System* (DNS) name to address lookup and use a dynamically built tunnel from site to site for the communication. (See Figure 2.) The tunnels are transient in that there is no state maintained for them, lasting only as long as a specific transaction uses the path. A 6to4 tunnel also bypasses the need to establish a tunnel to a wide-area IPv6 routing infrastructure, such as the 6bone.

The specification of a 48-bit external routing prefix in the IPv6 *Aggregatable Global Unicast Address Format* (AGGR)[7] (see Figure 3) that provides just enough space to hold the 32 bits required for the 32-bit IPv4 tunnel endpoint address (called V4ADDR in Figure 3) makes this setup possible.

Thus, this prefix has exactly the same format as normal prefixes assigned according to the AGGR. Within the subscriber site it can be used exactly like any other valid IPv6 prefix, for instance, for automated address assignment and discovery according to the normal IPv6 mechanisms for this.

Figure 3: 6to4 Prefix Format



The Simplest Use of 6to4

The simplest scenario for 6to4 is when several sites start to use IPv6 alongside IPv4, and have no native IPv6 ISP service available. Thus each site identifies a router to run dual stack (that is, IPv4 and IPv6 together) and 6to4 tunneling, ensuring that this router has a globally routable IPv4 address (that is, not in private IPv4 address space).

It is assumed that this new 6to4 router is reachable by IPv6-capable hosts within the site. Although the various ways in which these hosts may be reached are not discussed in detail here, they include using IPv6-enabled site IPv4 routers, operating special IPv6-only routers in parallel with site IPv4 routers, using the "6over4" mechanism[8], and employing other tunneling methods.

A new 6to4 site advertises the 6to4 prefix to its site via the *Neighbor Discovery* (ND) protocol[9], which will cause IPv6 hosts at this site to have their DNS name/address entries to include the 6to4 prefix for the site in them.

In operation, when one IPv6-enabled host at a 6to4 site tries to access an IPv6-enabled host by domain name at another 6to4 site, the DNS will return both an IPv4 and an IPv6 IP address for that host, indicating that it is reachable by both IPv4 and IPv6. The requesting host selects the IPv6 address, which will have a 6to4 prefix, and sends a packet off to its nearest router, eventually reaching its site boundary router, which we assume has 6to4 service as well.

Sending and Receiving Rules for 6to4 Routers

When the requesting site's 6to4 router sees that it must send a packet to another site (that is, there is a nonlocal destination), and that the next hop destination prefix contains the special 6to4 *Top Level Aggregation* (TLA) value of 2002::/16, the IPv6 packet is encapsulated in an IPv4 packet using an IPv4 protocol type of 41, as defined in the *Transition Mechanisms* RFC[4]. The source IPv4 address will be the one in the requesting site's 6to4 prefix (which is the IPv4 address of the outgoing interface to the Internet on the 6to4 router, and contained in the source 6to4 prefix of the IPv6 packet), and the destination IPv4 address will be the one in the next hop destination 6to4 prefix of the IPv6 packet.

When the destination site's 6to4 router receives the IPv4 packet, and recognizes that it has an IPv4 protocol type of 41, IPv4 security checks are made and the IPv4 header is removed, leaving the original IPv6 packet for local forwarding.

The sending rule above is the only modification to IPv6 forwarding, because the receiving rule was already specified for the basic IPv6 Transition Mechanism mentioned earlier[4]. Along with advertisement of the 6to4 prefix by appropriate entries in the DNS, any number of sites can interoperate without manual tunnel configuration.

It is not necessary to operate an exterior routing protocol (for instance, BGP4+) for 6to4 simple scenarios because the IPv4 exterior routing protocol is handling this function. Also, no new entries in IPv4 routing tables result from the use of 6to4.

The Return Path and Source Address Selection

Packets must flow in both directions to be useful; thus it is essential that IPv6 packets sent use a packet with a 6to4 prefix as a source address when talking to a site with a 6to4 prefix; in other words, the destination must have a 6to4 prefix. In the simple example given above, this is not an issue because both sites have only IPv4 connectivity, so they have 6to4 prefixes for their site to communicate with. DNS lookups for host systems at these sites will return only one IPv6 address, which will be the one with a 6to4 prefix. Source address selection is thus not an issue.

As we will soon see, source address selection is an issue for more complex 6to4 usage scenarios; therefore, some source address selection algorithm is necessary in IPv6 hosts. The exact form and method of the algorithm to use is under active study at the IETF IPv6 (ipng) working group[10], and an algorithm is likely to be chosen in early 2000. Meanwhile, for the purposes of understanding 6to4, it is sufficient to realize that when a 6to4 connected sending site is sending to a destination site using that site's 6to4 prefix, the sending host must guarantee that the source IPv6 address uses the sending site's 6to4 prefix.

More Complex 6to4 Usage Scenarios

Several more interesting 6to4 usage scenarios exist when a site has both 6to4 connectivity and native IPv6 connectivity. The simplest of these is when such a site is trying to reach another site that has only 6to4 connectivity, in which case the source address selection algorithm mentioned above is essential to ensure that the site's 6to4 IPv6 address is chosen. No destination selection is required because there is only one choice, that is, 6to4.

Similarly, when a site that has only 6to4 connectivity tries to reach a site with both 6to4 and native IPv6 connectivity, some host rule for choosing among multiple destination addresses must result in the 6to4 address being chosen, because only a local 6to4 IPv6 source address is available. Of course source selection is not an issue in this case because there is only the 6to4 IPv6 address to use.

Another variation of these scenarios is when a site with 6to4 and native IPv6 connectivity is trying to reach another site that has only native IPv6 connectivity, making a source address selection algorithm essential to make sure the site's native IPv6 address is chosen. No destination selection is required, because there is only one choice, that is, the native IPv6 address.

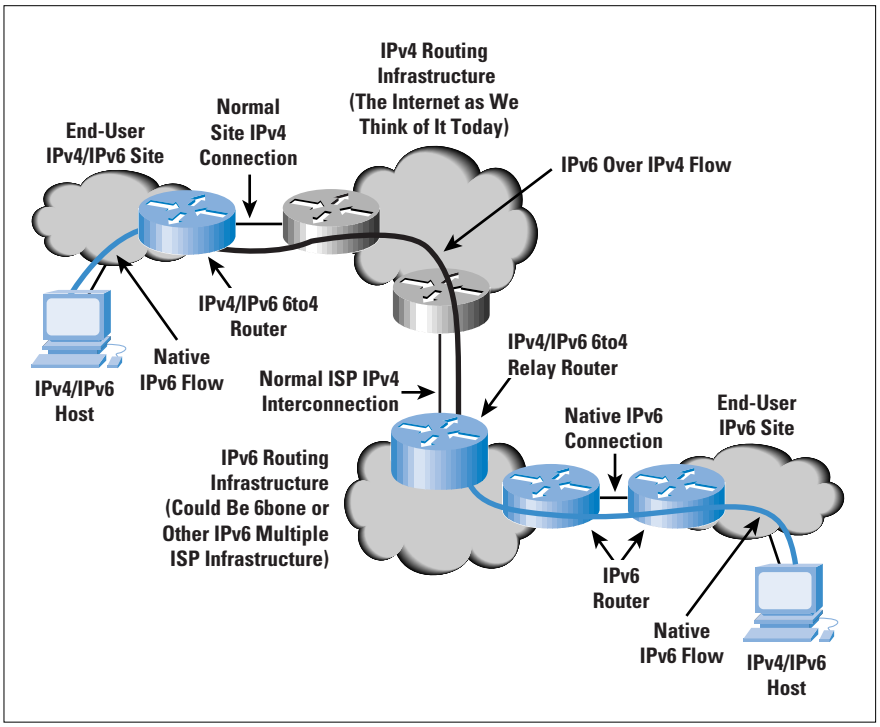
Similarly, when a site that has only native IPv6 connectivity tries to reach a site with 6to4 and native IPv6 connectivity, a host rule is essential for choosing among multiple addresses to ensure that a native IPv6 address is chosen, because only a local native IPv6 source address is available. Again, source selection is not an issue in this case because only the native IPv6 address can be used.

An interesting choice develops in the situation when both sites have 6to4 and native IPv6 connectivity as both 6to4-to-6to4 and native IPv6-to-native-IPv6 connections are a possibility. Current thinking as of the writing of this article is to prefer the native IPv6 connection.

The 6to4 Relay

The most interesting, and most complex, 6to4 scenario is that of sites with only 6to4 connectivity communicating with sites with only native IPv6 connectivity. This is accomplished by the use of a 6to4 relay that supports both 6to4 and native IPv6 connectivity (Figure 4). The 6to4 relay is nothing more than an IPv4/IPv6 dual-stack router.

Figure 4: The 6to4 Relay



The 6to4 relay advertises a route to 2002::/16 for itself into the native IPv6 infrastructure it is attached to. The native IPv6 network operators must filter out and discard any 6to4 (2002:...) prefix advertisements longer than /16. In addition, the 6to4 relay may advertise into its 6to4 connection whatever native IPv6 routes its policies allow, which the 6to4 router at the 6to4-only site picks up with either a BGP4+ peering session, or with a default route, to the 6to4 relay.

