The
Internet Protocol Journal

March 1999 Volume 2, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download
previous issues of IPJ in
PDF format from:
www.cisco.com/ipj

From The Editor

Today's Internet is comprised of numerous interconnected *Internet Service Providers* (ISPs), each serving many constituent networks and end users. Just as individual regional and national telephone companies interconnect and exchange traffic and form a global telephone network, the ISPs must arrange for points of interconnection to provide global Internet service. This interconnection mechanism is generally called "peering," and it is the subject of a two-part article by Geoff Huston. In Part I, which is included in this issue, he discusses the technical aspects of peering. In Part II, which will follow in our next issue, Mr. Huston continues the examination with a look at the business arrangements (called "settlements") that exist between ISPs, and discusses the future of this rapidly evolving marketplace.

In the early 1990s, concern grew regarding the possible depletion of the IP version 4 address space because of the rapid growth of the Internet. Predictions for when we would literally run out of IP addresses were published. Several proposals for a new version of IP were put forward in the IETF, eventually resulting in IP version 6 or IPv6. At the same time, new technologies were developed that effectively slowed address depletion, most notably *Classless Inter-Domain Routing* (CIDR) and *Network Address Translators* (NATs). Today there is still debate as to if and when IPv6 will be deployed in the global Internet, but experimentation and development continues on this protocol. We asked Robert Fink to give us a status report on IPv6.

We've already discussed the historical lack of security in Internet technologies and how security enhancements are being developed for every layer of the protocol stack. This time, Marshall Rose and David Strom examine the state of electronic mail security. We clearly have a way to go before we see "seamless integration" of security systems with today's e-mail clients.

Our first Letter to the Editor is included on page 46. As always, we would love to hear your comments and questions regarding anything you read in this journal. Please contact us at `ipj@cisco.com`

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

Interconnection, Peering and Settlements—Part I

by Geoff Huston, Telstra

Technology and business models share a common evolution within the Internet. To enable deployment of the technology within a service environment, a robust and stable business model also needs to be created. This tied destiny of technology and business factors is perhaps most apparent within the area of the interconnection of *Internet Service Providers* (ISPs). Here there is an interaction at a level of technology, in terms of routing signaling and traffic flows, and also an interaction of business models, in terms of a negotiation of benefit and cost in undertaking the interconnection. This article examines this environment in some detail, looking closely at the interaction between the capabilities of the technical protocols, their translation into engineering deployment, and the consequent business imperatives that such environments create.

It is necessary to commence this examination of the public Internet with the observation that the Internet is not, and never has been, a single network. The Internet is a collection of interconnected component networks that share a common addressing structure, a common view of routing and traffic flow, and a common view of a naming system. This interconnection environment spans a highly diverse set of more than 50,000 component networks, and this number continues, inexorably, to grow and grow. One of the significant aspects of this environment is the competitive Internet service industry, where many thousands of enterprises, both small and large, compete for market share at a regional, national, and international level.

Underneath the veneer of a highly competitive Internet service market is a somewhat different environment, in which every ISP network must interoperate with neighboring Internet networks in order to produce a delivered service outcome of comprehensive connectivity and end-to-end service. No ISP can operate in complete isolation from others while still offering public Internet services, and therefore, every ISP not only must coexist with other ISPs but also must operate in cooperation with other ISPs.

This article examines both the technical and business aspects that surround this ISP interaction, commonly referred to as "interconnection, peering, and settlements." It examines the business motivation for interconnection structures, and then the technical architectures of such environments. The second part looks at the business relationships that arise between ISPs in the public Internet space, and then examines numerous broader issues that will shape the near-term future of this environment.

[This article is based in part on material in *The ISP Survival Guide,* by Geoff Huston, ISBN 201-3-45567-9, published by Wiley. Used with permission.]

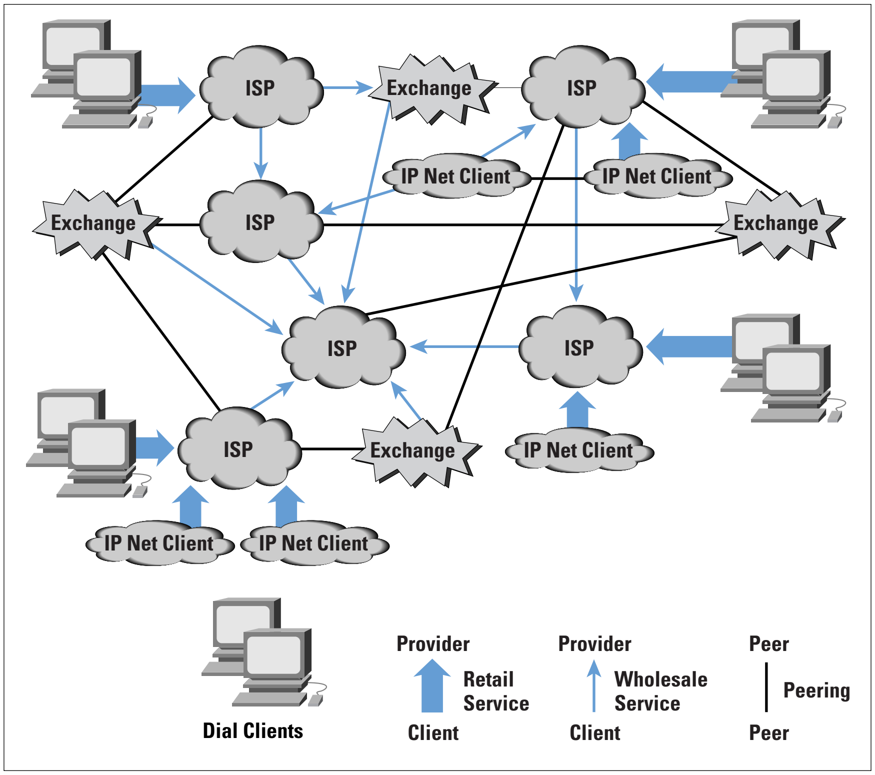
Interconnection: Retailing, Reselling, and Wholesaling

To provide some motivation for this issue of ISP interconnection, it is first appropriate to look at the nature of the environment. The regulatory framework that defined the traditional structure of other communications enterprises such as telephony or postal services was largely absent in the evolution of the Internet service industry. The resultant service industry for the Internet is most accurately characterized as an outcome of business and technology interaction, rather than a planned outcome of some regulatory process. This section examines this interaction between business and technology within the ISP environment.

A natural outcome of the Internet model is that the effective control of the retail service environment rests with a network client of an access service rather than with the access service provider. As such, a client of an ISP access service has the discretionary ability to resell the access service to third-party clients. In this environment, reselling and wholesaling are very natural developments within the ISP activity sector, with or without the explicit concurrence of the provider ISP. The provider ISP may see this reselling as an additional channel to market for its own Internet carriage services, and may adopt a positive stance by actively encouraging resellers into the market as a means of overall market stimulus, while tapping into the marketing, sales, and support resources of these reselling entities to continue to drive the volumes of the underlying Internet carriage service portfolio. The low barriers to entry to the wholesale market provide a means of increasing the scope of the operation, because to lift business cash-flow levels, the business enters into wholesale agreements that effectively resell the carriage components of the operation without the bundling of other services normally associated with the retail operation. This process allows the ISP to gain higher volumes of carriage capacity that in turn allow the ISP to gain access to lower unit costs of carriage.

Given that a retail operation can readily become a wholesale provider to third-party resellers at the effective discretion of the original retail client, is a wholesale transit ISP restricted from undertaking retail operations? Again, there is no such natural restriction from a technical or business perspective. An Internet carriage service is a commodity service that does not allow for a significant level of intrinsic product discrimination. The relatively low level of value added by a wholesale service operation implies a low unit rate of financial return for that operation. This low unit rate of financial return, together with an inability to competitively discriminate the wholesale product effectively, induces a wholesale provider into the retail sector as a means of improving the financial performance of the service operation. The overall result is that many ISPs operate both as clients and as providers. Few, if any, reasonable technical-based characterizations draw a clear and unambiguous distinction between a client and service provider when access services to networks are considered. A campus network may be a client of one or more service providers, while the network is also a service provider to campus users. Indeed most networks in a similar situation take on the dual role of client and provider, and the ability to resell an access service can extend to almost arbitrary depths of the reselling hierarchy. From this technical perspective, very few natural divisions of the market support a stable segmentation into exclusively wholesale and exclusively retail market sectors. The overall structure of roles is indicated in Figure 1.

Figure 1: ISP Roles and Relationships



The resultant business environment is one characterized by a reasonable degree of fluidity, in which no clear delineation of relative roles or markets exists. The ISP market environment is, therefore, one of competitive market forces in which each ISP tends to create a retail market presence. However, no ISP can operate in isolation. Each client has the expectation of universal and comprehensive reachability, such that any client of any other ISP can reach the client, and the client can reach a client of any other ISP. The client of an ISP is not undertaking a service contract that limits connectivity only to other clients of the same ISP. Because no provider can claim ubiquity of access, every provider relies on every other provider to complete the user-provided picture of comprehensive connectivity. Because of this dependent relationship, an individual provider's effort to provide substantially superior service quality may have little overall impact on the totality of client-delivered service quality. In a best-effort public Internet, the service quality becomes something that can be impacted negatively by poor local engineering but cannot be uniformly improved beyond the quality provided by the network's peers, and their peers in turn. Internet wholesale carriage services in such an environment are constrained to be a commodity service, in which scant opportunity exists for service-based differentiation. In the absence of service quality as an effective service discriminator, the wholesale activity becomes a price-based service with low levels of added value, or in other words a commodity market.

The implication in terms of ISP positioning is that the retail operation, rather than the wholesale activity, is the major area in which the ISP can provide discriminating service quality. Within the retail operation, the ISP can offer a wide variety of services with a set of associated service levels, and base a market positioning on factors other than commodity carriage pricing.

Accordingly, the environment of interconnection between ISPs does not break down into a well-ordered model of a set of wholesale carriage providers and associated retail service providers. The environment currently is one with a wide diversity of retail-oriented providers, where each provider may operate both as a retail service operator, and a wholesale carriage provider to other retailers.

Peer or Client?

One of the significant issues that arises here is: Can an objective determination be made of whether an ISP is a peer to, or a client of, another ISP? This is a critical question, because if a completely objective determination cannot be readily made, the question then becomes one of who is responsible for making a subjective determination, and on what basis.

This question is an inevitable outcome of the reselling environment, where the reseller starts to make multiple upstream service contracts, with a growing number of downstream clients of the reselling service. At this point, the business profile of the original reseller is little distinguished from that of the original provider. The original reseller sees no unique value being offered by the original upstream provider and may conclude that it is, in fact, adding value to the original upstream provider by offering the upstream provider high-volume carriage and close access to the reseller's client base. From the perspective of the original reseller, the roles have changed, and the reseller now perceives itself as a peer ISP to the original upstream ISP provider.

This assertion of role reversal is perhaps most significant when the generic interconnection environment is one of "zero-sum" financial settlement, in which the successful assertion by a client of a change from client to peer status results in the dropping of client service revenue without any net change in the cost base of the provider's operation. The party making the successful assertion of peer interconnection sees the opposite, with an immediate drop in the cost of the ISP operation with no net revenue change.

The traditional public regulatory resolution of such matters has been through an administrative process of "licensed" communications service providers, who become peer entities through a process of administrative fiat. In this model, an ISP becomes a licensed service provider through the payment of license fees to a communications regulatory body. The license then allows the service enterprise access to interconnection arrangements with other licensed providers. The determination of peer or client is now quite simple: A *client* is an entity that operates without such a carrier license, and a *peer* is one that has been granted such an instrument. However, such regulated environments are quite artificial in their delineation of the entities that operate within a market, and this regulatory process often acts as a strong disincentive to large-scale private investment, thereby placing the burden of underwriting the funding of service industries into the public sector. The regulatory environment is changing worldwide to shift the burden of communications infrastructure investment from the public sector, or from a uniquely positioned small segment of the private sector, to an environment that encourages widespread private investment. The Internet industry is at the leading edge of this trend, and the ISP domain typically operates within a deregulated valued-added communications service provider regulatory environment. Individual licenses are replaced with generic class licenses or similar deregulated structures in which formal applications or payments of license fees to operate in this domain are unnecessary. In such deregulated environments, no authoritative external entity makes the decision as to whether the relationship between two ISPs is that of a provider and client or that of peers.

If no public regulatory body wants to make such a determination, is there a comparable industry body that can undertake such a role? The early attempts of the *Commercial Internet eXchange* (CIX) arrangements in the United States in the early 1990s were based on a description of the infrastructure of each party, in which acknowledgments of peer capability were based on the operation of a national transit infrastructure of a minimum specified capability. This specification of peering within the CIX was subsequently modified so that CIX peer status for an ISP was simply based on payment of the CIX Association membership fee.

This CIX model was not one that intrinsically admitted bilateral peer relationships. The relationship was a multilateral one, in which each ISP executed a single agreement with the CIX Association and then effectively had the ability to peer with all other association member networks. The consequence of this multilateral arrangement is that the peering settlements can be regarded as an instance of “zero-sum” financial settlement peering, using a single-threshold pricing structure.

Other industry models use a functional peer specification. For example, if the ISP attaches to a nominated physical exchange structure, then the ISP is in a position to open bilateral negotiations with any other ISP also directly attached to the exchange structure. This model is inherently more flexible, as the bilateral exchange structure enables each represented ISP to make its own determination of whether to agree to a peer relationship or not with any other colocated ISP. This model also enables each bilateral peer arrangement to be executed individually, admitting the possibility of a wider diversity of financial settlement arrangements.

The bottom line is that a true peer relationship is based on the supposition that either party can terminate the interconnection relationship and that the other party does not consider such an action a competitively hostile act. If one party has a high reliance on the interconnection arrangement and the other does not, then the most stable business outcome is that this reliance is expressed in terms of a service contract with the other party, and a provider/client relationship is established. If a balance of mutual requirement exists between both parties, then a stable basis for a peer interconnection relationship also exists. Such a statement has no intrinsic metrics that allow the requirements to be quantified. Peering in such an environment is best expressed as the balance of perceptions, in which each party perceives an acceptable approximation of equal benefit in the interconnection relationship in its own terms.

This conclusion leads to the various tiers of accepted peering that are evident in the Internet today. Local ISPs see a rationale to viewing local competing ISPs as peers, and they still admit the need to purchase trunk transit services from one or more upstream ISPs under terms of a client contract with the trunk provider ISP. Trunk ISPs see an acceptable rationale in peering with ISPs with a similar role profile in trunk transit but perceive an inequality of relationship with local ISPs. The conclusion drawn here is that the structure of the Internet is one in which there is a strong business pressure to create a rich mesh of interconnection at various levels, and the architecture of interconnection structures is an important feature of the overall architecture of the public Internet.

Physical Interconnection Architectures: Exchanges and NAPs

One of the physical properties of electromagnetic propagation is that the power required to transmit an electromagnetic pulse over a distance varies in accordance with this distance. The shorter the distance between the transmitter and the receiver, the lower the transmission power budget required; *closer is cheaper*.

This statement holds true not only for electrical power budgets but also for data protocol efficiency. Minimizing the delay between the sender and receiver allows the protocol to operate faster and operate more efficiently as well; *closer is faster,* and *closer is more efficient.*

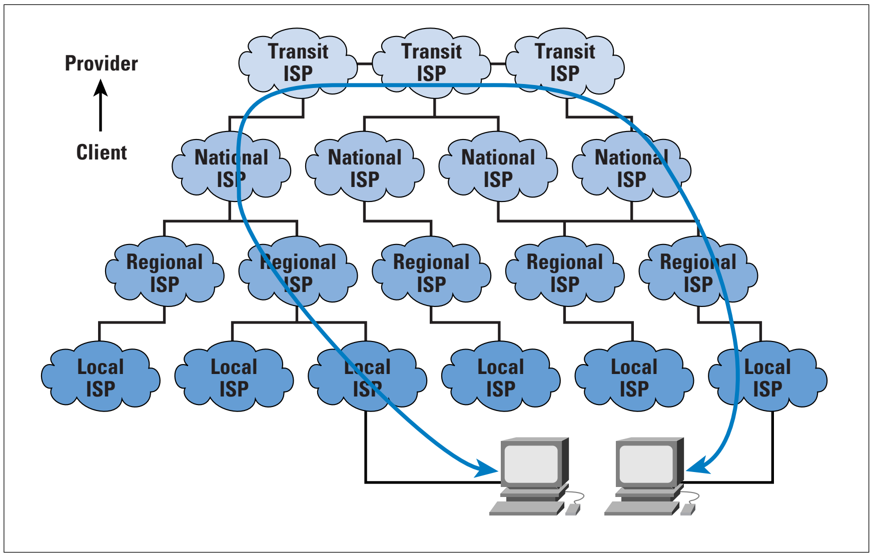
These observations imply that distinct and measurable advantages are gained by localizing data traffic; that is, by ensuring that the physical path traversed by the packets passed between the sender and the receiver is kept as physically short as possible. These advantages are realizable in terms of service performance, efficiency, and service cost.

How then are such considerations of locality factored into the structure of the Internet?

The Exchange Model

A strictly hierarchical model of Internet structure is one in which a small number of global ISP transit operators is at the "top;" a second tier is of national ISP operators; and a third tier consists of local ISPs. At each tier, the ISPs are clients of the tier above, as shown in Figure 2. If this hierarchical model is strictly adhered to, traffic between two local ISPs is forced to transit a national ISP, and traffic between two national ISPs transits a global ISP—even if both national ISPs operate within the same country. In the worst case, traffic between two local ISPs needs to transit a national ISP, then a global ISP from one hierarchy, then a second global ISP, and a second national ISP from an adjacent hierarchy in order to reach the other local ISP. If the two global providers interconnect at a remote location, the transit path of the traffic between these two local ISPs could be very long indeed.

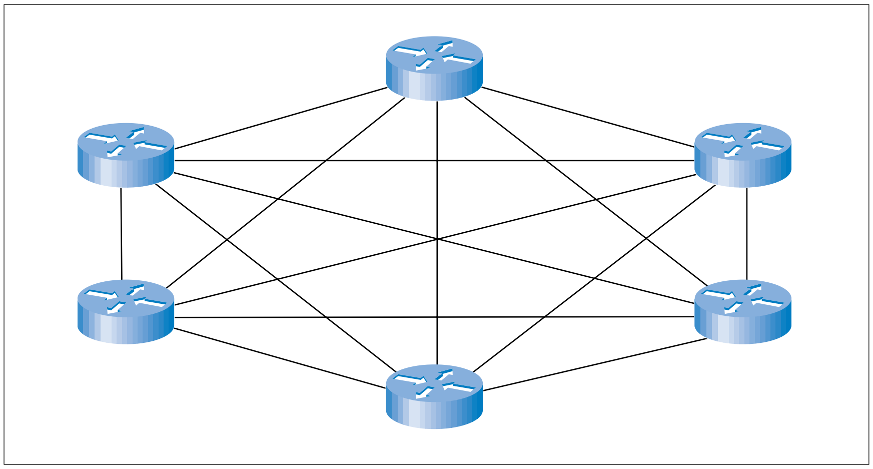
Figure 2: A Purely Hierarchical Structure for the Internet



As noted above, such extended paths are inefficient and costly, and such costs are ultimately part of the cost component of the price of Internet access. In an open, competitive market, strong pressure always is applied to reduce costs. Within a hierarchical ISP environment, strong pressure is applied for the two national providers, who operate within the same market domain, to modify this strict hierarchy and directly interconnect their networks. Such a local interconnection allows the two networks to service their mutual connectivity requirements without payment of transit costs to their respective global transit ISP providers. At the local level is a similar incentive for the local ISPs to reduce their cost base, and a local interconnection with other local ISPs would allow local traffic to be exchanged without the payment of transit costs to the respective transit providers.

Although constructing a general interconnection regime based on point-to-point bilateral connections is possible, this approach does not exhibit good scaling properties. Between *N* providers who want to interconnect, the outcome of such a model of single interconnecting circuits is $(N^2 - N) / 2$ circuits and $(N^2 - N) / 2$ routing interconnections, as indicated in Figure 3. Given that interconnections exhibit the greatest leverage within geographical local situations, simplifying this picture within the structure of a local exchange is possible. In this scenario, each provider draws a single circuit to the local exchange and then executes interconnections at this exchange location. Between *N* providers who want to interconnect, the same functionality of complete interconnection can be constructed using only *N* point-to-point circuits.

Figure 3: Fully Meshed Peering



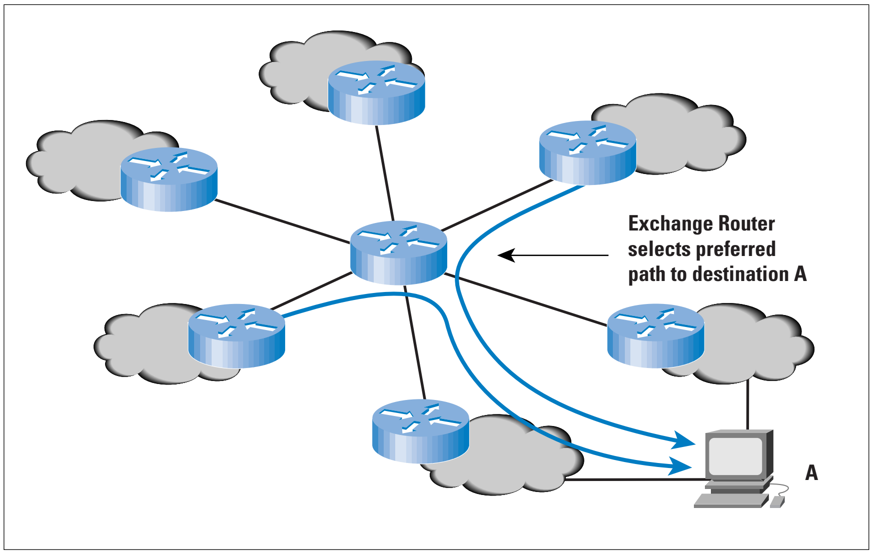
The Exchange Router

One model of an exchange is to build the exchange itself as a router, as indicated in Figure 4. Each provider's circuit terminates on the exchange router, and each provider's routing system peers with the routing process on the exchange router. This structure also simplifies the routing configuration, so that full interconnection of *N* providers is effected with *N* routing peer sessions. This simplification does allow greater levels of scaling in the interconnection architecture.

However, the exchange router model becomes an active component of the interconnect peering policy environment. In effect, each provider must execute a multilateral interconnection peering with all of the other connected providers. Selectively interconnecting with a subset of the providers present at such a router-based exchange is not easily achieved. In addition, this type of exchange must execute its own routing policy. When two or more providers are advertising a route to the same destination, the exchange router must execute a policy decision as to which provider's route is loaded in the router's forwarding table, making a policy choice of transit provider on behalf of all other exchange-connected providers.

Because the exchange is now an active policy element in the interconnection environment, the exchange is no longer completely neutral to all participants. This imposition on the providers may be seen as unacceptable, in that some of their ability to devise and execute an external transit policy is usurped by the exchange operator's policies.

Figure 4: An Exchange Router



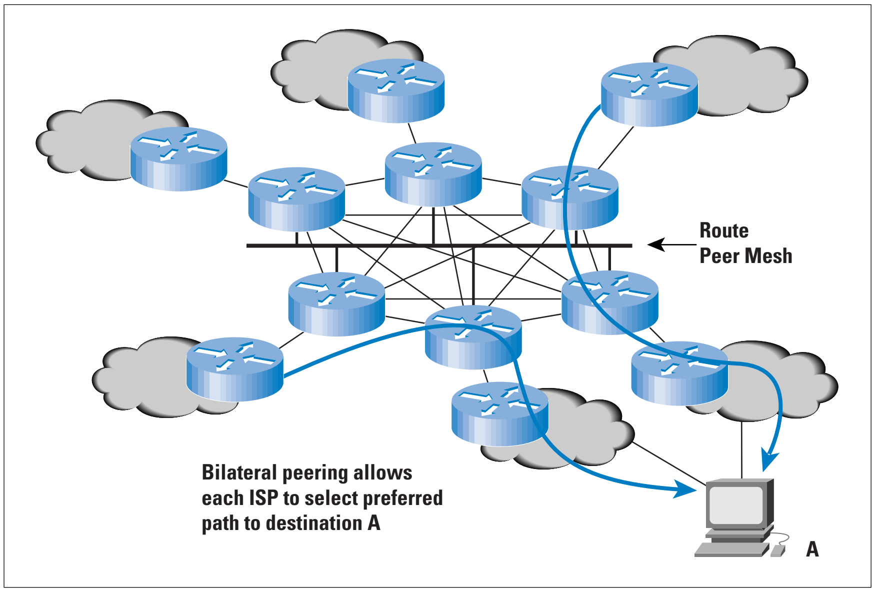
Typically, providers have a higher expectation of flexibility of policy determination from exchange structures than the base level of functionality that is provided by an exchange router. Providers want the flexibility to execute interconnections on a bilateral basis at the exchange, and to make policy decisions as to which provider to prefer when the same destination is advertised by multiple providers. They require the exchange to be neutral with respect to such individual routing policy decisions.

The Exchange Switch

The modification to the interprovider exchange structure is to use a local Layer 2 switch (or LAN) as the exchange element. In this model, a participating provider draws a circuit to the exchange and locates a dedicated router on the exchange LAN, as shown in Figure 5. Each provider executes a bilateral peering agreement with another provider by initiating a router peering session with the other party's router. When the same network destination is advertised by multiple peers, the provider can execute a policy-based preference as to which peer's route will be loaded in the local forwarding table. Such a structure preserves the cost efficiency of using N circuits to effect interconnection at the N provider exchange, while admitting the important policy flexibility provided by up to $(N^2 – N) / 2$ potential routing peer sessions.

Early interprovider exchanges were based on an Ethernet LAN as the common interconnection element. This physical structure was simple, and not all that robust under the pressures of growth as the LAN became congested.

Figure 5: An Exchange LAN



Subsequent refinements to the model have included the use of Ethernet switches as a higher capacity LAN, and the use of *Fiber Distributed Data Interface* (FDDI) rings, switched FDDI hubs, Fast Ethernet hubs, and switched Fast Ethernet hubs. Exchanges are very-high-traffic concentration points, and the desire to manage ever-higher traffic volumes has led to the adoption of Gigabit Ethernet switches as the current evolutionary technology step within such exchanges.

The model of the exchange colocation accommodates a model of diversity of access media, in which the provider's colocated router undertakes the media translation between the access link protocol and the common exchange protocol.