Thus the 6to4-only site will try to send a packet to the native IPv6-only site by forwarding an encapsulated (tunneled) IPv6 packet to the 6to4 relay, which removes the IPv4 header (decapsulates) and forwards the packet on to the IPv6-only site.

Potentially, multiple 6to4 relays are needed, one for each separate IPv6 routing realm (collection of IPv6 routing ISPs). In practice, it is expected that all native IPv6 ISP services will be interconnected even if the use of inter-IPv6-ISP manually configured tunnels are required to do so. This is currently the case as of early 2000, because all 6bone 3FFE::/16 TLA networks and all production 2001::/16 subTLA networks are interconnected with each other.

It is expected that native IPv6 service providers will choose to operate 6to4 relays as a simple extension of their service. There are no special rules or exceptions to 6to4 as described here for this to happen because the 6to4 relay is simply operated as part of an end-user site that belongs to the IPv6 ISP.

Other Issues

Several other 6to4 issues are presented below for completeness.

- The IPv6 *Maximum Transmission Unit* (MTU) size could prove too large for some intermediate IPv4 link when a 6to4 tunnel is in use, thus IPv4 fragmentation will occur. Though undesirable, fragmentation is not disastrous, so the IPv4 "Do Not Fragment" bit should not be set in the IPv4 packet carrying the 6to4 tunnel.
- How sites move IPv6 packets internal to a site is not important to the 6to4 process. For illustrative purposes in this article, it is generally assumed that native IPv6 transmission exists within a site. This may not be strictly true because "6over4," manual tunnels, and other methods of moving IPv6 packets could be in use. Nonetheless, it is not important to the 6to4 processes described here.
- Security issues with the 6to4 mechanism are not discussed here. The reader is referred to the current 6to4 draft for an explanation of these issues[6].
- 6to4 sites with IPv6 connectivity must not inject their 6to4 prefix into the IPv6 routing infrastructure via the native IPv6 connection.
- It is not possible to assume the general availability of wide-area IPv4 multicast, so the 6to4 mechanism must assume only unicast capability in its underlying IPv4 carrier network. However, it is expected that IPv6 multicast packets may be sent to, or sourced from, a 6to4 router in the IPv4 encapsulated form, as described above. When IPv6 multicast is supported, an IPv6 multicast routing protocol must be used.
- The use of IPv6 Anycast is compatible with 6to4 prefixes.
- 6to4 for hosts only, as opposed to sites, is possible and will likely be developed in the future. However, details of this feature are not discussed in this article.
- The 6to4 mechanism is unaffected by the presence of a firewall at the border router.
- When using IPv4 *Network Address Translation* (NAT), 6to4 mechanisms remain valid, and the NAT device includes a fully functional IPv6 router with the 6to4 mechanism included. Combining 6to4 and NAT in this way offers the advantages of NAT for IPv4 use, and the additional address space of IPv6.
- There is no significant impact to either IPv4 or IPv6 routing table size caused by the proper implementation of 6to4.

Summarizing 6to4

The 6to4 mechanism allows isolated IPv6 routing domains to communicate with other IPv6 routing domains, even in the total absence of native IPv6 service providers. It is a powerful IPv6 transition tool that will allow both traditional IPv4-based Internet end-user sites and new IPv6-only Internet sites to utilize IPv6 and operate successfully over the existing IPv4-based Internet routing infrastructure.

For Further Reading

[0] Fink, R., "IPv6—What and Where It Is," *The Internet Protocol Journal,* Volume 2, No. 1, March 1999.

[1] IPng and IPv6 information, including formal specifications, can be found at: **http://playground.sun.com/pub/ipng/html**

[2] "The Case for IPv6," an Internet Draft of the IAB, can be found at: **http://www.6bone.net/misc/case-for-ipv6.html**

[3] IETF IPv6 Transition Working Group (ngtrans) information, including status of all its current projects, can be found at: **http://www.6bone.net/ngtrans/**

[4] "Transition Mechanisms for IPv6 Hosts and Routers," RFC 1933, can be found at: **http://www.ietf.org/rfc/rfc1933.txt**

[5] The 6bone IPv6 Testbed Network is explained at: **http://www.6bone.net**

[6] "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels" ("6to4"), an Internet Draft of the IETF ngtrans WG, can be found at: **http://www.6bone.net/misc/6to4.txt**

[7] "IPv6 Aggregatable Global Unicast Address Format," RFC 2374, can be found at: **http://www.ietf.org/rfc/rfc2374.txt**

[8] "Transmission of IPv6 Packets over IPv4 Domains without Explicit Tunnels" ("6over4"), RFC 2529, can be found at: **http://www.ietf.org/rfc/rfc2529.txt**

[9] "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, can be found at: **http://www.ietf.org/rfc/rfc2461.txt**

[10] IETF IPv6 Working Group (ipngwg) information, can be found at: **http://www.ietf.org/html.charters/ipngwg-charter.html**

BRIAN E. CARPENTER is a network researcher with the IBM Internet Division at iCAIR in Evanston Illinois. He is currently the Chair of the Internet Architecture Board (IAB) of the IETF. You can reach him at: **brian@icair.org**

KEITH MOORE is a network researcher at the Innovative Computing Laboratory of the Computer Science Department at the University of Tennessee. He is currently a Co-Director of the IETF Applications Area in the Internet Engineering Steering Group. You can reach him at: **moore@cs.utk.edu**

ROBERT FINK is a network researcher with the U.S. Dept. of Energy's Energy Sciences Network (ESnet) at the Lawrence Berkeley National Laboratory. He is currently a co-chair of the IETF ngtrans (IPng Transition) Working Group, and leads the 6bone project. You can reach him at: **fink@es.net**

IP Security

by William Stallings

In 1994, the *Internet Architecture Board* (IAB) issued a report entitled "Security in the Internet Architecture" (RFC 1636). The report stated the general consensus that the Internet needs more and better security, and it identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

These concerns are fully justified. As confirmation, the 1998 annual report from the *Computer Emergency Response Team* (CERT) lists over 1,300 reported security incidents affecting nearly 20,000 sites. The most serious types of attacks included IP spoofing, in which intruders create packets with false IP addresses and exploit applications that use authentication based on IP address; and various forms of eavesdropping and packet sniffing, in which attackers read transmitted information, including logon information and database contents.

In response to these issues, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IP (IPv4) and IPv6, meaning that vendors can begin offering these features now, and many vendors do now have some *IP Security Protocol* (IPSec) capability in their products.

Applications of IPSec

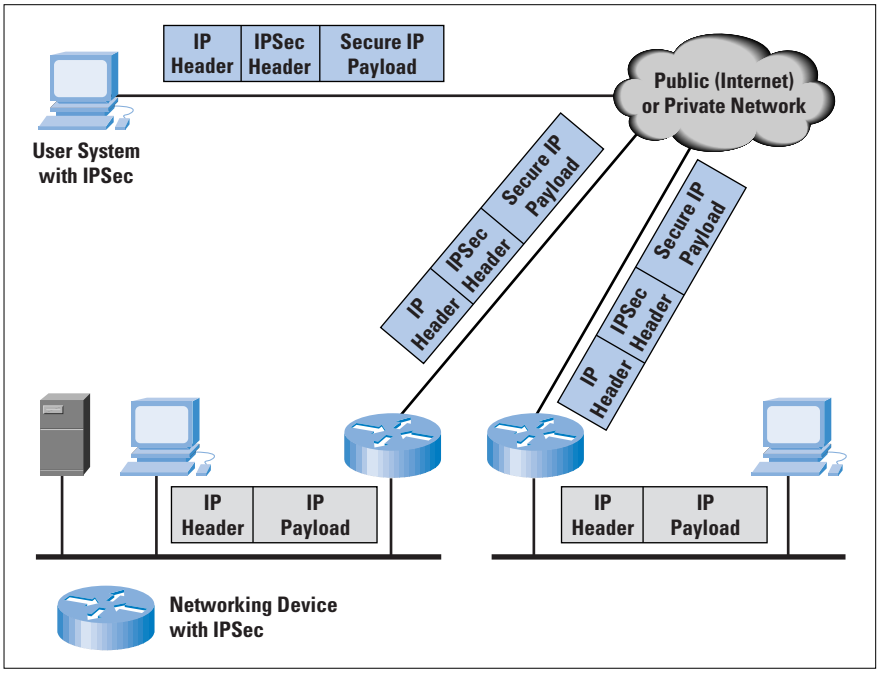
The Internet community has developed application-specific security mechanisms in numerous application areas, including electronic mail (*Privacy Enhanced Mail*, *Pretty Good Privacy* [PGP]), network management (*Simple Network Management Protocol Version 3* [SNMPv3]), Web access (Secure HTTP, *Secure Sockets Layer* [SSL]), and others. However, users have some security concerns that cut across protocol layers. For example, an enterprise can run a secure, private TCP/IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises. By implementing security at the IP level, an organization can ensure secure networking not only for applications that have security mechanisms but also for the many security-ignorant applications.