The local traffic exchange hub does represent a critical point of failure within the local Internet topology. Accordingly, the exchange should be engineered in the most resilient fashion possible, using standards associated with a premium quality data center. This structure may include multiple power utility connections, uninterruptible power supplies, multiple trunk fiber connections, and excellent site security measures.

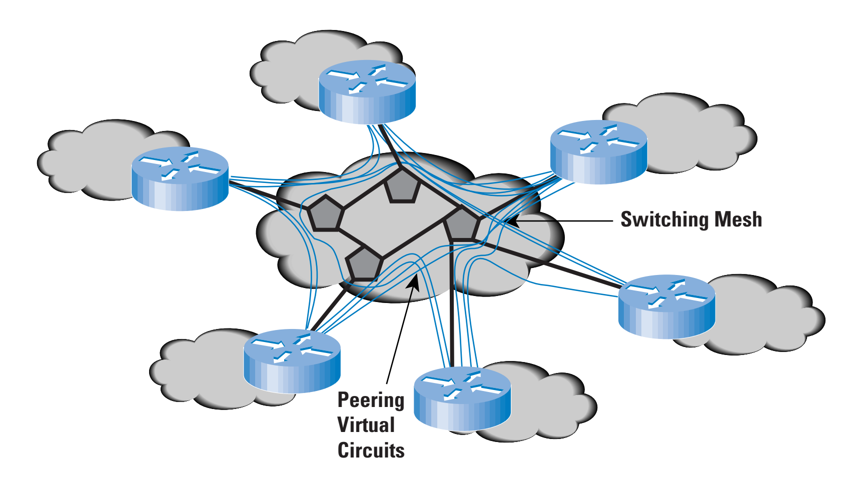
The exchange should operate neutrally with respect to every participating ISP, with the interests of all the exchange clients in mind. Thus, exchange facilities, which are operated by an entity that is not also a local or trunk ISP, enjoy higher levels of trust from the clients of the exchange.

There are also some drawbacks to an exchange, and a commonly cited example is that of imposed transit. If an exchange participant directs a default route to another exchange router, then in the absence of defensive mechanisms, the target router carries the imposed transit traffic even when there is no routing peering or business agreement between the two ISPs. Exchange-located routers do require careful configuration management to ensure that route peering and associated transit traffic matches the currently executed interconnection agreements.

Distributed Exchanges

Distributed exchange models also have been deployed in various locations. This deployment can be as simple as a metropolitan FDDI extension, in which the exchange comes to the provider's location rather than the reverse, as indicated in Figure 6. Other models that use an ATM-based switching fabric also have been deployed using *LAN Emulation* (LANE) to mimic the Layer 2 exchange switch functionality. Distributed exchange models attempt to address the significant cost of operating a single colocation environment with a high degree of resilience and security, but do so at a cost of enforcing the use of a uniform access technology between every distributed exchange participant.

Figure 6: A Distributed Exchange



However, the major challenge of such distributed models is that of switching speed. Switching requires some element of contention resolution, in which two ingress data elements that are addressed to a common egress path require the switch to detect the resource contention and then resolve it by serializing the egress. Switching, therefore, requires signaling, in which the switching element must inform the ingress element of switch contention. To increase the throughput of the switch, the latency of this signaling must be reduced. The dictates of increased switching speed have the corollary of requiring the switch to exist within the confines of a single location, if exchange performance is a paramount concern.

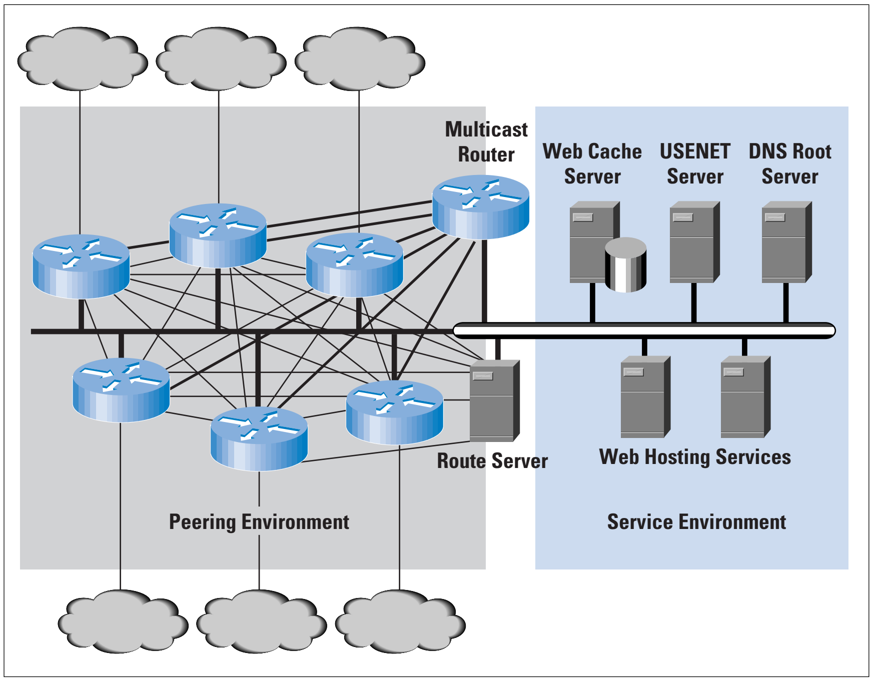
In addition to speed, the cost shift must be considered. In a distributed exchange model, the exchange operator operates the set of access circuits that form the distributed exchange. This process increases costs to providers, while it prevents the providers from using a specific access technology that matches their business requirements of cost and supportable traffic volume. Not surprisingly, to date the most prevalent form of exchange remains the third-party hosted colocation model. This model admits a high degree of diversity in access technologies, while still providing the substrate of an interconnection environment that can operate at high speed and therefore manage high traffic volumes.

Other Exchange-Located Services

The colocation environment is often broadened to include other functions, in addition to a pure routing and traffic exchange role. For a high-volume content provider, the exchange location offers minimal transit distance to a large user population distributed across multiple local service providers, as well as allowing the content provider to exercise a choice in selecting a nonlocal transit provider.

The exchange operator can also add value to the exchange environment by providing additional functions and services, as well as terminating providers' routers and large-volume content services. The exchange location within the overall network topology is an ideal location for hosting multicast services, because the location is optimal in terms of multicast carriage efficiency. Similarly, USENET trunk feed systems can exploit the local hub created by the exchange. The overall architecture of a colocation environment that permits value-added services, which can productively use the unique environment created at an exchange, is indicated in Figure 7.

Figure 7: Exchange-Located Service Platforms



Network Access Points

The role of the exchange was broadened with the introduction of the *Network Access Point* (NAP) in the architecture proposed by the National Science Foundation (NSF) in 1995 when the NSFNET backbone was being phased out.

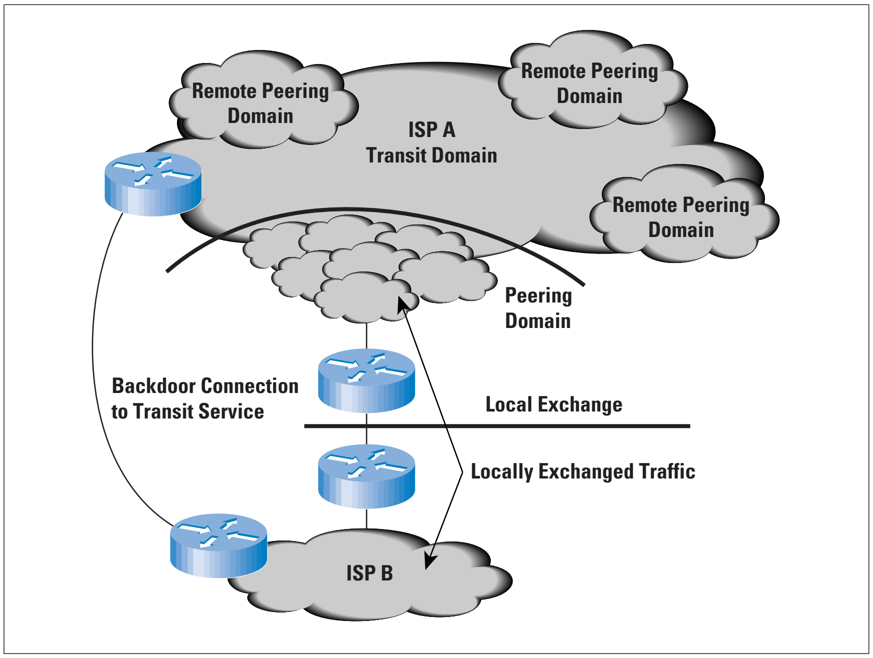
The NAP was seen to undertake two roles: the role of an exchange provider between regional ISPs who want to execute bilateral peering arrangements and the role of a transit purchase venue, in which regional ISPs could execute purchase agreements with one or more of a set of trunk carriage ISPs also connected at the NAP. The access point concept was intended to describe access to the trunk transit service.

This mixed role of both local exchange and transit operations leads to considerable operational complexity, in terms of the transit providers being able to execute a clear business agreement. What is the bandwidth of the purchased service in terms of requirements for trunk transit, versus the access requirements for exchange traffic? If a local ISP purchases a transit service at one of the NAPs, does that imply that the trunk provider is then obligated to present all the ISP's routes at remote NAPs as a peer? How can a trunk provider distinguish between traffic presented to it on behalf of a remote client versus traffic presented to it by a local service client?

The issue that the quality of the purchased transit service is colored by the quality of the service provided by the NAP operator should also be considered. Although the quality of the transit provider's network may remain constant, and the quality of the local ISP's network and ISP's NAP access circuit may be acceptable, the quality of the transit service may be negatively impacted by the quality of the NAP transit itself.

One common solution is to use the NAP colocation facility to execute transit purchase agreements and then use so-called *backdoor* connections for the transit service provision role. This usage restricts the NAP exchange network to a theoretically simpler local exchange role. Such a configuration is illustrated in Figure 8.

Figure 8: Peering and Transit Purchase



Exchange Business Models

For the ISP industry, many attributes are considered highly desirable for an exchange facility. The common model of an Internet exchange includes many, if not all, of the following elements:

- Operated by a neutral party who is not an ISP (to ensure fairness and neutrality in the operation of the exchange)
- Constructed in a robust and secure fashion
- Located in areas of high density of Internet market space
- Able to scale in size
- Operates in a fiscally sound and stable business fashion

A continuing concern exists about the performance of exchanges and the consequent issue of quality of services that traverse the exchange. Many of these concerns stem from an exchange business model that may not be adequately robust under pressures of growth from participating ISPs.

The exchange business models typically are based on a flat-fee structure. The most basic model uses a fee structure based on the number of rack units used by the ISP to colocate equipment at the exchange. When an exchange participant increases the amount of traffic presented over an access interface, under a flat-fee structure, this increased level of traffic is not accompanied by any increase in exchange fees. However, the greater traffic volumes do imply that the exchange itself is faced with a greater traffic load. This greater load places pressure on the exchange operator to deploy further equipment to augment the switching capacity, without any corresponding increase in revenue levels to the operator.

For an exchange operator to base tariffs on the access bandwidths is not altogether feasible, given that such access facilities are leased by the participating ISPs and the access bandwidth may not be known to the exchange operator. Nor is using a traffic-based funding model possible, because an exchange operator should refrain from monitoring individual ISP traffic across the exchange, given the unique position of the exchange operator. Accordingly, the exchange operator has to devise a fiscally prudent tariff structure at the outset that enables the exchange operator to accommodate large-scale traffic growth, while maintaining the highest possible traffic throughput levels.

Alternatively, there are business models in which the exchange is structured as a cooperative entity among numerous ISPs. In these models, the exchange is a nonprofit common asset of the cooperative body. Although widely used, these models are prone to the economic condition of the *Tragedy of the Commons*. It is in everyone's interest to maximize their exploitation of the exchange, while no single member wants to underwrite the financial responsibility for ensuring that the quality of the exchange itself is maintained.

The conclusion that can be drawn is that the exchange is an important component of Internet infrastructure, and the quality of the exchange is of paramount importance if it is to be of any relevance to ISPs. Using an independent exchange operator whose income is derived from the utility of the exchange is one way of ensuring that the exchange is managed proficiently and that the service quality is maintained for the ISP clients of the exchange.

A Structure for Connectivity

Enhancing the Internet infrastructure is quantified by the following objectives:

- Extension of reachability
- Enhancement of policy matching by ISPs
- Localization of connectivity
- Backup arrangements for reliability of operation
- Increasing capacity of connectivity
- Enhanced operational stability
- Creation of a rational structure of the connection environment to allow scalable structuring of the address and routing space in order to accommodate orderly growth

We have reached a critical point within the evolution of the Internet. The natural reaction of the various network service entities in response to the increasing number of ISPs will be to increase the complexity of the interconnection structure to preserve various direct connectivity requirements. Today, we are in the uncomfortable position of increasingly complex interprovider connectivity environments, a situation that is stressing the capability of available technologies and equipment. The inability to reach stable cost-distribution models in a transit arrangement creates an environment in which each ISP attempts to optimize its position by undertaking as many direct 1:1 connections with peer ISPs as it possibly can. Some of these connections are managed via the exchange structure. Many more are implemented as direct links between the two entities. Given the relative crudity of the inter-*Autonomous System* (AS) routing policy tools that we use today, this structure must be a source of considerable concern. The result of a combination of an increasingly complex mesh of inter-AS connections, together with very poor tools to manage the resultant routing space, is an increase in the overall instability of the Internet environment. In terms of meeting critical immediate objectives, however, such dire general predictions do not act as an effective deterrent to these actions.

The result is a situation in which the inter-AS space is the critical component of the Internet. This space can be viewed correctly as the *demilitarized zone* within the politics of today's ISP-based Internet. In the absence of any coherent policy, or even a commonly accepted set of practices, the lack of administration of this space is a source of paramount concern.

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. 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 a member of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, a collaboration with Paul Ferguson. Both books are published by John Wiley & Sons.
E-mail: `gih@telstra.net`

IPv6—What and Where It Is

by Robert L. Fink, Energy Sciences Network

The current Internet Protocol, known as IPv4 (for version 4), has served the Internet well for over 20 years, but is reaching the limits of its design. It is difficult to configure, it is running out of addressing space, and it provides no features for site renumbering to allow for an easy change of *Internet Service Provider* (ISP), among other limitations. Various mechanisms have been developed to alleviate these problems (for example, *Dynamic Host Configuration Protocol* [DHCP] and *Network Address Translation* [NAT]), but each has its own set of limitations.

The *Internet Engineering Task Force* (IETF) took on this problem in the early 1990s by starting an IPng (*Internet Protocol next generation*) project. After an over two-year-long process of defining goals and features, getting the best possible advice from industry and user experts, and sponsoring a protocol design competition, a new Internet Protocol was selected. Many proposed protocols were reviewed, analyzed, and evaluated. An evolved combination of several of them (*Simple Internet Protocol* [SIP], the *"P" Internet Protocol* [PIP], and *Simple Internet Protocol Plus* [SIPP]), each using fixed-length addressing, resulted in a final variation, called IPv6, which was selected over a version of the ISO OSI *Connectionless Network Protocol* (CLNP) (known as the *TCP and UDP with Bigger Addresses* (TUBA) IPng proposal).

Much work has been done since the selection of IPv6 in 1994. Over 50 implementations of IPv6 are believed to be under way or completed. A constantly growing international IPv6 testbed, called the *6bone,* now spans 260 sites in 39 countries, with over 25 different IPv6 implementations in use. Most router companies, including 3Com, Bay, Cisco Systems, Digital, Nokia, and Telebit support IPv6. IPv6 is also available for Digital, HP, IBM, Sun, WinTel, and many other end-user host systems.

IPv6 Addresses—Larger and Different

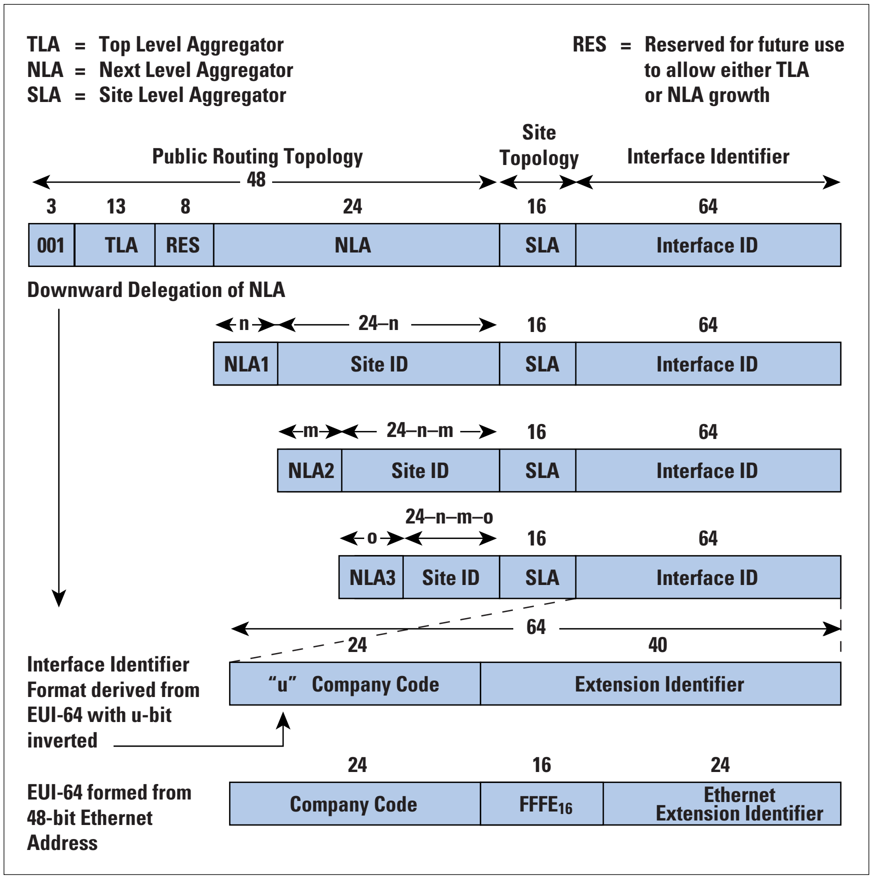
The larger 128-bit IPv6 address (versus the 32-bit IPv4 address) allows more flexibility in designing newer addressing architectures, as well as providing large enough address spaces for predicted future growth of the Internet and Internet-related technologies. A new addressing format, called the *Aggregatable Global Unicast Address Format*, has been developed to help solve route complexity scaling problems with the current IPv4 Internet. The current IPv4 provider-based addressing used in the Internet relies on separate IPv4 addresses being assigned to ISPs in contiguously numbered blocks for routing efficiency; that is, the routers need to carry fewer routes.

However, there is currently much fragmentation in the IPv4 address space. This situation, aggravated by sites not being able to easily renumber, causes many more separate routes than necessary, in turn leading to route computation complexity (too many routes, too many dynamic changes, too much computation in routers).

Public Routing Topology Prefixes

With the new aggregatable style addressing (see Figure 1), the left-most 48 bits of the address are defined as a *Public Routing Topology* (PRT) prefix. The first 3-bit field of this prefix specifies that the addressing format is aggregatable. The next 13-bit portion specifies the *Top Level Aggregator* (TLA) ID that constrains the top level of Internet routing to 8,192 major transit providers and a new concept of routing exchanges. Each TLA (top level transit ISP) is then responsible for all the remaining public routing topology assignment below it; that is, the *Next Level Aggregator* (NLA) ID. As shown in Figure 1, the NLA may have a tiered hierarchy to allow multiple levels (NLA1, NLA2, and so on) of other ISPs, each of which would then have control of the assignment of the space below it. The right-most portion of the NLA field, at whatever level it may be, would identify the end-user "leaf" site. An 8-bit reserved field has been defined to allow the growth of either the TLA or the NLA fields.

Figure 1: Aggregatable Global Unicast Address Format


TLA = Top Level Aggregator
NLA = Next Level Aggregator
SLA = Site Level Aggregator
RES = Reserved for future use to allow either TLA or NLA growth
Public Routing Topology
48
Site Topology
Interface Identifier
3
13
8
24
16
64
001
TLA
RES
NLA
SLA
Interface ID
Downward Delegation of NLA
n
24–n
16
64
NLA1
Site ID
SLA
Interface ID
m
24–n–m
16
64
NLA2
Site ID
SLA
Interface ID
o
24–n–m–o
16
64
NLA3
Site ID
SLA
Interface ID
64
24
40
Interface Identifier Format derived from EUI-64 with u-bit inverted
"u" Company Code
Extension Identifier
24
16
24
EUI-64 formed from 48-bit Ethernet Address
Company Code
FFFE16
Ethernet Extension Identifier

The advantage of this style of addressing is that it allows automatic address clustering, or aggregation, into a constrained set of routes, which are represented through the TLA field. If the initial assignment of 13 bits (8,192 TLAs) is insufficient in the future, either the reserved field or another piece of the IPv6 128-bit address space could be utilized. Note that only one-eighth of the current IPv6 address space has been assigned to aggregatable addressing.

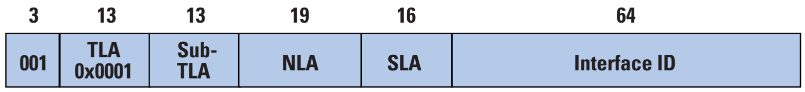
Even with this new concept of addressing, sites will still occasionally want to change their ISP (as in the current IPv4-based Internet) and thus will need to readdress to keep the addressing structure constrained. This is where *Site Renumbering,* which will be discussed later, comes in.

IPv6 TLA Assignment

To begin the production use of IPv6, ISPs providing IPv6 service need to be assigned TLAs so they may assign NLAs to transits and sites they are serving. Until recently, this was not possible. Recent discussions between the IETF, the IANA (*Internet Assigned Numbers Authority*), and the major address registries (APNIC, ARIN, and RIPE-NCC), have resulted in agreements that will provide a way to request and assign TLAs by early 2nd quarter 1999.

The process agreed upon is based on the above discussions that have been published as a recommendation in an Informational RFC on TLA assignments. The basic idea is to provide a slow start mechanism for TLAs by assigning one TLA ID to be used for defining a Sub-TLA field of 13 bits out of the reserved and NLA fields (see Figure 2). This will allow transits to demonstrate their need for a full TLA based on usage of the assigned Sub-TLA. These rules, based on much current practice with IPv4, are necessary to keep aggregatable addressing functional and effective for hierarchical routing as IPv6 comes into use.

Figure 2: Sub-TLA Format for IPV6 Address Assignment



3	13	13	19	16	64
001	TLA 0x0001	Sub- TLA	NLA	SLA	Interface ID

SUB-TLA IDs are assigned out of TLA ID 0x0001 as shown above.

Note that use of the Reserved field to create the Sub-TLA field is specific to TLA ID 0x0001. It does not effect any other TLA.

Rules for assigning these Sub-TLAs include:

- Must have a plan to offer native IPv6 service within three months from assignment; must have a verifiable track record providing Internet transit to other organizations
- Must make payment of a registration fee to the IANA and reasonable fees for services rendered by the address registry
- Must maintain registries of sites and next-level providers and make them available publicly and to the registries; must provide utilization statistics of NLA space below the assigned TLA (or Sub-TLA) and also show evidence of carrying TLA routing and transit traffic

These rules are intended to minimize route explosion and address assignment misuse to aid in the stability of the IPv6-based Internet.

Site Topology Prefixes

In addition to identifying the address of the site with the PRT prefix, aggregatable addressing provides for a site to have aggregation as well using a 16-bit *Site Level Aggregator* (SLA). The SLA might be as simple as a subnet number (more than 64,000 of them!), or a tiered hierarchy such as the NLA provides. However it is structured, the SLA is under the control of the site, and identifies the subnet that a host interface is attached to (IPv6's addressing, as IPv4's, specifies interfaces on systems, not the entire system).

It is very unlikely that an organization will ever need more than one PRT prefix, given the size and flexibility of the SLA and the *System Interface Identifier* field (described below).

System Interface Identifiers

Now that we have identified how to reach the site and the subnet a system is attached to, an interface identifier (ID) specifies the local logical address of the interface on the local subnet (or *link* as it is often called). The interface ID is formed and derived from the new IEEE EUI-64 media-level address that is an expansion of the well-known Ethernet 48-bit address format that allows for more device identifiers to be assigned by each manufacturer. The global/local bit is also inverted to make manually assigned (that is, local) addresses easy to form with only leading zeros.

If the IPv6 node is attached to an Ethernet "link," then the 48-bit address is turned into 64 bits by a filler field inserted in the middle (see Figure 1).

This enlarged Interface ID will allow newer technologies, such as *FireWire*, and newer applications, such as traffic lights and PCS/PDA telephones, to have unique interface identifiers assigned to them from a global address space.

The use of a media-level address for a network-level Interface ID allows the very important IPv6 Stateless Address Autoconfiguration Protocol to work.

Stateless Address Autoconfiguration

Automatic configuration of IPv6 end systems (hosts) is one of the most important features of IPv6. In the current IPv4 Internet, you must either manually configure IP address, network mask, and default gateway, or rely on having a DHCP server. With IPv6, this process can take place automatically, with no reliance on outside systems, using the IPv6 *Stateless Address Autoconfiguration Protocol*.

This can be done because the *Media Access Control* (MAC) address is used to form the host's interface ID. For example, if a host has an Ethernet interface that it is trying to configure for use with IPv6, the 48-bit Ethernet MAC address is formed into a 64-bit interface ID, which is the right-most 64 bits of the IPv6 address (see Figure 1). Then, using the *Neighbor Discovery* (ND) protocol, which is unique to IPv6, this formed interface ID is checked to see that it does not have a duplicate on this link (that is, subnet). If it does, a randomly generated token can be used (though a rare occurrence, it is a necessary protection against illegal Ethernet address usage and situations where the same address may be used on multiple interfaces for legitimate reasons).

At this point, an *ND Router Solicitation* multicast message is sent out to discover if there is a local IPv6 capable router, what the local site's topology ID for the host's subnet is, and what the site's public topology routing prefix is. Neighbor Discovery can also be used to control whether the site then wishes to continue with further configuration using Stateful Autoconfiguration with DHCPv6.

IPv6 Autoconfiguration thus provides for standalone operation of two or more hosts on a local LAN link with no router present, provides for operation within a site with no outside Internet connectivity present, and allows for easy changing of the site's public topology routing prefix, either when external connectivity comes on line, or when the external connectivity is changed, such as when a different ISP is chosen.