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

- Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
- Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an *Internet Service Provider* (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.
- Establishment of extranet and intranet connectivity with partners: IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- Enhancement of electronic commerce security: Most efforts to date to secure electronic commerce on the Internet have relied upon securing Web traffic with SSL since that is commonly found in Web browsers and is easy to set up and run. There are new proposals that may utilize IPSec for electronic commerce.

The principal feature of IPSec that enables it to support these varied applications is that it can encrypt or authenticate *all* traffic at the IP level. Thus, all distributed applications, including remote logon, client/server, e-mail, file transfer, Web access, and so on, can be secured. Figure 1 shows a typical scenario of IPSec usage. An organization maintains LANs at dispersed locations. Traffic on each LAN does not need any special protection, but the devices on the LAN can be protected from the untrusted network with firewalls. Since we live in a distributed and mobile world, the people who need to access the services on each of the LANs may be at sites across the Internet. These people can use IPSec protocols to protect their access. These protocols can operate in networking devices, such as a router or firewall that connects each LAN to the outside world, or they may operate directly on the workstation or server. In the diagram, the user workstation can establish an IPSec tunnel with the network devices to protect all the subsequent sessions. After this tunnel is established, the workstation can have many different sessions with the devices behind these IPSec gateways. The packets going across the Internet will be protected by IPSec but will be delivered onto each LAN as a normal IP packet.

Figure 1: An IP Security Scenario



Benefits of IPSec

The benefits of IPSec include:

- When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPSec is below the transport layer (TCP, UDP), so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router. Even if IPSec is implemented in end systems, upper layer software, including applications, is not affected.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPSec can provide security for individual users if needed. This feature is useful for offsite workers and also for setting up a secure virtual subnetwork within an organization for sensitive applications.

Is IPSec the Right Choice?

There are already numerous products that implement IPSec, but it is not necessarily the security solution of choice for a network administrator. Christian Huitema, who at the time of the development of the initial IP-Sec documents was the head of the IAB, reports that the debates over how to provide Internet-based security were among the most heated that he ever observed. One issue concerns whether security is being provided at the right protocol layer. To provide security at the IP level, it is necessary for IPSec to be a part of the network code deployed on all participating platforms, including Windows NT, UNIX, and Macintosh systems. Unless a desired feature is available on all the deployed platforms, a given application may not be able to use that feature.

On the other hand, if the application, such as a Web browser/server combination, incorporates the function, the developer can guarantee that the features are available on all platforms for which the application is available. A related point is that many Internet applications are now being released with embedded security features. For example, Netscape and Internet Explorer support SSL, which protects Web traffic. Also, many vendors are planning to support *Secure Electronic Transaction* (SET), which protects credit-card transactions over the Internet. However, for a virtual private network, a network-level facility is needed, and this is what IPSec provides.

The Scope of IPSec

IPSec provides three main facilities: an authentication-only function, referred to as *Authentication Header* (AH), a combined authentication/encryption function called *Encapsulating Security Payload* (ESP), and a key exchange function. For virtual private networks, both authentication and encryption are generally desired, because it is important both to (1) assure that unauthorized users do not penetrate the virtual private network and (2) assure that eavesdroppers on the Internet cannot read messages sent over the virtual private network. Because both features are generally desirable, most implementations are likely to use ESP rather than AH. The key exchange function allows for manual exchange of keys as well as an automated scheme.

The IPSec specification is quite complex and covers numerous documents. The most important of these, issued in November 1998, are RFCs 2401, 2402, 2406, and 2408.

Security Associations

A key concept that appears in both the authentication and confidentiality mechanisms for IP is the *Security Association* (SA). An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed, for two-way secure exchange, then two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both. A security association is uniquely identified by three parameters:

- *Security Parameters Index* (SPI): The SPI assigns a bit string to this SA that has local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- *IP destination address:* Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- *Security protocol identifier:* This indicates whether the association is an AH or ESP security association.

Hence, in any IP packet, the security association is uniquely identified by the destination address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

An IPSec implementation includes a security association database that defines the parameters associated with each SA. A security association is defined by the following parameters:

- *Sequence number counter:* A 32-bit value used to generate the sequence number field in AH or ESP headers
- *Sequence counter overflow:* A flag indicating whether overflow of the sequence number counter should generate an auditable event and prevent further transmission of packets on this SA
- *Anti-replay window:* Used to determine whether an inbound AH or ESP packet is a replay, by defining a sliding window within which the sequence number must fall
- *AH information:* Authentication algorithm, keys, key lifetimes, and related parameters being used with AH
- *ESP information:* Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP
- *Lifetime of this security association:* A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur
- *IPSec protocol mode:* Tunnel, transport, or wildcard (required for all implementations); these modes are discussed later
- *Path MTU:* Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations)

The key management mechanism that is used to distribute keys is coupled to the authentication and privacy mechanisms only by way of the security parameters index. Hence, authentication and privacy have been specified independent of any specific key management mechanism.

SA Selectors

IPSec provides the user with considerable flexibility in the way in which IPSec services are applied to IP traffic. IPSec provides a high degree of granularity in discriminating between traffic that is afforded IPSec protection and traffic that is allowed to bypass IPSec, in the former case relating IP traffic to specific SAs.

The means by which IP traffic is related to specific SAs (or no SA in the case of traffic allowed to bypass IPSec) is the nominal *Security Policy Database* (SPD). In its simplest form, an SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic. In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry.

Each SPD entry is defined by a set of IP and upper-layer protocol field values, called *selectors*. In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA. Outbound processing obeys the following general sequence for each IP packet:

- Compare the values of the appropriate fields in the packet (the selector fields) against the SPD to find a matching SPD entry, which will point to zero or more SAs.
- Determine the SA (if any) for this packet and its associated SPI.
- Do the required IPSec processing (that is, AH or ESP processing).

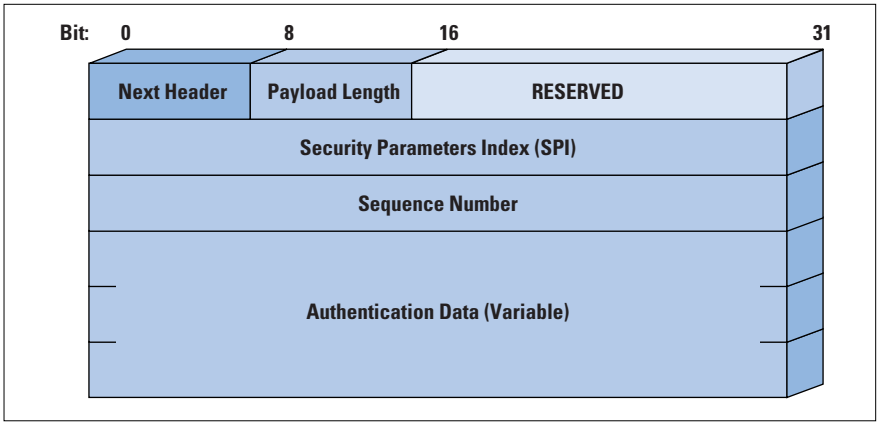
The following selectors determine an SPD entry:

- *Destination IP address:* This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (for instance, behind a firewall).
- *Source IP address:* This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (for instance, behind a firewall).
- *UserID:* UserID is used to identify a policy tied to a valid user or system name.
- *Data sensitivity level:* The data sensitivity level is used for systems providing information flow security (for instance, "Secret" or "Unclassified").
- *Transport Layer protocol:* This value is obtained from the IPv4 protocol or IPv6 *Next Header* field. This may be an individual protocol number, a list of protocol numbers, or a range of protocol numbers.
- *IPSec protocol* (AH or ESP or AH/ESP): If present, this is obtained from the IPv4 Protocol or IPv6 Next Header field.
- *Source and destination ports:* These may be individual TCP or *User Datagram Protocol* (UDP) port values, an enumerated list of ports, or a wildcard port.
- *IPv6 class:* This class is obtained from the IPv6 header. It may be a specific IPv6 Class value or a wildcard value.
- *IPv6 flow label:* This label is obtained from the IPv6 header. It may be a specific IPv6 flow label value or a wildcard value.
- *IPv4 Type of Service* (TOS): The TOS is obtained from the IPv4 header. It may be a specific IPv4 TOS value or a wildcard value.

Authentication Header

The authentication header provides support for data integrity and authentication of IP packets. The data integrity feature ensures that undetected modification to the content of a packet in transit is not possible. The authentication feature enables an end system or network device to authenticate the user or application and filter traffic accordingly; it also prevents the address spoofing attacks observed in today's Internet. The AH also guards against the replay attack described later.

Figure 2: IPSec Authentication Header



Authentication is based on the use of a *Message Authentication Code* (MAC); hence the two parties must share a secret key. The authentication header consists of the following fields (Figure 2):

- *Next Header* (8 bits): This field identifies the type of header immediately following this header.
- *Payload Length* (8 bits): This field gives the length of the authentication header in 32-bit words, minus 2. For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
- *Reserved* (16 bits): This field is reserved for future use.
- *Security Parameters Index* (32 bits): This field identifies a security association.
- *Sequence Number* (32 bits): This field contains a monotonically increasing counter value.
- *Authentication Data* (variable): This variable-length field (must be an integral number of 32-bit words) contains the *Integrity Check Value* (ICV), or MAC, for this packet.

Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The *Sequence Number* field is designed to thwart such attacks.

When a new SA is established, the *sender* initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past $2^{32} - 1$ back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of $2^{32} - 1$ is reached, the sender should terminate this SA, and negotiate a new SA with a new key.

Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered. Therefore, the IPSec authentication document dictates that the *receiver* should implement a window of size *W,* with a default of $W = 64$. The right edge of the window represents the highest sequence number, *N,* so far received for a valid packet. For any packet with a sequence number in the range from $N - W + 1$ to N that has been correctly received (that is, properly authenticated), the corresponding slot in the window is marked. Inbound processing proceeds as follows when a packet is received:

- If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
- If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
- If the received packet is to the left of the window, or if authentication fails, the packet is discarded; this is an auditable event.

Message Authentication Code

The message authentication algorithm is used to calculate a message authentication code, using an algorithm known as *HMAC.* HMAC takes as input a portion of the message and a secret key and produces a MAC as output. This MAC value is stored in the Authentication Data field of the AH header. The calculation takes place over the entire enclosed TCP segment plus the authentication header. When this IP packet is received at the destination, the same calculation is performed using the same key. If the calculated MAC equals the value of the received MAC, then the packet is assumed to be authentic. The authentication data field is calculated over:

- IP header fields that either do not change in transit (immutable) or that are predictable in value upon arrival at the endpoint for the AH SA. Fields that may change in transit and whose value on arrival are unpredictable are set to zero for purposes of calculation at both source and destination.
- The AH header other than the Authentication Data field. The Authentication Data field is set to zero for purposes of calculation at both source and destination.
- The entire upper-level protocol data, which is assumed to be immutable in transit (for instance, a TCP segment or an inner IP packet in tunnel mode).

For IPv4, examples of immutable fields are *Internet Header Length* and *Source Address.* An example of a mutable but predictable field is the Destination Address (with loose or strict source routing). Examples of mutable fields that are zeroed prior to ICV calculation are the *Time to Live* (TTL) and *Header Checksum* fields.

Note that both source and destination address fields are protected, so that address spoofing is prevented. For IPv6, examples in the base header are *Version* (immutable), *Destination Address* (mutable but predictable), and *Flow Label* (mutable and zeroed for calculation).

Encapsulating Security Payload

The encapsulating security payload provides confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality. As an optional feature, ESP can also provide the same authentication services as AH.

Figure 3: IPSec ESP Format

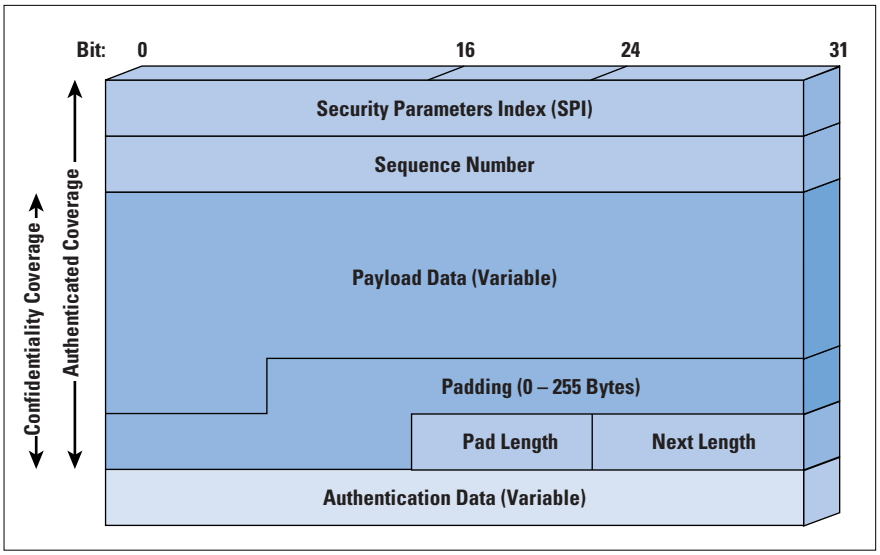


Figure 3 shows the format of an ESP packet. It contains the following fields:

- *Security Parameters Index* (32 bits): Identifies a security association
- *Sequence Number* (32 bits): A monotonically increasing counter value
- *Payload Data* (variable): A transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption
- *Padding* (0–255 bytes): Extra bytes that may be required if the encryption algorithm requires the plaintext to be a multiple of some number of octets
- *Pad Length* (8 bits): Indicates the number of pad bytes immediately preceding this field
- *Next Header* (8 bits): Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP)
- *Authentication Data* (variable): A variable-length field (must be an integral number of 32-bit words) that contains the integrity check value computed over the ESP packet minus the Authentication Data field

Encryption and Authentication Algorithms

The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an *Initialization Vector* (IV), then this data may be carried explicitly at the beginning of the Payload Data field. If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext. The current specification dictates that a compliant implementation must support the *Data Encryption Standard* (DES). A number of other algorithms have been assigned identifiers and could, therefore, be used for encryption; these include:

- Three-key triple DES
- RC5
- International Data Encryption Algorithm (IDEA)
- Three-key triple IDEA
- CAST
- Blowfish

It is now well known that DES is inadequate for secure encryption, so it is likely that many future implementations will use triple DES and eventually the *Advanced Encryption Standard* (AES). As with AH, ESP supports the use of a MAC, using HMAC.

Padding

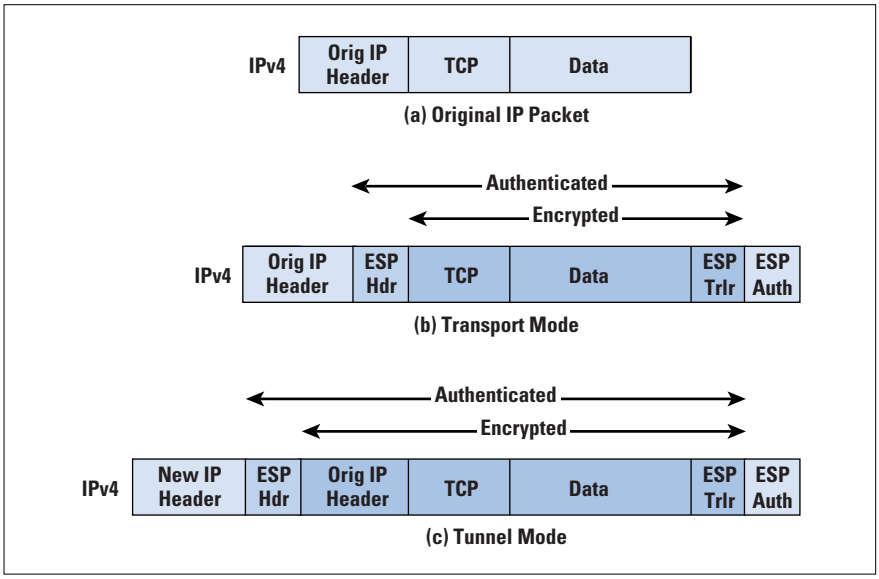
- The Padding field serves several purposes: If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (for instance, the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
- The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.
- Additional padding may be added to provide partial traffic flow confidentiality by concealing the actual length of the payload.

Figure 4 indicates the scope of ESP encryption and authentication in both transport and tunnel modes.

Transport and Tunnel Modes

Both AH and ESP support two modes of use: *transport* and *tunnel* mode.

Figure 4: Scope of ESP Encryption and Authentication



Transport Mode

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. Examples include a TCP or UDP segment, or an *Internet Control Message Protocol* (ICMP) packet, all of which operate directly above IP in a host protocol stack. For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (for instance, TCP, UDP, ICMP) and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet. This setup is shown in Figure 4b. If authentication is selected, the ESP Authentication Data field is added after the ESP trailer. The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the ciphertext plus the ESP header.

Typically, transport mode is used for end-to-end communication between two hosts (for instance, communications between a workstation and a server, or two servers). When a host runs AH or ESP over IPv4, the payload is the data that normally follows the IP header. For IPv6, the payload is the data that normally follows both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.

ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header. All IPv4 packets have a *Next Header* field. This field contains a number for the payload protocol, such as 6 for TCP and 17 for UDP. For transport mode, the IP Next Header field is decimal 51 for AH, or 50 for ESP. This tells the receiving machine to interpret the remainder of the packet after the IP header as either AH or ESP. Both the AH and ESP headers also have a Next Header field.

As an example, let's examine a Telnet session within an ESP packet in transport mode. The IP header would contain 51 in the Next Header field. In the ESP header, the Next Header field would be 6 for TCP. Within the TCP header, Telnet would be identified as port 23.

Transport mode operation may be summarized for ESP as follows:

- At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced with its ciphertext to form the IP packet for transmission. Authentication is added if this option is selected.
- The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but will not need to examine the ciphertext.
- The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment. This process is similar for AH, however the payload (upper layer protocol) is not encrypted.

Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application. This mode of operation is also reasonably efficient, adding little to the total length of the IP packet. One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets.

Tunnel Mode

Tunnel mode encapsulates an entire IP packet within an IP packet to ensure that no part of the original packet is changed as it is moved through a network. The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way need to examine the inner IP header. For ESP, this is shown in Figure 4c. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header. Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus ciphertext plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis. Tunnel mode is used when one or both ends of an SA is a security gateway, such as a firewall or router that implements IPSec. With tunnel mode, a number of hosts on networks behind firewalls may engage in secure communications without implementing IPSec. The unprotected packets generated by such hosts are tunneled through external networks by tunnel mode SAs set up by the IPSec process in the firewall or secure router at the boundary of the local network.

Whereas the transport mode is suitable for protecting connections between hosts that support the ESP feature, the tunnel mode is useful in a configuration that includes a firewall or other sort of security gateway that protects a trusted network from external networks. In this latter case, encryption occurs only between an external host and the security gateway or between two security gateways. This setup relieves hosts on the internal network of the processing burden of encryption and simplifies the key distribution task by reducing the number of needed keys. Further, it thwarts traffic analysis based on ultimate destination.

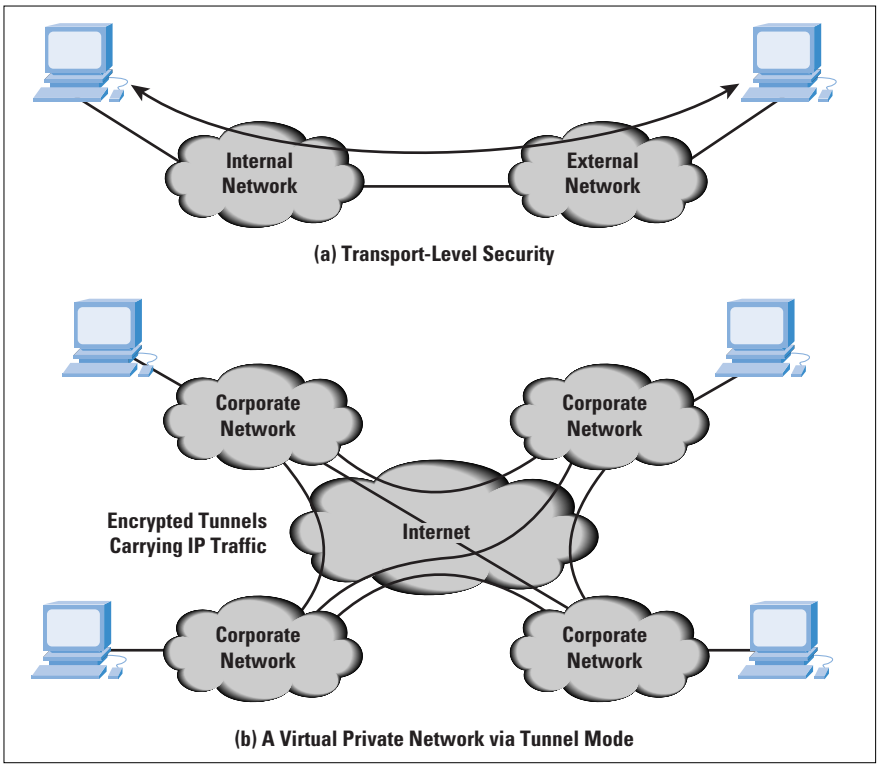
Let's use the diagram in Figure 1 as an example of how tunnel mode IPSec operates. The following steps occur for transfer of a transport-layer segment from the user system to one of the servers on one of the protected LANs.

- The user system prepares an inner IP packet with a destination address of the target host on the internal LAN. For a Telnet session, this packet would be a TCP packet with the original SYN flag set with a destination port set to 23. This entire IP packet is prefixed by an ESP header; then the packet and ESP trailer are encrypted and Authentication Data may be added. The Next Header field of the ESP header would be decimal 4 for IP-in-IP, indicating that the entire original IP packet is contained as the "payload." The resulting block is encapsulated with a new IP header (base header plus optional extensions such as routing and hop-by-hop options for IPv6) whose destination address is the firewall; this forms the outer IP packet. The Next Header field for this IP packet is 50 for ESP.
- The outer packet is routed to the destination firewall. Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers but does not need to examine the ciphertext.
- The destination firewall examines and processes the outer IP header plus any outer IP extension headers. Then, on the basis of the SPI in the ESP header, the gateway decrypts the remainder of the packet to recover the plaintext inner IP packet. This packet is then transmitted in the internal network.
- The inner packet is routed through zero or more routers in the internal network to the destination host. The receiver would have no indication that the packet had been encapsulated and protected by the "tunnel" between the user system and the gateway. It would see the packet as a request to start a Telnet session and would respond back with a TCP SYN/ACK, which would go back to the gateway. The gateway would encapsulate that packet into an IPSec packet and transport it back to the user system through this "tunnel." That return packet would be processed to find the original packet, which would contain the SYN/ACK for the Telnet session.

Common Uses of IPSec in Real Networks

Figure 5 shows two ways in which the IPSec ESP service can be used. In the upper part of the figure, encryption (and optionally authentication) is provided directly between two hosts. Figure 5b shows how tunnel mode operation can be used to set up a *Virtual Private Network* (VPN). In this example, an organization has four private networks interconnected across the Internet. Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts. By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability. The former technique is supported by a transport mode SA, while the latter technique uses a tunnel mode SA.

Figure 5: Transport-Mode versus Tunnel-Mode Encryption



Key Management

The key management portion of IPSec involves the determination and distribution of secret keys. The IPSec Architecture document mandates support for two types of key management:

- *Manual:* A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.
- *Automated:* An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration. An automated system is the most flexible but requires more effort to configure and requires more software, so smaller installations are likely to opt for manual key management.

The default automated key management protocol for IPSec is referred to as *Internet Key Exchange* (IKE). IKE provides a standardized method for dynamically authenticating IPSec peers, negotiating security services, and generating shared keys. IKE has evolved from many different protocols and can be thought of as having two distinct capabilities. One of these capabilities is based on the *Internet Security Association and Key Management Protocol* (ISAKMP). ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes. ISAKMP by itself does not dictate a specific key exchange algorithm; rather, ISAKMP consists of a set of message types that enable the use of a variety of key exchange algorithms. The actual key exchange mechanism in IKE is derived from Oakley and several other key exchange protocols that had been proposed for IPSec. Key exchange is based on the use of the Diffie-Hellman algorithm, but provides added security. In particular, Diffie-Hellman alone does not authenticate the two users that are exchanging keys, making the protocol vulnerable to impersonation. IKE includes mechanisms to authenticate the users.

Public Key Certificates

An important element of IPSec key management is the use of public key certificates. In essence, a public key certificate is provided by a trusted *Certificate Authority* (CA) to authenticate a user's public key. The essential elements include:

- Client software creates a pair of keys, one public and one private. The client prepares an unsigned certificate that includes a user ID and the user's public key. The client then sends the unsigned certificate to a CA in a secure manner.
- A CA creates a signature by calculating the hash code of the unsigned certificate and encrypting the hash code with the CA's private key; the encrypted hash code is the signature. The CA attaches the signature to the unsigned certificate and returns the now signed certificate to the client.
- The client may send its signed certificate to any other user. That user may verify that the certificate is valid by calculating the hash code of the certificate (not including the signature), decrypting the signature using the CA's public key, and comparing the hash code to the decrypted signature.

If all users subscribe to the same CA, then there is a common trust of that CA. All user certificates can be placed in the directory for access by all users. In addition, a user can transmit his or her certificate directly to other users. In either case, once B is in possession of A's certificate, B has confidence that messages it encrypts with A's public key will be secure from eavesdropping and that messages signed with A's private key are unforgeable.

If there is a large community of users, it may not be practical for all users to subscribe to the same CA. Because it is the CA that signs certificates, each participating user must have a copy of the CA's own public key to verify signatures. This public key must be provided to each user in an absolutely secure (with respect to integrity and authenticity) way so that the user has confidence in the associated certificates. Thus, with many users, it may be more practical for there to be many CAs, each of which securely provides its public key to some fraction of the users. In practice, there is not a single CA but rather a hierarchy of CAs. This complicates the problems of key distribution and of trust, but the basic principles are the same.

Whither IP Security

The driving force for the acceptance and deployment of secure IP is the need for business and government users to connect their private WAN/LAN infrastructure to the Internet for (1) access to Internet services and (2) use of the Internet as a component of the WAN transport system. Users need to isolate their networks and at the same time send and receive traffic over the Internet. The authentication and privacy mechanisms of secure IP provide the basis for a security strategy.

Because IP security mechanisms have been defined independent of their use with either the current IP or IPv6, deployment of these mechanisms does not depend on deployment of IPv6. Indeed, it is likely that we will see widespread use of secure IP features long before IPv6 becomes popular.