Domain Name System—Forward and Reverse

The *Domain Name System* (DNS) is an essential component of the Internet. To provide a mapping from a domain name to an IPv6 address, as well as an IPv4 address, a new DNS record type of "AAAA," or "quad A," is defined. This is a clever word play on the "A" record type that the original DNS specification defines for 32-bit IPv4 addresses, because IPv6 addresses are four times larger (128-bits), hence "AAAA"!

Most existing implementations of DNS already support AAAA records and existing IPv4 queries of DNS can access these records; that is, you don't need a DNS operating over IPv6 to retrieve these new AAAA records. This support also includes reverse lookups, similar to IPv4s, although a new reverse lookup proposal that will allow automatic partitioning of the delegation information on arbitrary bit boundaries is under consideration. This new capability should make for more reliable reverse registry than exists with IPv4, and easier maintenance when sites change their PRT prefix.

When a host with both IPv4 and IPv6 operating on it ("dual stack") queries the DNS for the address of a remote host, the A and AAAA records returned are used to indicate what protocol to use in communicating with that remote host. If no AAAA record is returned, IPv4 must be used. If only a AAAA record is returned, IPv6 must be used. If both A and AAAA are returned, either IPv4 or IPv6 may be used.

A new modification of the IPv6 DNS extensions is nearing completion that allows the automatic joining of the routing prefixes and Interface IDs when a host's IPv6 address is returned, thus making it easier to renumber a site. This new IPv6 DNS feature makes changing a site's PRT prefix (renumbering) very easy as only one entry, the PRT prefix, needs to be changed. This setup also facilitates easy support of multiple addresses for each host. These enhancements are very useful; IPv4 does not have this feature.

Renumbering Sites When ISPs Change

Because IPv6 addressing is based on the PRT prefix assigned by its ISP, it is essential that it be easy for a site to renumber itself when its choice of ISP changes. To aid in this, a new *Router Renumbering* (RR) protocol, in conjunction with Autoconfiguration, Neighbor Discovery and the new Aggregatable Unicast addressing PRT prefix are used.

RR allows a site's network administrator to set new PRT prefixes into the site's routers, as well as lower the lifetime of existing ISP PRT prefixes to specify an overlap interval, after which the old ISP's service is discontinued.

Hosts learn their new routing prefixes either when they restart, and thus are automatically configured with Autoconfiguration, or when they are informed by their local router that a new prefix is to be used during periodic router notification updates using ND.

For example, a new ISP service is readied for service while the old ISP is notified that it will provide service for just 60 more days. After the new PRT prefix is announced to the site's routers by RR, hosts will use the new prefix (that is, new ISP) for all new connections, while existing connections continue to work until the old prefix is withdrawn (that is, after 60 days in this example).

The easy renumbering of an IPv6 site will make easy a task that is currently very painful for an IPv4 site because hosts are often manually configured in many networks.

The 6bone—An IPv6 Testbed

The 6bone is an international IPv6 testbed network that is overseen and directed through the IETF *IPng Transition Working Group* (ngtrans) that provides:

- Testing of IPv6 implementations and standards
- Testing of IPv6 transition strategies
- A place to gain early applications and operations experience
- Motivation and a place for implementers, users, and ISPs to try IPv6
- An experimental first step toward transition

In the early phases of IPv6 deployment, most native IPv6 transport is restricted to site LANs with the ability to experiment with it locally. Some sites in Great Britain, The Netherlands, and Japan are using native IPv6 over WAN links.

ISPs and various other private IPv4 transit providers may not place IPv6 in their production routers in this early phase of IPv6 deployment, leaving early IPv6 testers with the need to use the existing IPv4 Internet infrastructure to deliver IPv6 packets among themselves when remotely located. Thus an IPv6 transition feature, IPv6 encapsulation (that is, *tunneling*) over IPv4, is used for parts of the 6bone where native IPv6 may not be available. In this way, the 6bone is also thoroughly testing out its own transition technology as well as providing IPv6 service.

The 6bone is a diverse community of users, ISPs, and developer organizations, many of whom provide transit on the public spirited basis of promoting and gaining early experience with IPv6. It is expected that production variations of the 6bone will also be created to more formally carry production IPv6 traffic.

Components of the 6bone

The 6bone provides this needed IPv6 transport over the public Internet infrastructure, relying on:

- Dual IPv4/IPv6 stacks in the client host
- IPv6 packets encapsulated (tunneled) in IPv4 packets
- Dual IPv4/IPv6 stack backbone routers that know IPv6 routes of 6bone participants
- DNS that supports IPv6 AAAA records
- A 6bone Routing Registry to keep track of sites and their tunnels
- A mailing list, various IPv6 tools, and a 6bone Web site at: **`www.6bone.net`**

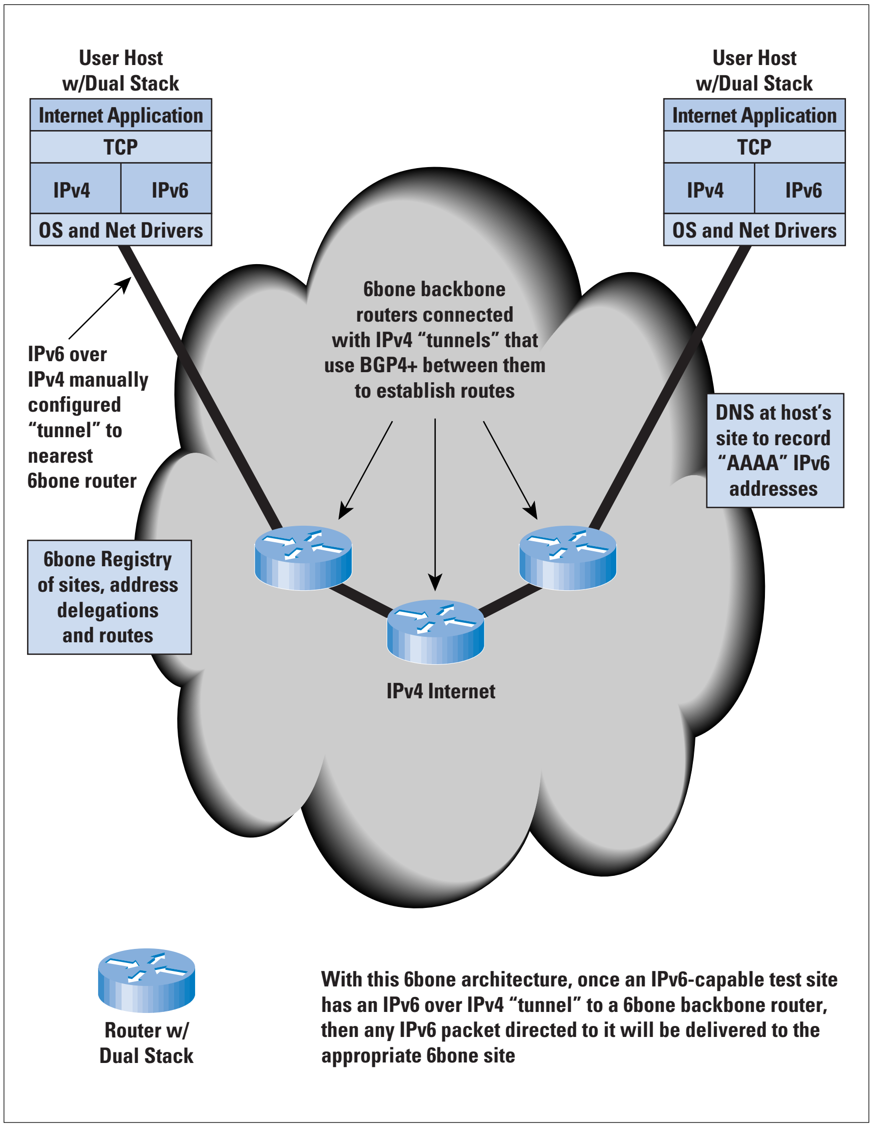
Figure 3 shows a conceptual overview of how a basic 6bone is structured and a picture of the current 6bone backbone structure can be seen at:

`http://www.cs-ipv6.lancs.ac.uk/ftp-archive/6Bone/Maps/full-backbone.gif`

…with the pseudo TLA site-to-site peering indicated by various colored links.

To date, the 6bone has spread to 260 organizations in 39 countries (see Table 1 on page 25).

Figure 3: 6bone Conc eptual Architecture


User Host w/Dual Stack
Internet Application
TCP
IPv4
IPv6
OS and Net Drivers
User Host w/Dual Stack
Internet Application
TCP
IPv4
IPv6
OS and Net Drivers
6bone backbone routers connected with IPv4 "tunnels" that use BGP4+ between them to establish routes
IPv6 over IPv4 manually configured "tunnel" to nearest 6bone router
DNS at host's site to record "AAAA" IPv6 addresses
6bone Registry of sites, address delegations and routes
IPv4 Internet
Router w/ Dual Stack

With this 6bone architecture, once an IPv6-capable test site has an IPv6 over IPv4 "tunnel" to a 6bone backbone router, then any IPv6 packet directed to it will be delivered to the appropriate 6bone site

6bone History

Serious work to evolve and refine the IPv6 protocols sufficient to allow the start of various implementations of IPv6 began in 1994. By early 1996, it was obvious that a testing environment was needed, so in March 1996, several implementers and users met and agreed to start an international testbed called the 6bone.

By June 1996, two groups raced to provide the first IPv6 connectivity: the University of Lisbon (Portugal), the Naval Research Laboratory (U.S.), and Cisco Systems (U.S.); a Danish universities consortium (UNI-C), a French universities consortium (G6), and a Japanese universities consortium (WIDE).

Table 1: Countries with Sites Participating in the 6bone

AT-Austria	FI-Finland	NL-The Netherlands
AU-Australia	FR-France	NO-Norway
BE-Belgium	GB-United Kingdom	PL-Poland
BG-Bulgaria	GR-Greece	PT-Portugal
BR-Brazil	HK-Hong Kong	RO-Romania
CA-Canada	HU-Hungary	RU-Russian Federation
CH-Switzerland	IE-Ireland	SE-Sweden
CM-Cameroon	IT-Italy	SG-Singapore
CN-China	JP-Japan	SI-Slovenia
CZ-Czech Republic	KR-Korea	SK-Slovakia
DE-Germany	KZ-Kazakhstan	TW-Taiwan
DK-Denmark	LT-Lithuania	US-United States
ES-Spain	MX-Mexico	ZA-Zaire

6bone Backbone and Addressing

By the end of 1997, the 6bone converted to the new aggregatable addressing format, a change necessitated by having originally adopted an early prototype provider-based addressing format discussed during early IPv6 design efforts.

Along with the change to a new addressing format was the need to clean up the routing used among the 6bone backbone transit sites. It was originally thought that IDRPv6 (a new Internet Domain Routing Protocol based on earlier IPv4 work) would be the prevailing *Exterior Gateway Protocol* (EGP) used for IPv6 Internet peering.

By mid 1996, various ISPs made it known that a new EGP for IPv6 was not a practical alternative, given the explosive growth of the Internet and the current evolution and widespread use of the *Border Gateway Protocol 4* (BGP4) by ISPs. There was a need to allow for multiprotocol extensions to BGP4, allowing ISPs to more easily adapt their operations to IPv6. This situation led to the rapid evolution of BGP4+, an extension of BGP4 to include IPv6 and IPv4 multiprotocol routing.

By mid 1997, the decision was made to convert the 6bone backbone to BGP4+ for its EGP. See **`http://www.cs-ipv6.lancs.ac.uk/ftp-archive/6Bone/Maps/full-backbone.gif`** for a recent picture of the 6bone backbone sites using the new aggregatable addressing format and the current status of the conversion to BGP4+.

6bone Future Plans

To date, most 6bone efforts have been to prove out basic IPv6 interoperability among the many implementations, and to create a reliable international testbed infrastructure. This has included making its backbone operationally ready with the new aggregatable addressing format and use of BGP4+ for high-reliability routing and transit.

Now that the 6bone has completed these conversions, serious work can begin on testing site renumbering, security, applications, and transition mechanisms.

Other IPv6 Trials and Testing

Other testing venues have also been very important to the evolution of IPv6: the University of New Hampshire *Inter Operability Laboratory* (IOL), various trade show demonstration networks, for example, NetWorld+Interop, and various vendor-sponsored interoperability testing.

By early 1998, the UNH IOL had hosted five IPv6 test sessions, though specific details about participating vendors are not released.

In a positive sign of industry response to evolving IPv6 specifications, the late July 1997 UNH testing resulted in the successful interoperability of all participants using the new aggregatable addressing format, no more than two months from its first Internet Draft.

Implementations

To date, over 50 different IPv6 host and router implementations are either completed or under way. More than 30 implementations have been tested and used on the 6bone.

Router implementations to date include: 3Com, Bay, Cisco Systems, Digital, Fujitsu LR550, Hitachi NR60, Inria BSD, Linux, Merit MRT, Nokia, NRL for BSD, Telebit, WIDE KAME and ZETA for BSD, and WIDE v6d.

Host implementations to date include: Apple MacOS OpenTransport demo version, Digital OpenVMS, Digital UNIX, FTP Software Windows95, Fujitsu LR450, 460, and 550, Hitachi NR60, IBM AIX, Inria BSD, Linux, HP-UX (SICS), Microsoft Research WindowsNT versions 4 and 5, Sony CSL Apertos IPv4/v6 stack, Sun Solaris, Trumpet Winsock for IPv6, UNH for BSD, NRL for BSD, WIDE KAME and ZETA for BSD, and WIDE v6d.

Several new Windows implementations that will operate under Windows95/98/NT are under way.

Transition from IPv4 to IPv6—A Seamless Approach

IPv6 is unlikely to become the Internet network-layer protocol of choice unless there is literally no choice to be made by the end user, little effort by network and system administrators, and it can operate alongside IPv4 for the indefinite future. Therefore, it must be very easy for the private network (your corporate net) and public network (your ISP) operators to equip, enable, and operate IPv6, while operating IPv4, in such a way that the user doesn't notice that IPv6 is there at all.

A system administrator, but not the user, must be conscious of IPv6 in a minimal sense. It is just another protocol stack that any Internet-based applications will operate over if the system is configured and distributed to do so by the system administrator.

At the network operator level, IPv6 is just another routing stack that can easily be turned on in the site's and ISP's routers (many sites certainly support IPX, AppleTalk, DECnet,...). IPv6 interdomain routing can be operated just like IPv4s because it uses BGP4+.

With the aid of the new *Dynamic DNS Registration Protocol* and IPv6's Stateless Autoconfiguration, users can boot up their system after it has been enabled with an IPv6 stack, in addition to its IPv4 stack, and become IPv6-ready without being aware of it at all. The system would automatically be configured with an IPv6 address, have itself registered automatically in the DNS with the host's existing name alongside its new IPv6 address (in addition to its DNS IPv4 address registration), and when finding a remote host with IPv6, start talking IPv6—all this without the user being required to consciously take action.

Early Production IPv6 Networks

In October of 1998, the *6REN* initiative, was established by the U.S. Energy Sciences Network (ESnet). The 6REN is a voluntary coordination initiative of *Research and Education Networks* (RENs) that provide production IPv6 transit service to facilitiate high quality, high performance, and operationally robust IPv6 networks.

The first participants were ESnet (the U.S. Dept. of Energy's Energy Sciences Network), Internet2 (the advanced Internetworking development collaboration comprised of many large U.S. research universities), CANARIE (the Canadian joint government and industry initiative for advanced networking), vBNS (the MCI network for NSF advanced networking) and WIDE (the Japanese research effort to establish a "Widely Integrated Distributed Environment").

Other profit and not-for-profit networks worldwide have been invited to join the 6REN. It is expected that during 1999 a sizable production environment capable of advanced demonstrations and deployment of Internet applications over IPv6 networks will be in place.

The Future for IPv6

It is too early to predict with total certainty that the Internet will adapt to the use of the IPv6 protocol. However, it should be obvious that IPv6 offers many important features for a next-generation Internet: automatic configuration, greatly expanded addressing, easy site renumbering, built-in security, and more.

One possible scenario for IPv6 is where it becomes the protocol of choice for newer applications not currently using Internet technology; for example, controlling traffic lights, reading electric meters, and so on. In these uses, IPv6 does not require coexistence with IPv4 because some form of gateway function would provide interconnection to the current Internet.

Another scenario (which doesn't exclude the previous one) is that Microsoft provides IPv6 support for a future version of Windows Networking on Windows OS, and promotes it within corporate America for its better features in supporting advanced corporate application/networking needs. In this scenario, the Internet will learn to carry IPv6 somehow, even if it is via automatically created tunnels that operate over IPv4 (somewhat similar to the 6bone's tunneling, but with dynamic creation of the tunnels as needed). It is expected that after Microsoft ships IPv6 and large corporations begin using it, ISPs will deploy IPv6 to get their business.

Yet another possibility is that the Internet telephony revolution will come to the conclusion that only IPv6 can provide cost-effective, scalable, end-to-end worldwide telephony implementations. This may be even more important as new classes of wireless networked devices, for example, PDAs and PCS phones, are integrated and built in very large volume.

Also, in parts of Asia and China, where there is little Internet connectivity at present, and very few IPv4 addresses assigned, IPv6 may become very popular because it will allow rapid growth without concerns about address space.

The probability is high that not just one of the above scenarios will happen, but that all will occur, in addition to others not yet imagined.

Whatever the implementation scenario, the probability that IPv6 will augment IPv4 as a part of the Internet of the future is very high!

References

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

[2] 6BONE information, including diagrams, hookup info, and registry access is at: **`http://www.6bone.net`**

[3] An IEEE EUI-64 overview can be found at: **`http://standards.ieee.org/db/oui/tutorials/EUI64.html`**

[4] "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[5] "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, December 1998.

[6] "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.

[7] "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)," RFC 2463, December 1998.

[8] "IP Version 6 Addressing Architecture," RFC 2373, July 1998.

[9] "An IPv6 Aggregatable Global Unicast Address Format," RFC 2374, July 1998.

[10] "DNS Extensions to support IP version 6," RFC 1886, December 1995.

[11] "Proposed TLA and NLA Assignment Rules," RFC 2374, December 1998.

[12] "Transition Mechanisms for IPv6 Hosts and Routers," RFC 1933, April 1996.

[13] "Router Renumbering for IPv6," Internet Draft, Work in Progress, **`draft-ietf-ipngwg-router-renum-06.txt`**, November 1998.

[14] *IPv6: The New Internet Protocol,* Christian Huitema, ISBN 0-13-850505-5, Prentice Hall, 1998.

[15] *IPng: Internet Protocol Next Generation,* Edited by Scott O. Bradner and Allison Mankin, ISBN 0-201-63395-7, Addison-Wesley, 1996.

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

Secure E-Mail: Problems, Standards, and Prospects

by Marshall T. Rose and David Strom

As we spend more and more time using e-mail, most of us eventually find that we need to be able to prove our identity to our correspondents and secure the contents of our messages so that others can't view them readily. Proving your identity is called *authentication*. In the physical world, this is accomplished by photo identification, such as a driver's license, passport, or corporate identity card. When the time comes to prove who you are (for example, before a major purchase), you show your card. Your appearance and signature match the photo and signature on your card, and the purchase is made.

On the Internet, however, the process isn't as easy. Does e-mail from **`sidney@example.com`** really originate from our friend Sidney at the Example Corporation? Maybe it's from someone else, who just happens to be using Sidney's machine when he is out to lunch. Or, worse, someone trying to impersonate Sidney illicitly. And even if the message actually is from the "real" Sidney, how can we be sure: Is there an electronic analog to a signature?

Most of us are trusting individuals; we tend to believe that people are who they say they are unless we have particular reasons to doubt their identity. But on the Internet, we have to look beyond face value. And proving that someone indeed did send a particular message is a very difficult problem.

This may be one of the main reasons why corporations employ Lotus Notes and other Internet-based messaging systems that are not 100-percent pure. They want to ensure that all messages carry the appropriate authentication with them at all times. In order for new users of Notes to start using the software, they must first obtain an electronic certificate that authenticates them to the system. The certificate is created by the Notes system administrator, who works in conjunction with that particular Notes server owned by that particular corporation.

Securing the message contents is also a challenge: all e-mail sent over the Internet, unless otherwise protected, is sent in clear ASCII text. If you have the tools, the time, and the technical expertise, you can capture this traffic and read anyone's correspondence. It isn't simple, but it is quite possible.

Besides being sent as clear text, e-mail can also be intercepted and its contents changed between the time the sender composes the message and the recipient reads it. Again, this task is neither likely nor simple, but it can be accomplished if someone is determined enough to do it. Therefore, senders can neither prove nor deny that they sent a particular message to you; it could be real or a forgery, and you have no way of knowing which.

Cryptography Standards

It would be great if we could say that the future for secure e-mail is bright, and that there will be standards in place that will help. However, the state of secure e-mail standards for the Internet is best described as a "terrible mess"! (Ed.: a less charitable phrase is used in the book from which this material is adopted.) Think that characterization is unprofessional? It is actually quite detached, considering the amount of culpability enjoyed by the principals of the Internet's secure e-mail debacle. We would love to write an article describing the high crimes and misdemeanors of these scoundrels, but that would only publicize the guilty, not punish them. So, instead we'll survey the horizon and try to make sense of what little terrain there is.[1]

In brief, no technologies for secure e-mail in the Internet meet all of the following criteria:

- Multivendor
- Interoperable
- Approved or endorsed by the Internet's standardization body

There are two competing technologies, each of which satisfies at most one of these criteria. However, for any 100-percent-pure Internet solution to succeed, we feel it must be based on technologies that satisfy all three.

Basic Concepts

In order to understand secure e-mail, you need to know only three concepts:

- Data encryption (privacy)
- Message integrity (authentication)
- Key management

Everything else is a matter of data formats.

Data Encryption

When the contents of a message are to be protected from third-party disclosure, it is necessary to agree upon an encryption algorithm. Because cryptographic algorithms are constantly being scrutinized, a secure e-mail standard must be extensible with respect to the algorithms that it allows.

Historically, *symmetric encryption algorithms* are used for this purpose. A symmetric algorithm is one in which the same key is used to both encrypt and decrypt the data. Symmetric algorithms are chosen because they are computationally less burdensome (in other words, faster to execute) than asymmetric algorithms.

As such, each time a message is to be encrypted, a new session key is generated for that purpose. Although one could send the session key via some secure path, it is easier to include the session key along with the message, but encrypted so that only the intended recipient can decipher it. Upon deciphering the session key, the recipient can apply the encryption algorithm and retrieve the original contents.

For example, Network Associates' Pretty Good Privacy (PGP), one of the two technologies we'll examine, uses an asymmetric algorithm to encrypt the session key and a symmetric algorithm to encrypt the user's data.

Message Integrity

When the contents of a message are to be verified as authored by a particular user and unaltered by any other user, it is necessary to agree upon a *signature* and *hash algorithm*. The former is used to verify the authenticity of the message, and the latter is used to verify the integrity of the message. Again, any secure e-mail standard must be extensible with respect to the algorithms that it uses for these purposes.

For signature algorithms, asymmetric algorithms are typically used. These algorithms utilize a public key and a secret key. A signature algorithm combined with a secret key allows someone to generate a digital signature for the contents of a message. A signature algorithm combined with a public key allows someone to verify the digital signature for a message. As you might expect, signature algorithms are one-way functions: You can't reconstruct the input to a signature function by looking at its output.

Hash algorithms are often called *message digest algorithms*. They simply compute a checksum on their input; no keys are involved. Hash algorithms are also one-way functions, and a good hash algorithm is one in which very similar inputs produce dramatically different outputs. Hence, if even a single bit is altered or corrupted in transit, the hash value will be different.

Key Management

All discussion now hinges on how keys are used for asymmetric algorithms. Specifically, how do you trust the identity of the secret key used to make a digital signature? To start, we have to introduce the notion of a *public key certificate*. Although the actual formats vary, at its heart a certificate contains three things:

- The identity of the "owner" of the certificate
- A public key
- Zero or more guarantees to the validity of the binding between the identity contained in the key and the owner in the "real world"

So, the next step is to ask what these identities and guarantees look like. Unfortunately, we now enter the realm of sociology rather than technology. The only theoretical limitation on an identity is that you have to be able to represent it digitally. It could be a name (for example, "Jim Bidzos") or an e-mail address (for example, **`prz@pgp.com`**) or a key in some database (for example, the name of an object in a directory). More interesting examples could include a series of assertions (for example, your driver's license number is this, your passport number is that, and so on).

Fortunately, the guarantees are a bit simpler to describe—they are digital signatures from other public keys that vouch for the veracity of the binding. For example, if you encountered a public key certificate in which the identity was someone's passport number, it would be natural to expect that the certificate contains a digital signature from the government entity (or its agent) that issued the passport. However, this begs another question: Why should you trust the entities that have signed someone's public key? It turns out that our two contending technologies have different answers to that question.

As you might expect, certificates have some additional properties, such as a date the certificate becomes valid, the date the certificate expires, and a "fingerprint." The fingerprint is simply a hash of the identity and public key so you can tell if it has been altered in transit.

Finally, *certificate revocation lists* identify certificates that are no longer valid. For example, if the secret key associated with a certificate is accidentally disclosed, then the corresponding certificate is revoked.

Pretty Good Privacy: The Web of Trust

Pretty Good Privacy (PGP) is encryption for the masses. Despite the fact that it required a couple of complete rewrites in order to achieve stability, it gets the job done.

An effort is under way to provide a "standards-based" version of the PGP technology, termed *OpenPGP*. The "pre-standards" version of PGP uses the RSA algorithm for signatures and the IDEA algorithm for encryption. The version being developed is more flexible with respect to the algorithms it supports.

The most remarkable thing about PGP is its trust model. Remember the earlier question: How do you know whether you should believe the identity in a public key certificate? To answer this in the context of PGP, each user assigns two attributes to the PGP certificates that they encounter: *trust* and *validity*. Trust is a measure as to how accurate the certificate's owner is with respect to signing other certificates. Validity indicates whether or not you think the identity in the certificate refers to the certificate's owner.

So, initially your local collection of certificates starts out with one—your own PGP certificate. You then sign your friend's certificate and he or she signs yours. Because you trust yourself when signing those certificates, your friend's certificates are automatically considered valid. Then, based on your judgment of your friend's abilities to sign other certificates accurately, you assign a level of trust to his or her PGP certificates. As you receive messages containing other people's certificates, if they are signed by you, or any of your trustworthy friends, they are automatically deemed valid. This organic, highly decentralized approach toward validating public key certificates is termed the *web of trust*.