Recommended Web Sites

- The IPSec Working Group of the IETF. Charter for the group and latest RFCs and Internet Drafts for IPSec:
  `http://ietf.org/html.charters/ipsec-charter.html`
- IPSec Resources: List of companies implementing IPSec, implementation survey, and other useful material:
  `http://web.mit.edu/tytso/www/ipsec/index.html`

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He has a Ph.D. in computer science from M.I.T. His latest book is *Local and Metropolitan Area Networks, Sixth Edition* (Prentice Hall, 2000). His home in cyberspace is **`WilliamStallings.com`** and he can be reached at **`ws@shore.net`**

Quality of Service—Fact or Fiction?

by Geoff Huston, Telstra

Much has been written about the potential of *Quality of Service* (QoS) and the Internet. However, much of the material is strong on promise, but falls short in critical analysis. In an effort to balance the picture, we present here a brief status report on the QoS effort, exposing some of the weaknesses in the current QoS architectures.

The QoS Service

The default service offering associated with the Internet is a *best-effort* service, where the network treats all traffic in exactly the same way. There is no consistent service outcome from the Internet best-effort service model. When the load level is low, the network delivers a high-quality service. The best-effort Internet does not deny entry to traffic, so as the load levels increase, the network congestion levels increase, and service-quality levels decline uniformly. This decline in service is experienced by all traffic passing through a congestion point, and is not limited to the most recently admitted traffic flows. For many applications, this best-effort response is perfectly acceptable. When network capacity is available, the application can make use of the resource, whereas when the level of contention for network bandwidth is high, each application will experience similar levels of congestion. A best-effort network service is a good match to opportunistic applications that can vary their data transfer rate in response to signaled network load.

The objective of various Internet QoS efforts is to augment this service with a number of selectable service responses. These service responses may be different from the best-effort service by some form of superior service response, such as lower delay, lower jitter, or greater bandwidth. These responses are relative, where the service outcome is claimed to be no worse than best effort at any time, and superior to best-effort under congestion load. Alternatively, QoS service responses may be distinguished by providing a consistent, and therefore predictable, service response that is unaffected by network congestion levels. These are quantitative service responses, where the characteristics of the service can be measured against a constant outcome. A quantitative service many be one that constrains jitter to a maximum level, or one that makes a certain bandwidth available, within parameters of bounded jitter, similar to a conventional leased line. Such constant-rate services may be superior to best-effort services when the network is under load, but they may also offer inferior service when the network is under negligible load. The essential attribute of these services is one of consistency.

Why is there a need for relative or consistent service profiles within the Internet? The underlying reasons for introducing QoS into the Internet appear to be threefold: First is the desire to provide high-quality support for IP voice and video services, second is the desire to manage the ser-

vice response provided to low-speed access devices, such as Internet mobile wireless devices, and third is the desire to provide a differentiated Internet access service, providing a network client with a range of service-quality levels at a range of prices.

Obviously this is a broad agenda, where there are requirements to extend specific network services to applications, requirements to adapt network services to particular transmission characteristics, and requirements to manage network resources to achieve particular response characteristics for an aggregated collection of traffic.

Approaches to QoS

The relevant efforts within the *Internet Engineering Task Force* (IETF) have been addressing standards for QoS mechanisms within the network.

The initial approach to QoS was that of the *Integrated Services* architecture. This approach focuses on the application as the trigger for QoS. Here, the application first signals its service requirements to the network in the form of a reservation, and the network responds to this request. The application proceeds only if the network has indicated that it is able to carry the additional load at the requested service level by committing to the reservation. The reservation remains in force until the application explicitly requests termination of the reservation, or the network signals to the application that it is unable to continue the reservation. The essential feature of this model is the "all-or-nothing" nature of the service model. Either the network commits to the reservation, in which case the application does not have to monitor the level of network response to the service, or the network indicates that it cannot meet the reservation. This approach imposes per-application state within the network, and for large-scale networks, such as the global Internet itself, this approach alone does not appear to be viable (see Figure 1).

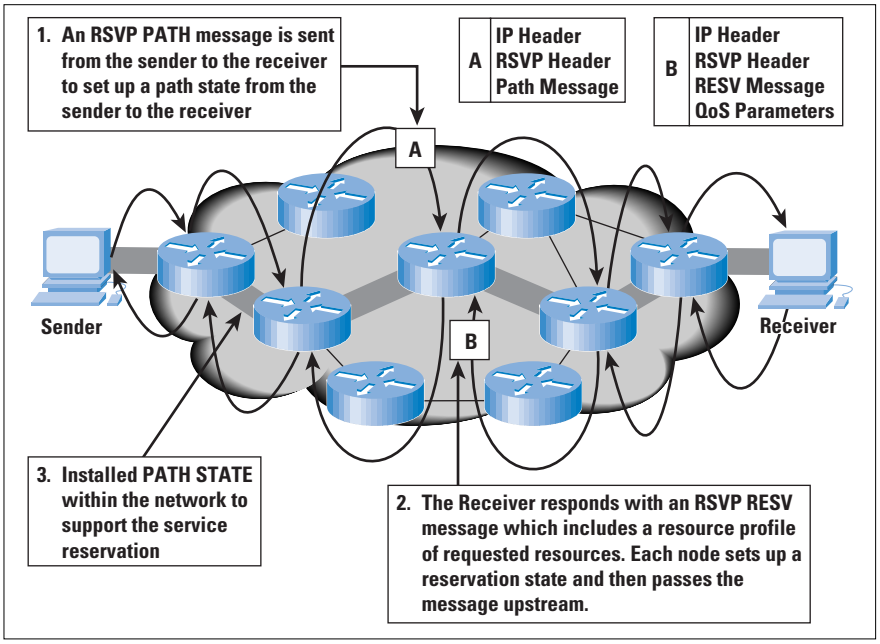


Figure 1: The Integrated Services QoS Architecture

The subsequent approach to QoS mechanisms has been to look at the core of the network, and examine those mechanisms that can provide differentiated service outcomes with appropriate scaling properties. This approach, the *Differentiated Services* architecture, includes dropping the concept of a per-application path state across the network using instead the concept of aggregated service mechanisms. Within the aggregated service model, the network provides a smaller number of different service classes and aggregates similar service demands from a set of applications into a single service class. Aggregated services are typically seen as an entry filter, where on entry to the network each packet is classified into a particular service profile. This classification is carried within the IP packet header, using 6 bits from the deprecated IP *Type of Service* (TOS) header to carry the service coding. The network then uses this service code in the packet header to treat this packet identically to all other packets within the same service code. While this approach does possess the ability to scale across the entire Internet, there are numerous unresolved issues relating to the quality signaling between individual applications and the network. The aggregated service model does not allow an individual application to sense if it is receiving the necessary service response from the network (see Figure 2).

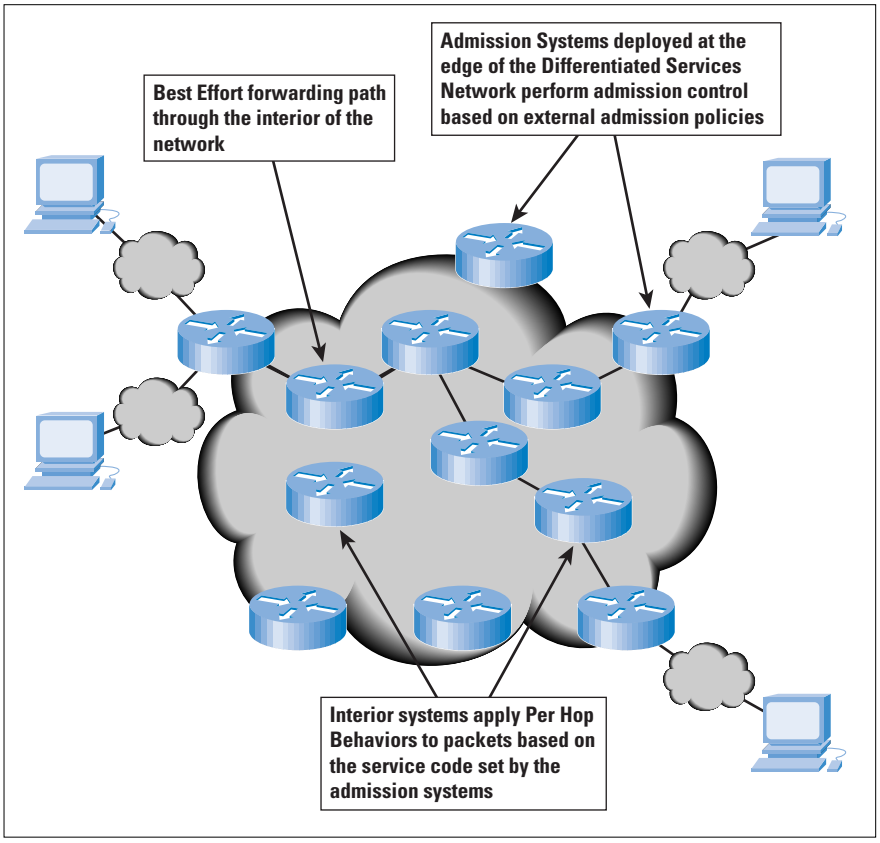


Figure 2: The Differentiated Services QoS Architecture

QoS Deployment

Neither approach alone is adequate to meet the QoS requirements. The Integrated Services approach alone imposes an excessive load in the core of large networks through the imposition of a per-application path state. The Differentiated Services approach does provide superior scaling properties through the use of aggregated service elements, but includes no concept of control signaling to inform the traffic conditioning elements of the current state of the network, or the current per-application requirements.

The underlying question then becomes: Is a combination of these two approaches sufficient to allow QoS to be widely deployed on the Internet?

At this stage the response does appear to be a "No." Perhaps this strong negative response should be further qualified. The existing tools are insufficient to support widespread use of QoS-based services on the multiprovider public Internet. The qualification is that within the enterprise network environment there are much stronger drivers for QoS mechanisms and much greater levels of administrative control over the overall network architecture, while within the multiprovider public Internet, these drivers are not apparent. The enterprise approach may also have some parallels within a single IP carrier's network, or even across some forms of bilateral agreements between carriers. However, such approaches are not anticipated to be a widespread feature of the public Internet service environment.

Let's look more closely at the public Internet and QoS to see why there is a mismatch between the two. The major stumbling blocks in attempting to address how QoS could be deployed in the public Internet are both engineering and economic in nature.

From an engineering perspective, we need to remember that in order to actually deliver any reasonable assurance of a quality-differentiated service, the service-quality mechanism chosen must be deployed across all networks along the end-to-end paths of the quality-service traffic. In a heterogeneous multiprovider environment such as the public Internet, this outcome is very unlikely. Within the tens of thousands of component service providers that make up the global Internet, such uniformity of action is highly improbable. The IPv6 transition structure correctly identifies the first step as isolated "islands" of IPv6 functionality, interconnected by some form of IPv6 "bridges." While the potential scenario of initial QoS deployment may be similar, in terms of isolated islands of deployment of QoS services, there is a much stricter requirement for the "bridges" across the non-QoS-aware parts of the network; namely, that they do not distort the service outcomes. In effect, this scenario requires a QoS response from a non-QoS system (see Figure 3).

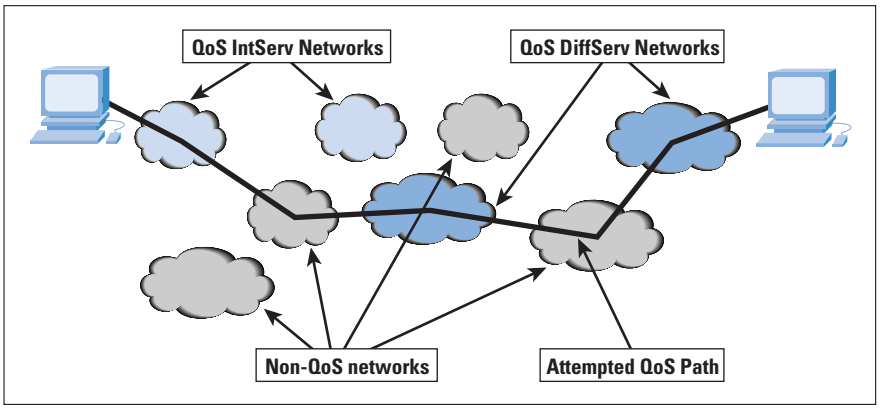


Figure 3: Attempted End-to-End QoS across the Public Internet

The engineering issues are deeper than simply the considerations of transition within a potential deployment scenario. The issues include:

- The need for QoS-enabled applications that can predict their service requirements in advance, and be able to signal these requirements into the network.
- In the case of the differentiated service approach of admission controls, there is a requirement for the interior of the network to be able to signal current load conditions to the network admission systems. This system also requires that the admission control points be able to use admission-decision support systems in order to include consideration of the service load, the current network load, and the policy parameters of the network that may allow some level of preemption of various admission decisions in order to meet high-priority service requirements.
- The signaling and negotiation aspect of QoS extends into the interdomain space, where two or more service providers need to negotiate mutually acceptable service profiles, and associated service access. This extends beyond the addition of bilateral agreements and encompasses the requirement to add QoS attributes to interdomain routing protocols. The tools and operating techniques required to support this functionality remain poorly defined.
- Measurement of service performance remains an area in which existing measurement tools are lacking. While it is possible to instrument every active device within a network into a network management system, such an element-by-element view does not readily translate to the end-to-end view of application service performance.

From an economic perspective, we must remember that no current Internet retail tariff includes a concept of end-to-end tariffed transactions. All tariffs are access based, because application transactions are not readily visible to the Internet network. In addition, no technically stable or financially stable structure of interprovider interconnection financial settlements exists today. The financial model of the Internet from an economic viewpoint is very polarized, with only customer and zero-dollar peer arrangements dominating the interprovider space. However, end-to-end QoS transactions demand a different economic model.

The initiator of the end-to-end QoS transaction has the discretion of choosing whether to request an end-to-end service profile. If such a profile is requested, the initiator should pay the initiating provider a retail tariff to cover the entire end-to-end cost of the transaction, and the initiating provider must then indicate a willingness to financially settle with transit peer networks in order for these transit peers to devote additional resources to service the traffic associated with this transaction, and so forth through the entire path of transit providers. The arbitrary nature of the Internet transits, the dynamic nature of routing, and the lack of transaction setups in any scalable form of QoS mechanisms make this entire scenario highly improbable within our current understanding of interprovider policy-management mechanisms.

The relatively loosely coordinated structure of the public Internet will have to change from the state we have today if we want to use QoS-based services. The changes include:

- A common selection of a set of QoS mechanisms to deploy,
- Ubiquitous deployment of these mechanisms across both service provider and client networks,
- The adoption of a uniform set of retail tariffs for QoS services,
- The definition and common acceptance of multi-party QoS-related financial settlements that support fair and equitable cost distribution among multiple providers, and
- The definition of commonly accepted service performance metrics and related measurement methodologies to allow end-to-end and network-by-network service outcomes to be objectively assessed.

This is a significant agenda for the industry at large to undertake, and more so in an environment that features diversity and vigorous competition between various public Internet service providers.

An additional factor is also working against QoS deployment in the public Internet space. The increasing availability of very-high-speed transmission systems is bringing network carriage capacity down to the level of an abundant commodity across large parts of the Internet world. As the unit costs of network capacity decline in the face of increasing levels of availability of transmission systems, the market niche that QoS could occupy in managing a scarce resource is shrinking. The driver for QoS deployment is not that the best-effort service is not good enough. The problem that QoS is attempting to address is one of allocation of network capacity at those points in time when the network is under heavy load, or, in other words, taking on the task of allocating capacity when there is not enough network capacity to meet every demand. When a network is under load, the QoS response is to place additional control functionality in both applications and in the network to manage this allocation function. Obviously such an activity imposes additional costs on the network operators and the network client. Such additional costs have not created any additional network capacity.

The total sum of demand remains in excess of capacity after the deployment of QoS mechanisms. The alternative approach is to incur additional cost by augmenting the capacity of the network. This approach minimizes the impact of load on the network causing disruption to individual transactions. Again this approach imposes additional costs onto the network, but in an environment of abundant transmission capacity, it may often be the more cost-effective approach.

Where does this leave QoS and the public Internet? There is no doubt that QoS is a very stimulating area of research, with much to offer the enterprise network environment, but in asking for QoS to be deployed within the existing incarnation of the public multiprovider Internet, we may be simply asking for too much at this point in time. More effort is required to turn a QoS Internet into a reliable production platform.

Further Reading

[1] Huston, G., *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, John Wiley & Sons, January 2000.

A detailed examination of Internet Quality of Service technologies and their potential application within the Internet.

[2] Kilkki, K., *Differentiated Services for the Internet,* ISBN 1578701325, Macmillan Technical Publishing, June 1999.

An in-depth look at the Differentiated Services architecture and its use in enabling networks to handle traffic classes in a specific manner.

[3] Durham, D., and Yavatar, R., *Inside the Internet's Resource Reservation Protocol: Foundations for Quality of Service,* ISBN 0471322148, John Wiley & Sons, April 1999.

At the core of the Integrated Services architecture is a signaling protocol to undertake service reservations. The Resource ReSerVation Protocol (RSVP) is a signaling protocol that can undertake this role. This book describes both the Integrated Services architecture and RSVP in detail.

[4] Odlyzko, A., "The Economics of the Internet: Utility, Utilization, Pricing, and Quality of Service," 1998. Available at:
`www.research.att.com/~amo`

A paper arguing the point of view that overprovisioning data networks is a viable and economically sustainable response to the demands for service quality within data networks, and that such a response is technically and economically superior to implementing QoS responses within the network.

[5] Braden, R., Clark, D., and Shenker, S., "Integrated Services in the Internet Architecture: An Overview," RFC 1633, June 1994.

This RFC describes the components of the Integrated Services architecture, a proposed extension to the Internet architecture, and protocols to support real-time traffic flows through service-quality commitments.

[6] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., "An Architecture for Differentiated Services," RFC 2475, Proposed Standard, December 1998.

The architecture description for the Differentiated Services enhancements to the Internet Protocol. This architecture achieves scalability by aggregating traffic classification state, which is conveyed by means of IP-layer packet marking using the Differentiated Services (DS) field. Packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path. Sophisticated classification, marking, policing, and shaping operations need to be implemented only at network boundaries or hosts. Network resources are allocated to traffic streams by service-provisioning policies that govern how traffic is marked and conditioned upon entry to a differentiated services-capable network, and how that traffic is forwarded within that network.

[7] Gray, T., "Enterprise QoS Survival Guide: 1999 Edition," 1999. Available at:
http://staff.washington.edu/gray/papers/eqos22.html

A detailed view of an approach to supporting QoS in an enterprise environment. The paper is an excellent example of the procedural steps involved in network engineering, detailing the intended environment, the available tools and the desired outcomes, and then examining the viability of a number of QoS solutions.

[8] Huston, G., "Next Steps for the IP QoS Architecture." Available at:
www.ietf.org/internet-drafts/draft-iab-qos-00.txt

While there has been significant progress in the definition of IP QoS architecture, there are a number of aspects of QoS that appear to need further elaboration as they relate to translating a set of tools into a coherent platform for end-to-end service delivery. This document highlights the outstanding issues relating to the deployment and use of QoS mechanisms within the Internet, noting those areas where further standards work may be required. This draft is a work item of the Internet Architecture Board Working Group of the IETF.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and is the chair of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089 and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Book Review

Removing the Spam

Removing the Spam: Email Processing and Filtering, Geoff Mulligan, ISBN 0-201-37957-0, Addison-Wesley, 1999.
http://cseng.aw.com/bookdetail.qry?ISBN=0-201-37957-0&ptype=0

Do not be fooled by the title of this book. You might purchase this book, part of the Addison-Wesley Networking Basics Series, thinking you are just getting information dealing with unsolicited commercial e-mail (commonly called, to Hormel's displeasure, "spam"). The title is probably the work of a marketeer who thought "spam" in the title would *sell!* The subtitle really describes the meat of the matter. This short, but thorough, book is about e-mail processing and filtering—dealing with spam, yes, but so much more.

A collection of essential information for the Internet e-mail "gatekeeper," *Removing the Spam* is really geared for the gatekeeper using a UNIX-based system, so NT system administrators be forewarned. Being an e-mail gatekeeper on the Internet involves keeping the e-mail flowing, making sure the automated processes in place do the job, supporting e-mail "mailing lists," and providing the services and features your users want or need for e-mail processing.

Commercial products support some of the many requirements, but the best software for most of these functions is freely available on the Internet. Geoff provides answers to the requirements using the most popular and commonly used solutions: *Sendmail* for mail delivery, *procmail* for e-mail filtering, and *majordomo* and *smartlist* for mailing-list management.

The book, however, tries to do a bit too much. Geoff indicates that the intended audience is not only the system administrator, but also the e-mail end users wanting to filter their own personal e-mail as well as those who want to run their own mailing list. Because of this broad audience, there are times when the book delves too long in the basics, giving the impression of topics added to lengthen the book. The overview of IP protocols, the brief history of the Internet, suggestions for users dealing with spammers, and mailing-list etiquette are examples that come to mind. Nevertheless, the other topics covered are "net essentials," and worth skimming over the already known.

The book clearly defines spam and its evils, and presents the tools and techniques available for removing, or at least minimizing, the spam. It is probably too ambitious when covering e-mail forgery and tracing e-mail spam, but leaves no essential unmentioned.

Sendmail coverage is good, dealing with installation as well as configuration, highlighting antispam features, and how to use them. Though not covering as much detail as other books that focus on Sendmail, the important elements of building and modifying are handled, as well as Sendmail's use of data bases, including the infamous "Realtime Blackhole List" (**`http://maps.vix.com/rbl/`**).

The e-mail gatekeeper, as well as end users of e-mail, can use procmail to preprocess e-mail before final delivery. Procmail is powerful and flexible, and, so, can be difficult to configure properly. Configuration files examples with explanations allow even the procmail-savvy reader to learn and try something new.

The mailing list section again instructs both system administrator and user. Information about subscribing, unsubscribing, and getting information from the mailing list software is useful for the user. The administrator will appreciate the examples of getting, installing, configuring, and running majordomo and smartlist. Geoff gives suggestions about when a manual versus automated solution is best.

About the Author

I knew Geoff back in our Digital Equipment Corporation days when he worked in the Network Systems Lab. My group ran one of the corporate Internet gateways, modeled after the one at NSL. Further, the group I ran also productized and delivered what is arguably the first commercial Internet firewall, based on a design from the team at NSL. All this to say, Geoff certainly has the background to write about these topics. Since those days, Geoff has been busy with other Internet endeavors, such as starting USA.NET and creating the NetAddress product (permanent, follow-you-anywhere e-mail addresses) and helping develop the Sun Microsystems Sunscreen Firewall. He also founded Geocast Network Systems. In various roles, in differing capacities, Geoff has had to wrestle with the matters covered in his book. What he writes is based on experience learned in the danger zone of the Internet gateway.

Organization

The book is divided into four chapters. The first chapter, the introduction really, is strangely entitled "The Dawn of Electronic Mail." This is also the "roughest" chapter. It is difficult to understand why some topics are covered in the order that they are here (and why some are covered at all—the aforementioned "list etiquette" and "Size and Growth of the Internet," for example). It introduces (needlessly, I think) The Internet Protocols, but then reviews the basics of understanding e-mail systems. It introduces spam, along with antispam resources, and the topics in the rest of the book to be covered in detail: e-mail processing, filtering, and e-mail lists.

Chapter 2 is entitled "Sendmail" and covers obtaining, installing, configuring, and running Sendmail on a UNIX machine. It gives the commands to build and install Sendmail and your Sendmail configuration file. This coverage is not detailed enough for *every* situation, but gives the most common configuration information, which should satisfy most readers' needs. Included are instructions for using Sendmail to help stop (or avert) spam at the mail gateway.

Chapter 3 unravels the mysteries behind procmail configuration for e-mail filtering. This chapter covers getting the software, installing it, and using procmail—the latter for system administrators and users alike. There are example "ready-to-run filters" included. Caveat: Some of the scripts have inherent errors. No doubt these errors are unfortunate publication glitches, but they do detract from the usefulness of this chapter. Geoff has compiled an errata list with corrected scripts. This can be found at: **http://www.hz.com/spam/eratta**

Chapter 4 covers mailing lists, specifically discussing administering them "by hand" (just using Sendmail) or "automatically" (majordomo and smartlist). Again, examples are given with step-by-step commands.

Closing Thoughts

Production errors aside (the serious ones in the procmail chapter and others that are just nits to pick—the "P" in ARPA stands for "Projects," not "Project"), this book is useful as an introduction as well as a reminder of things forgotten. I can recommend this book to the novice or seasoned e-mail gatekeeper, and I will recommend it to the students in my Sendmail courses.

—*Frederick M. Avolio, Avolio Consulting*
fred@avolio.com

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

Fragments

ICANN Launches Membership Web Site for Individual Internet Users

The *Internet Corporation for Assigned Names and Numbers* (ICANN) recently announced the launch of its At Large Membership Web site. After considerable public input, the ICANN Board has developed this program as a new way for Internet users from all over the globe to participate directly in the ICANN process. Individuals can register to become ICANN members at **http://members.icann.org**

The At Large Membership of ICANN will give individual members of Internet communities worldwide a voice in the selection of Directors to the ICANN Board. By becoming an ICANN member, individuals will have an opportunity to become part of the ICANN "bottom-up" approach to making policy concerning Internet names and addresses. The basic requirements for applying to become an ICANN At Large member are: The completion of an online membership application, a working Internet e-mail address, and a single physical residence verified by a postal mail address. Thanks to a grant from the Markle Foundation, the initial launch of ICANN's At Large Membership program has been funded without the need for membership dues.

The ICANN Board will consider and adopt further policy about composition and structure of the At Large Membership, and establish rules for the nomination and election of candidates for the At Large Council. It is hoped that the target goal of 5,000 members can be reached in the next few weeks in order to move forward with the At Large Elections later this year.

ICANN is a non-profit, international corporation formed to oversee a select set of Internet technical management functions currently managed by the U.S. Government, or by its contractors and volunteers. Specifically, ICANN is assuming responsibility for coordinating the management of the *Domain Name System* (DNS), the allocation of IP address space, the assignment of protocol parameters, and the management of the root server system.

Online Registration for INET 2000 Now Open

INET 2000, the annual conference of the Internet Society (ISOC) will be held in Yokohama, Japan, July 18–21. You can register for this event by visiting ISOC's Web site at:
http://www.isoc.org/inet2000/register.shtml

Denial of Service Attacks

In early February, several high-profile Internet Web sites were severely disrupted by a number of so-called *Distributed Denial of Service* (DDoS) attacks. We plan to publish an article on this topic in the future. Meanwhile, we recommend you visit the Denial of Service Resource Page at **http://www.denialinfo.com/**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.