Key servers are also available that are repositories of PGP certificates. If you need to send e-mail to someone, but don't have his or her certificate, you can query a server to see if a copy is there. Of course, the usual rules apply with respect to assigning trust and validity—it's up to you! Key servers also help when you receive e-mail from someone new. Although the message will contain a copy of someone's PGP certificate, you may not know about any of the signatories. So, you can go to a key server and fetch the certificates for the signatories; you might decide to trust them after seeing who signed their certificates.

We've simplified the web of trust in that validity isn't "all or nothing," as we implied previously. Rather, PGP offers a flexibility spectrum of possibilities; for example, requiring two trustworthy signatories before considering a certificate to be valid. But the one thing that should be clear is that trust and validity are *different*. You will probably have many keys in your local collection of certificates that are considered valid, but probably only a few of those will be considered authorized to vouch for others.

Secure MIME: The Hierarchy of Trust

There is an interesting concept in advertising called "ambush marketing." The basic idea is that your advertising campaign leverages off the brand and promotion of a competitor. *Secure Multipurpose Internet Mail Extensions,* or S/MIME, is an example of ambush marketing in the Internet. Although MIME is an Internet standard, which has been implemented by hundreds of vendors and provisioned in tens of thousands of networks, S/MIME is the product of a closed vendor consortium.

S/MIME has two versions: version 2 and version 3. As of this writing, products that claim to implement S/MIME implement version 2. They use the RSA algorithm for signatures and a weak algorithm for encryption (RC2 with 40-bit keys). An effort is under way to provide a "standards-based" version of the S/MIME technology—version 3. The version being developed is more flexible with respect to the algorithms it supports. S/MIME uses a hierarchical model for establishing trust. For example, if your employer assigns you an S/MIME certificate, he will act as a certification authority and sign that certificate. As a consequence, trust is established on the basis of a hierarchical relationship between the *subject* of a certificate (the identity) and the *issuer* (the signatory).

This model has some strengths: users rely on the certification authorities implicitly. However, a bootstrapping problem still exists: How do you know to trust the issuer? The answer is that your local collection of certificates also has some "top-level" certificate authorities, and it is these authorities that sign the public key certificates of the issuers. If the hierarchy of trust can be kept to one or two levels, this is manageable in practice.

The web and hierarchical models of trust share many attributes in common. For example, when you receive a message, it contains a copy of the certificate that was used to make the digital signature. If you aren't familiar with the signatories, you can look in a remote repository of keys. The only difference between the two models here is that the hierarchical model needs key servers to make its key infrastructure work. Because of this, keys are usually stored in a directory service accessed via the *Lightweight Directory Access Protocol* (LDAP).

Data Formats

The **`multipart/encrypted`** and **`multipart/signed`** contents are used to convey secure e-mail. Fortunately, they are both very simple content types.

A **`multipart/signed`** content has two subordinate body parts. The first contains the data that is being authenticated and can be any MIME content type (**`text/HTML`**, **`multipart/mixed`**, and so on). The second contains the digital signature used to authenticate the content. The **`multipart/signed`** content has two mandatory parameters. The *protocol* parameter defines the technology used to generate the digital signature, and the *micalg* (for "MIC algorithm") parameter defines the hashing algorithm used (for "MIC" read: *message integrity check*). The value of the protocol parameter is also the content type used for the second body part. The only tricky part is that the digital signature is calculated on the data before a transfer encoding, if any, is applied.

Let's make this a little more concrete. If we assume that the OpenPGP effort produces an Internet standard based on the current draft (a reasonable assumption at 50,000 feet), then the structure of a **`multipart/signed`** message created using PGP technology would look like the following:

- The protocol parameter would be **`application/pgp-signature`**
- The micalg parameter would be **`pgp-md5`**
- The first body part would be labeled as whatever you wanted to sign
- The second body part would labeled as **`application/pgp-signature`**

The second body part, a data structure defined by the OpenPGP document, contains the digital signature along with any supporting material (for example, a copy of the sender's PGP certificate).

Note that you don't encrypt the first body part in a **`multipart/signed`** content. In this way, if only some of your recipients have secure e-mail, but you still want to sign it for those who do, everyone can still read the first body part.

A **`multipart/encrypted`** content has two subordinate body parts. The first contains the information needed to decipher the encrypted data (for example, the encrypted session key along with an indication as to the certificate needed to decipher the session key). The second contains the encrypted data, labeled as **`application/octet-stream`**. The **`multipart/encrypted`** content has one mandatory parameter, protocol, which defines the technology used to encrypt the data. The value of the protocol parameter is also the content type used for the first body part.

To further define this concept, if we use OpenPGP as the basis for a hypothetical example, then the structure of a **`multipart/encrypted`** would look like the following:

- The protocol parameter would be **`application/pgp-encrypted`**
- The first body part would be labeled as **`application/pgp-encrypted`**
- The second body part would labeled as **`application/octet-stream`**. In practice, the input to the encryption algorithm would be **`multipart/signed`** .

Finally, one or more MIME content types might be defined for sending certificates, certificate revocation lists, and so on. These are all specific to the particular secure e-mail technology being used.

Encrypting Your Messages

If we look at popular commercial e-mail products, many of them include support for some kind of encryption. Both Microsoft's Outlook Express and Netscape Messenger include support for S/MIME, although we'll see in a moment that the two have radically different capabilities. And Qualcomm's Eudora Pro package comes with an add-on module for supporting PGP, which you may or may not have installed when you installed the software. In order to encrypt a message, you need to go through the following process:

1. Choose which of the two competing technologies (and specific e-mail software) you wish to use for your encrypted correspondence. Both methods have advantages and disadvantages.
2. Choose whether you want to just digitally sign your messages or encrypt their entire contents, or both.
3. Either choose an enterprise certificate authority and set up the appropriate server software, or obtain a certificate from a public authority. Again, both methods have advantages and disadvantages.
4. Enroll with this certificate authority and obtain an encryption certificate or key for a particular machine and a single e-mail address.

5. Exchange keys with your correspondents, and manage where these keys are stored on your machine.
6. Encrypt and decrypt messages.

If this process seems rather involved and complex, it is. The process is not nearly where it should be to enable encryption to be useful by most e-mail users, and won't be for some time. If all of this seems overwhelming to you, we certainly understand.[2] It is to us, too! But let's go through these six steps in more detail.

PGP vs. S/MIME

Our discussion in the standards section might have convinced you that encryption technology is still very much a work in progress, and after you begin to use the encryption features of your own e-mail software, you'll be further convinced. Nevertheless, unless you plan to test lots of different software products, you should first decide on which product and which encryption technology you intend to use. You definitely want to limit yourself to as small a universe as possible, because running more than one e-mail software product will only make your encryption life miserable. So which to choose?

PGP is everyman's product. It was designed for single individuals to use and still remains the easiest method to set up and get going, although it is far from simple. The version of PGP that comes with the Eudora Pro box is the individual version; a separate and more capable version is available for workgroups or businesses, called *PGP for Business Security*. This business version is the one we recommend, even if you are the only person in your corporation that will use encryption. You'll find that after you start, others will follow, and you might as well start off with the more capable version.

If you want to use PGP, you will need to run a separate piece of software to encrypt and decrypt your messages. If you already use software such as Messenger or Outlook Express, that is certainly more cumbersome than using the built-in S/MIME features of those two products.

In 1999, PGP is more capable than S/MIME when it comes to setting up an enterprise encryption policy and putting it into practice on a daily basis. For example, with PGP you could establish that all outgoing and incoming encrypted messages are first copied to a special archive, and that all outgoing messages are encrypted with a special administrator's key that can be used in an emergency to read the message if the sender forgets his key or leaves the company. S/MIME doesn't have this ability yet, although this feature is being developed for the future.

PGP is a single-vendor solution: All your software must eventually come from Network Associates to run the various certificate servers and encryption modules. With S/MIME, you'll have some degree of choice, although we found that in practice you probably want to make use of the same e-mail product when exchanging encrypted messages if you want them to be read with a minimum of difficulty. Not all S/MIME packages can exchange encrypted messages with each other because of differences in their implementations. When Dan Backman of *Network Computing* magazine tested five different products, he found several that couldn't read messages sent by others.[3]

Part of the problem with S/MIME is the various choices of "strength" of cryptographic algorithms that are in use in today's browsers and e-mail software. This debate is more about politics than technology, because the U.S. government places restrictions on various algorithms, as mentioned earlier. Two different parameters are of interest: the length of the key itself used in any certificate and the type of encryption technology used. Netscape software supports key lengths ranging from 512 to 1024 bits, for example. In addition, several choices are available for encryption technology; they are labeled *RC2* (which can either be 40-bit encryption, the only one allowed for export by the U.S. government, or more complex encryption of 64, 128, or even 255 bits), and *Data Encryption Standard* (DES). RSA, Inc., developed RC2. On the other hand, the U.S. government developed DES. Debate abounds as to which is the better or more or less proprietary technology.

These details are outside the scope of this article, but you should know that the larger the key size and encryption algorithm, the more difficult it is for someone to decode an intercepted message.

Digital Signature Required?

Your next choice is to consider whether to just make use of a digital signature, to encrypt the entire message, or to make use of both technologies. All encryption products can do both, but in somewhat different ways.

Digital signatures guarantee that your recipients have received your message without any tampering and that they can trust that the message came from you. The actual message body, and any attachments, arrive without any encryption, meaning that someone could still capture this traffic and read your correspondence. You might want to use a digital signature without encrypting the message, if you care that your message was received intact and that your correspondents can know that you sent it.

There are two different types of signed messages: *clear* and *opaque*. With clear-signed messages, you can still read the message text, even if you don't have any encryption functions in your e-mail software. The signature is carried along with the message in a separate MIME portion of the message from the message body, which remains untouched and still readable. This feature can be handy, especially if you correspond with many people and they probably haven't adopted any particular encryption product, or if they are using older versions of e-mail software that don't support encryption. Clear signing is also useful in circumstances where your encryption technology isn't compatible with your correspondents' technology. PGP supports only clear signing in its products.

One problem with clear signing is e-mail gateways. They often will break the encryption of the signature, because they will either add or remove characters from the message, and that sloppiness could invalidate the signature block. After all, part of the role of the signature is to ensure that the message was delivered intact and unaltered!

Opaque signing means that your recipients will get a blank message if they aren't running any encryption software, or if their encryption software doesn't work with yours. Opaque signing wraps the entire message in a Base64 encoding, which is usually left alone by most e-mail gateways. This encoded message then gets transmitted and then decoded by the S/MIME recipient.

PGP places its signature inside the encrypted envelope when it sends messages, making it difficult to determine the signature of such a message until you first decrypt it. The PGP producers claim that this feature offers extra protection in case the message is compromised or copied en route. Newer versions of PGP offer a MIME option that places the signature outside the encrypted envelope. This is how S/MIME products work, making it easier to determine who sent it.

Choose Your Certificate Authority

Now you have another decision to face, and that is how to set up what is called the *certificate authority* (CA) for your enterprise. This software runs on a UNIX or NT server and manages the keys or certificates of everyone in your corporation. It serves as a central place of trust and signs all of your users' certificates. If you trust your CA, in theory you should be able to trust the certificates that are signed by the CA, called *inherited trust*.

The problem is that there isn't any "central" CA for the entire universe of e-mail users. Although there are several public CAs that anyone can use, either for free or for a fee, they don't necessarily trust each other, nor should they. What happens if an employee of VeriSign becomes disgruntled and starts issuing bad certificates? There should be checks and audits to ensure that these types of problems can't undermine the entire CA system, just as there are checks and audits to prevent rogue banking employees from crediting their own accounts.

Setting up a CA is the beginning of setting up a very complex security infrastructure for your enterprise. Your CA needs to establish a link of trust from all your users to the administrator or operator of the CA itself, and from your CA to other CAs with which you communicate.

There are two different kinds of CAs: One uses software that you install on your own server inside your enterprise and you maintain; the other is public servers. Having your own server places the burden on creating and revoking certificates on your security administrator, or whoever is going to operate the CA server. In many cases, these products can be administered from a Web browser after they are installed, and the servers can handle certificates from a wide variety of S/MIME products, one of the few shining spots on the interoperability scene at the moment.

PGP for Business comes with its own version of a certificate server. It runs on a Windows desktop machine and typically is used by the administrator of the entire security apparatus to handle certificates. It can handle only PGP certificates.

Some popular software products that function as certificate servers are listed below.

Vendor	URL	Product
Enterprise CAs:		
Netscape	`www.netscape.com`	Certificate Server
Xcert	`www.xcert.com`	Sentry CA
Public CAs:		
VeriSign	`www.verisign.com`	Secure Server ID
Thawte	`www.thawte.com`	Public CA

Enroll and Acquire Your Certificate

When you have your certificate authority either in mind or installed, you next have to set up how you want to acquire your own certificate.

You have two broad methods: by Web or by e-mail. Actually, you don’t have any choice: If you have picked your e-mail product and CA at this point in the process, you have to use whatever method comes with that choice. Netscape Messenger and Microsoft’s Outlook Express, among others, make use of their related Web browsers to enroll certificates, as you might suspect. And other products make use of e-mail to send and enroll certificates. For example, Xcert’s Sentry CA sends you a message telling you that your certificate has been granted, but in the e-mail it has URLs for both Communicator and Internet Explorer where you can download the certificate and place it inside the appropriate software. Why two different links? Because each product supports a different way of acquiring certificates, of course. So much for standards.

Exchange and Manage Certificates

Now comes the hard part—dealing with the certificates of your correspondents, and managing both theirs as well as other certificates around your corporation.

As we mentioned in our standards section, you need to exchange certificates with your correspondents before you can begin to exchange encrypted e-mail. And that means sending your public key to them, and getting their public keys from them, before you can exchange actual encrypted messages. If you are corresponding with someone who doesn't have the same CA in common, you'll first need to establish a trust relationship and exchange root CA certificates before you can exchange the individual certificates. This is somewhat painful, but when you get the hang of it, it isn't that difficult.

After you begin to exchange more than a few of these certificates, you might think that this is a job for a directory server, and, thankfully, the vendors are already there. The CA server can set up entries in an LDAP directory to keep track of who is issued a certificate, and you can query this LDAP server to find who has them. That is the good news, and indeed the PGP product makes use of its own LDAP server to keep track of its certificates. However, the LDAP server is only used by PGP; if you want a general-purpose LDAP server to keep track of your users, you'll have to install something else.

As a challenge for open systems and interoperability, we installed the Xcert Sentry CA and Netscape's Directory Server on a test network. The Xcert was used to create and manage our certificates for our test corporation, and the entries were placed in the Netscape LDAP directory. We created the certificates using the Netscape browser and stored the information in our Messenger e-mail software. After going through the process described previously, we had a valid certificate and could see it in the Security|Messenger settings. Although the Sentry CA couldn't automatically deposit a certificate in the Netscape LDAP server, we (operating as the security administrator) could do so with a few simple Web forms and keystrokes. So far, so good.

The challenge was trying to pry these certificates loose using other products, such as Outlook Express. There we ran into trouble, mainly because the Netscape software creates the certificate in a nonstandard place in the LDAP directory. According to the standards documents, the certificate should be placed in a particular spot in the LDAP directory schema, called *usercertificate*. Netscape, for whatever reason, places them at a location called *usersmimecertificate*. This meant that non-Netscape products couldn't view the certificates in our directory, because they were looking in the wrong place.

This brings up a very good point: The connection between a user and his or her certificate is tenuous at best. Just because you know that `david@strom.com` is the e-mail address of David Strom and you have his certificate, it doesn't mean that any of your expensive software tools can make this connection. This situation will create all sorts of headaches for your security administrators, and it means that you need to maintain at least two directories on your own machine—one for users and one for certificates.

It would be nice if the address books of our e-mail software could handle this automatically, but they don't.

That's not the only issue with managing certificates. What if someone leaves the company? Or changes his or her e-mail address? Or if you want to use the same certificate, but on several different machines? Most certificates are tied to a particular machine and a particular e-mail address, meaning that any new address will require a new certificate. Again, we find this situation unacceptable.

Encrypt and Decrypt Messages

Now you can finally go and encrypt your messages. Various options are available in your e-mail software to do this, and you can choose to sign a message as well as to encrypt it.

That is the encryption portion. What about the decryption side? If you have done your homework and exchanged certificates as we discussed earlier, then when you receive your encrypted message, it should automatically decrypt and display in plain text. You shouldn't have to do anything else—unless the encryption system is broken by a gateway or product incompatibility.

Futures

The obvious question is whether the Internet needs two standards for secure e-mail.

Proponents for both sides can make superficially compelling arguments. PGP proponents point to a grassroots constituency and a huge installed base of legacy systems. PGP emphasizes privacy for individuals. S/MIME proponents, on the other hand, point to some major vendors and an emphasis on nonrepudiation.

If history is any judge, the PGP side will win because less infrastructure is required to make it work. S/MIME has to solve all the problems that PGP has to solve, plus a few more. However, these things aren't decided overnight. So, our prediction is rather straightforward: The two sides will compete in the Internet marketplace for a couple of years, but ultimately the game is PGP's to lose. It requires less infrastructure and fewer broad agreements to achieve ubiquity.

Endnotes

[0] Our thanks to Dan Backman of *Network Computing* magazine for his help in sharing his lab and providing many valuable insights in the preparation of this article. This article is based, in part, on *Internet Messaging: From the Desktop to the Enterprise,* ISBN 0-13-9786100-4 Prentice-Hall, 1998.

[1] See **`http://strom.com/places/smime.html`** for details regarding product interoperability testing for encrypted e-mail packages.

[2] There is an alternative to this process. The United Parcel Service has produced a file transfer utility called NetDox, available at **`www.netdox.com`**. It requires special software to be installed on each computer, and it simplifies the certificate and encryption process somewhat. But this is yet another proprietary solution to the encrypted e-mail problem—something we think goes in the wrong direction.

[3] The article has more in-depth examination of testing MIME interoperability and features of Messenger, Outlook Express, Baltimore's MailSecure, OpenSoft's ExpressMail, and two Worldtalk plug-ins for Eudora and Outlook Express. See "Secure E-Mail Clients: Not Quite Ready for S/MIME Prime Time. Stay Tuned." *Network Computing,* February 1, 1998, **`techweb.cmp.com/nc/902/902r2.html`**.

MARSHALL T. ROSE is Chief Technology Officer of MessageMedia Inc. (formerly First Virtual Holdings, Inc.). He is responsible for the design, specification, and implementation of several Internet-standard technologies and is an author of over 60 of the Internet's RFCs, and several books on Internet technologies. He can be reached at **`mrose@dbc.mtview.ca.us`**

DAVID STROM is an independent consultant and frequent speaker at NetWorld+Interop shows around the world, where he teaches a class on e-commerce and Web storefronts. He was founding editor-in-chief of *Network Computing* magazine and has written over a thousand articles for various computer trade publications. He is also publisher of the e-mail newsletter *Web Informant*, an almost-weekly series of essays on Web marketing, technology, and culture. He can be reached at **`david@strom.com`**

Book Review

IP Multicasting

IP Multicasting: The Complete Guide to Interactive Corporate Networks, by Dave Kosiur, ISBN 0-471-24359-0 Wiley Computer Publishing, 1998, `http://www.wiley.com/compbooks/kosiur`

There is nothing remarkable about the statement: As technology becomes more affordable, applications once limited to power users find their way to the mainstream desktop. Video streaming, audio streaming, collaborative applications, and videoconferencing are all examples of applications once found exclusively on high-end workstations but now making their way to the mainstream desktop. If widespread deployment of these applications is to occur, we must be prepared to supply a supporting infrastructure.

The use of IP multicasting is gaining popularity, but many of the fundamentals that drive this and other network technologies, such as routing protocols and transport protocols, are still being debated. This book supplies a comprehensive view of the state-of-the-art as well as practical procedures one can follow in order to incorporate mulitcasting into existing network topologies.

Organization

Chapter 2 presents an introduction to TCP/IP basics and routing. Chapter 3, The Basics of Multicasting, addresses three sender-based multicasting protocols (ST-II, XTP, and MTP) and concentrates on IP multicast (a receiver-based multicasting protocol). The book would be much easier to follow if this chapter had been combined with Chapter 6.

Chapter 4, Multicast Routing Concepts, Chapter 5, Multicast Routing Protocols, and Chapter 6, Transport Protocols, constitute the heart of this book.

Beginning with basic concepts of unicast routing and routing algorithms, the author extends the models to deal with the problems of routing multicast data. Tree maintenance techniques form the bulk of Chapter 4.

Chapter 5 covers four multicast routing protocols: *Distance Vector Multicast Routing Protocol* (DVMRP); *Multicast Open Shortest Path First* (MOSPF); *Protocol Independent Multicast* (PIM); and *Core-Based Trees* (CBT). Placing the emphasis on PIM, Kosiur covers both PIM-SM (*sparse mode*) and PIM-DM (*dense mode*). He does a nice job of describing each of the protocols and summarizes each by reviewing its advantages and disadvantages. Finally, the author concludes by examining ways of achieving interdomain routing and protocol interoperability.

In Chapter 6, Kosiur provides an overview of the *Real-Time Transport Protocol* (RTP)/*Real-Time Transport Control Protocol* (RTCP) and the *Real-Time Streaming Protocol* (RTSP).

In addition, he discusses a dozen or more multicast protocols, all trying to answer the question: "How is retransmission of lost packets handled?" He classifies the protocol approaches into *receiver-based* or *sender-based*. In my opinion, this is the most interesting problem of multicasting. Answer this question wrong, and you find yourself with a nonscalable network cluttered with acknowledgments (ACKs).

Chapters 4 through 7 all consider delivering Quality of Service and so I was a little surprised to see Chapter 7 devoted to the subject.

Kosiur provides a good introduction to RSVP (*Resource ReserVation Protocol*), but until we see RSVP in wide deployment I would look at the previous three chapters for practical knowledge on the topic. In Chapter 7, and then in Chapter 11 he covers a lot of practical issues concerning Quality of Service, as well as ways to support multicasting over various networks, such as ATM, Frame Relay, and ISDN/dialup networks.

Chapter 9 is a compilation of some free and commercial software packages that use multicasting. Chapter 10 covers *Mbone* (the Multicast backbone), a popular experimental multicasting network. It is arguable that the state of multicasting wouldn't be where it is today without the Mbone.

A C+

This book rates a C+. Kosiur certainly has an understanding of the material, but his descriptions are neither clear nor concise. Reading this book is difficult, and learning from it even more so, but better organization could turn it into a gem.

—Neophytos Iacovou
University of Minnesota
Academic & Distributed Computing Services
iacovou@boombox.micro.umn.edu

Letter to the Editor

I just read the September 1998 issue of *The Internet Protocol Journal* and thoroughly enjoyed it. It was well written with excellent technical detail but more importantly, the contributors wrote in an understandable and organized method. This is not always the norm for good technical resources; so many times it is simply the reprint of a vendor's documentation.

"What is a VPN—Part II," written by Paul Ferguson and Geoff Huston, was a great article which described the various components and methodologies of VPNs. The information and explanation of the Virtual Private Dial Networking implementations, voluntary versus compulsory tunneling, subscriber's perspectives and real world applications clarified my understanding and knowledge on this subject. I also appreciate an article that ends with a conclusion. I have already located Part I of this article and will be reading it soon. There is one comment; it would be interesting to know which vendor when an example is used, regarding specifically the Frame Relay service provider.

The "Reliable Multicast Protocols and Applications" article was useful and informative, including the scaling issues and the information regarding the new reliable multicast protocols. The details of the *Pretty Good Multicast* (PGM) protocol and how it may improve scaling for multicast was very interesting.

The *Gigabit Ethernet* book review written by Ed Tittel was one of the most informative and well structured book reviews that I have read, especially in a smaller publication. Thanks for providing three pages for book reviews in a forty-seven page publication. This review provided all the information that would assist with the determination of purchasing the book or not.

I hope you continue to publish IPJ in hard copy. I do read and gather information from the Web like everyone else, but I prefer a physical copy to carry with me if I am traveling or at my home. Thanks again for a great publication and I can hardly wait for the next issue.

—*Joe Brannan*
joe.brannan@pepsi.com

Ed.: We appreciate your comments about our publication. Regarding your question about the Frame Relay example, it is our policy to avoid as much as possible any discussion of products, but we encourage readers to contact the authors directly for that kind of information.

We certainly plan to continue the print edition of IPJ. We are also developing a companion Web site (at **www.cisco.com/ipj***) that will contain additional information such as glossaries, links to other documents, updates, corrections, and so on. Thanks for writing.*

—*Ole Jacobsen*
ole@cisco.com

Fragments

ICANN Update

Back in the summer of 1997, the Clinton Administration decided that it was time to privatize the remaining Internet functions that were being managed within the federal research establishment, mostly dealing with Internet names and addresses. These functions had been handled very successfully over many years by the *Internet Assigned Numbers Authority* (IANA) under the direction Dr. Jon Postel and his staff at the Information Sciences Institute of the University of Southern California under contract to DARPA. But it was clear from the rapid expansion of the Internet, the emergence of important players on the industry side, and rising controversy over issues such as Network Solutions' monopoly in issuing domain names for **`.com`**, that change was necessary.

After two major policy papers and months of argumentative debate, the government recognized the *Internet Corporation for Assigned Names and Numbers* (ICANN) as the new body to assume responsibility for these largely technical management functions. Working from plans drawn up by Dr. Postel, his advisors and the Jones Day law firm, ICANN is endeavoring to satisfy the many constituencies that seek a voice in future decisions on Internet naming and addressing. Sadly, Jon died last fall just as his plan was approaching endorsement by the federal government.

The young organization, incorporated at the end of September, 1998, began operation in early November, has an initial Board of nine appointed Directors headed by Chairman Esther Dyson, and an interim President/CEO Mike Roberts. They are responsible for completing organizational details, devising a representation structure for electing their successors, and beginning to deal with a backlog of undone policy work, such as a determination on if, how and when new top level domains (TLDs) will be created. The new Board has Directors from six countries and plans to hold meetings quarterly in locations throughout the world, beginning with Singapore in March, 1999 and Berlin in May, 1999.

Being neither a Congressionally chartered corporation nor an industry trade association, but something in between, ICANN is an international organization that faces a tough political future with many skeptics challenging the notion that the Internet community can successfully govern itself in the important naming and addressing area. But with a startup fund from corporate contributions, Chairman Dyson and President Roberts, both short timers by design, are determined to get ICANN off the ground and into operation in coming months. More information is available at: **`www.icann.org`**

This publication is distributed on an "as-is" basis, without warranty of any kind either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. This publication could contain technical inaccuracies or typographical errors. Later issues may modify or update information provided in this issue. Neither the publisher nor any contributor shall have any liability to any person for any loss or damage caused directly or indirectly by the information contained herein.

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

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

CISCO SYSTEMS

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-J4
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED