BETTER, FASTER,

MORE SECURE

BRIAN CARPENTER, IBM and INTERNET ENGINEERING TASK FORCE

Who's in charge of the Internet's future?

Since I started a stint as chair of the IETF (Internet Engineering Task Force) in March 2005, I have frequently been asked, "What's coming next?" but I have usually declined to answer. Nobody is in charge of the Internet, which is a good thing, but it makes predictions difficult (and explains why this article starts with a disclaimer: It represents my views alone and not those of my colleagues at either IBM or the IETF).

The reason the lack of central control is a good thing is that it has allowed the Internet to be a laboratory for innovation throughout its life—and it's a rare thing for a major operational system to serve as its own development lab. As the old metaphor goes, we frequently change some of the Internet's engines in flight.

This is possible because of a few of the Internet's basic goals:

- Universal connectivity—anyone can send packets to anyone.
- Applications run at the edge—so anyone can install and offer services.
- "Cheap and cheerful" core technology—so transmission is cheap.
- Natural selection—no grand plan, but good technology survives and the rest dies.

Of course, this is an idealistic view. In recent years, firewalls and network address translators have made universal connectivity sticky. Some telecommunications operators would like to embed services in the network. Some transmission technologies try too hard, so they are not cheap. Until now, however, the Internet has remained a highly competitive environment and natural selection has prevailed, even though there have been attempts to protect incumbents by misguided regulation.

In this environment of natural selection, predicting technology trends is very hard. The scope is broad—the IETF considers specifications for how

BETTER, FASTER, MORE SECURE

IP runs over emerging hardware media, maintenance and improvements to IP itself and to transport protocols including the ubiquitous TCP, routing protocols, basic application protocols, network management, and security. A host of other standards bodies operate in parallel with the IETF.

To demonstrate the difficulty of prediction, let's consider only those ideas that get close enough to reality to be published within the IETF; that's about 1,400 new drafts per year, of which around 300 end up being published as IETF requests for comments (RFCs). By an optimistic rough estimate, at most 100 of these specifications will be in use 10 years later (i.e., 7 percent of the initial proposals). Of course, many other ideas are floated in other forums such as ACM SIGCOMM. So, anyone who agrees to write about emerging protocols has at least a 93 percent probability of writing nonsense.

What would I have predicted 10 years ago? As a matter of fact, I can answer that question. In a talk in May 1996 I cautiously quoted Lord Kelvin, who stated in 1895 that "heavier-than-air flying machines are impossible," and I incautiously predicted that CSCW (computer-supported collaborative work), such as packet videoconferencing and shared whiteboard, would be the next killer application after the Web, in terms of bandwidth and realtime requirements. I'm still waiting.

A little earlier, speaking to an IBM user meeting in 1994 (before I joined IBM), I made the following specific predictions:

- Desktop client/server is the whole of computing. The transaction processing model is unhelpful.
- Cost per plug of LAN will *increase*.
- Internet and IPX will merge and dominate.
- Desktop multimedia is more than a gimmick, but only part of desktop computing.
- Wireless mobile PCs will become very important.
- Network management (including manageable equipment and cabling) is the major cost.

Well, transaction processing is more important in 2006 than it has ever been, and IPX has just about vanished. The rest, I flatter myself, was reasonably accurate.

All of this should make it plain that predicting the future of the Internet is a mug's game. This article focuses on observable challenges and trends today.

HALT, WHO GOES THERE?

The original Internet goal that anyone could send a packet to anyone at any time was the root of the extraordinary growth observed in the mid-1990s. To quote Tim Berners-Lee, "There's a freedom about the Internet: As long as we accept the rules of sending packets around, we can send packets containing anything to anywhere."¹ As with all freedoms, however, there is a price. It's trivial to forge the origin of a data packet or of an e-mail message, so the vast majority of traffic on the Internet is unauthenticated, and the notion of identity on the Internet is fluid.

Anonymity is easy. When the Internet user community was small, it exerted enough social pressure on miscreants that this was not a major problem area. Over the past 10 years, however, spam, fraud, and denial-of-service attacks have become significant social and economic problems. Thus far, service providers and enterprise users have responded largely in a defensive style: firewalls to attempt to isolate themselves, filtering to eliminate unwanted or malicious traffic, and virtual private networks to cross the Internet safely.

These mechanisms are not likely going away, but what seems to be needed is a much more positive approach to security: Identify and authenticate the person or system you are communicating with, authorize certain actions accordingly, and if needed, account for usage. The term of art is AAA (authentication, authorization, accounting).

AAA is needed in many contexts and may be needed at several levels for the same user session. For example, a user may first need to authenticate to the local network provider. A good example is a hotel guest using the hotel's wireless network. The first attempt to access the Internet may require the user to enter a code supplied by the front desk. In an airport, a traveler may have to supply a credit card number to access the Internet or use a preexisting account with one of the network service providers that offer connectivity. A domestic ADSL customer normally authenticates to a service provider, too. IETF protocols such as EAP (Extensible Authentication Protocol) and RADIUS (Remote Authentication Dial-in User Service) are used to mediate these AAA interactions. This form of AAA, however, authenticates the user only as a sender and receiver of IP packets, and it isn't used at all where free service is provided (e.g., in a coffee shop).

Often (e.g., for a credit card transaction) the remote server needs a true identity, which must be authenti-

cated by some secret token (in a simple solution, a PIN code transmitted over a secure channel). But the merchant who takes the money may not need to know that true identity, as long as a trusted financial intermediary verifies it. Thus, authentication is not automatically the enemy of privacy.

Cryptographic authentication is a powerful tool. Just as it can be used to verify financial transactions, it can in theory be used to verify any message on the Internet. Why, then, do we still have spoofed e-mail and even spoofed individual data packets?

For individual data packets, there is a solution known as IPsec (IP security), which is defined in a series of IETF specifications and widely (but not universally) implemented. It follows the basic Internet architecture known as the end-to-end principle: Do not implement a function inside the network that can be better implemented in the two end systems of a communication. For two systems to authenticate (or encrypt) the packets they send to each other, they have only to use IPsec and to agree on the secret cryptographic keys. So why is this not in universal usage? There are at least three reasons:

- Cryptographic calculations take time during the sending and receiving of every packet. This overhead is not always acceptable except for very sensitive applications.
- Management of cryptographic keys has proved to be a hard problem, and usually requires some sort of preexisting trust relationship between the parties.
- Traversing firewalls and network address translators adds complexity and overhead to IPsec.

Thus, IPsec deployment today is limited mainly to virtual private network deployments where the overhead is considered acceptable, the two ends are part of the same company so key management is feasible, and firewall traversal is considered part of the overhead. More general usage of IPsec may occur as concern about malware within enterprise networks rises and as the deployment of IPv6 reduces the difficulties caused by network address translation.

For e-mail messages, mechanisms for authentication or encryption of whole messages have existed for years (known as S/MIME and PGP). Most people don't use them. Again, the need for a preexisting trust relationship appears to be the problem. Despite the annoyance of spam, people *want* to be able to receive mail from anybody without prior arrangement. Operators of Internet services *want* to receive unsolicited traffic from unknown parties; that's how they get new customers. A closed network may be good for some purposes, but it's not the Internet.

It's worth understanding that whereas normal end users can at worst send malicious traffic (such as denial-of-service attacks, viruses, and fraudulent mail), an ISP can in theory spy on or reroute traffic, or make one server simulate another. A hardware or software maker can in theory insert "back doors" in a product that would defeat almost any security or privacy mechanism. Thus, we need trustworthy service providers and manufacturers, and we must be very cautious about downloaded software.

To summarize the challenges in this area:

- How can identity be defined, authenticated, and kept private?
- How can trust relationships be created between arbitrary sets of parties?
- How can cryptographic keys be agreed upon between the parties in a trust relationship?
- How can packet origins be protected against spoofing at line speed?
- How can we continue to receive messages from unknown parties without continuing to receive unwanted messages?

The IETF is particularly interested in the last three questions. Work on key exchange has resulted in the IKEv2 standard (Internet key exchange, version 2), and work continues on profiles for use of public-key cryptography with IPsec and IKEv2. At the moment, the only practical defense against packet spoofing is to encourage ingress filtering by ISPs; simply put, that means that an ISP should discard packets from a customer's line unless they come from an IP address assigned to that customer. This eliminates spoofing only if every ISP in the world plays the game, however. Finally, the problem of spam prevention remains extremely hard; there is certainly no silver bullet that will solve this problem. At the moment the IETF's contribution is to develop the DKIM (Domain Keys Identified Mail) specification. If successful, this effort will allow a mail-sending domain to take responsibility, using digital signatures, for having taken part in the transmission of an e-mail message and to publish "policy" information about how it applies those signatures. Taken together, these measures will assist receiving domains in detecting (or ruling out) certain forms of spoofing as they pertain to the signing domain.

We are far from done on Internet security. We can expect new threats to emerge constantly, and old threats will mutate as defenses are found.

WHERE DID MY PACKET GO?

The Internet has never promised to deliver packets; technically, it is an "unreliable" datagram network, which

BETTER, FASTER, MORE SECURE

may and does lose a (hopefully small) fraction of all packets. By the end-to-end principle, end systems are required to detect and compensate for missing packets. For reliable data transmission, that means retransmission, normally performed by the TCP half of TCP/IP. Users will see such retransmission, if they notice it at all, as a performance glitch. For media streams such as VoIP, packet loss will often be compensated for by a codec—but a burst of packet loss will result in broken speech or patchy video. For this reason, the issue of QoS (quality of service) came to the fore some years ago, when audio and video codecs first became practical. It remains a challenge.

One aspect of QoS is purely operational. The more competently a network is designed and managed, the better the service will be, with more consistent performance and fewer outages. Although unglamorous, this is probably the most effective way of providing good QoS.

Beyond that, there are three more approaches to QoS, which can be summarized as:

- Throw bandwidth at the problem.
- Reserve bandwidth.
- Operate multiple service classes.

The first approach is based on the observation that both in the core of ISP networks and in properly cabled business environments, raw bandwidth is cheap (even without considering the now-historical fiber glut). In fact, the only place where bandwidth is significantly limited is in the access networks (local loops and wireless networks). Thus, most ISPs and businesses have solved the bulk of their QoS problem by overprovisioning their core bandwidth. This limits the QoS problem to access networks and any other specific bottlenecks.

The question then is how to provide QoS management at those bottlenecks, which is where bandwidth reservations or service classes come into play. In the reservation approach, a session asks the network to assign bandwidth all along its path. In this context, a session could be a single VoIP call, or it could be a semi-permanent path between two networks. This approach has been explored in the IETF for more than 10 years under the name of “Integrated Services,” supported by RSVP (Resource Res-

ervation Protocol). Even with the rapid growth of VoIP recently, RSVP has not struck oil—deployment seems too clumsy. A related approach, however—building virtual paths with guaranteed bandwidth across the network core—is embodied in the use of MPLS (MultiProtocol Label Switching). In fact, a derivative of RSVP known as RSVP-TE (for traffic engineering) can be used to build MPLS paths with specified bandwidth. Many ISPs are using MPLS technology.

MPLS does not solve the QoS problem in the access networks, which by their very nature are composed of a rapidly evolving variety of technologies (ADSL, CATV, various forms of Wi-Fi, etc.). Only one technology is common to all these networks: IP itself. Therefore, the final piece of the QoS puzzle works at the IP level. Known as Differentiated Services, it is a simple way of marking every packet for an appropriate service class, so that VoIP traffic can be handled with less jitter than Web browsing, for example. Obviously, this is desirable from a user viewpoint, and it’s ironic that the more extreme legislative proposals for so-called “net neutrality” would ostensibly outlaw it, as well as outlawing priority handling for VoIP calls to 911.

The challenge for service providers is how to knit the four QoS tools (competent operation, overprovision of bandwidth, traffic engineering, and differentiated services) into a smooth service offering for users. This challenge is bound up with the need for integrated network management systems, where not only the IETF but also the DMTF (Distributed Management Task Force), TMF (TeleManagement Forum), ITU (International Telecommunication Union), and other organizations are active. This is an area where we have plenty of standards, and the practical challenge is integrating them.

However, the Internet’s 25-year-old service model, which allows any packet to be lost without warning, remains; and transport and application protocols still have to be designed accordingly.

BACK TO THE FUTURE

As previously mentioned, MPLS allows operators to create virtual paths, typically used to manage traffic flows across an ISP backbone or between separate sites in a large corporate network. At first glance, this revives an old controversy in network engineering—the conflict between datagrams and virtual circuits. More than three decades ago this was a major issue. At that time, conventional solutions depended on end-to-end electrical circuits (hardwired or switched, and multiplexed where convenient).

The notion of packet switching, or datagrams, was introduced in 1962 by Paul Baran and became practicable from about 1969 when the ARPANET started up. To the telecommunications industry, it seemed natural to combine the two concepts (i.e., send the packets along a predefined path, which became known as a virtual circuit). The primary result was the standard known as X.25, developed by the ITU.

To the emerging computer networking community, this seemed to add pointless complexity and overhead. In effect, this controversy was resolved by the market, with the predominance of TCP/IP and the decline of X.25 from about 1990 onward. Why then has the virtual circuit approach reappeared in the form of MPLS?

First, you should understand that MPLS was developed by the IETF, the custodian of the TCP/IP standards, and it is accurate to say that the primary target for MPLS is the transport of IP packets. Data still enters, crosses, and leaves the Internet encased in IP packets. Within certain domains, however, typically formed by single ISPs, the packets will be routed through a preexisting MPLS path. This has two benefits:

- At intermediate switches along the path, switching will take place at full hardware speed (today electronically, tomorrow optically), without any complex routing decisions being made at line speed.
- The path itself is established with whatever security, bandwidth, and QoS characteristics it is considered to need, using network management techniques known collectively as “traffic engineering.”

Note that not all experts are convinced by these benefits. Modern IP routers are hardly slow, and as previously noted, QoS may in practice not be a problem in the network core. Most ISPs insist on the need for such traffic engineering, however.

Even with MPLS virtual paths, the fundamental unit of transmission on the Internet remains a single IP packet.

THE EVERLASTING PROBLEM

In 1992, the IETF’s steering group published a request for comments² that focused on two severe problems facing the Internet at that time (just as its escape from the research community into the general economy was starting): first, IP address space exhaustion; and second, routing table explosion. The first has been contained by a hack (network address translation) and is being solved by the growing deployment of IPv6 with vastly increased address space. The second problem is still with us. The number of entries in the backbone routing tables of the Internet was below 20,000 in 1992 and is above 250,000 today (see figure 1).

This is a tough problem. Despite the optimistic comment about router speed, such a large routing table needs to be updated dynamically on a worldwide basis. Furthermore, it currently contains not only one entry for each address block assigned to any ISP anywhere in the world, but also an entry for every user site that needs to be “multihomed” (connected simultaneously to more than one ISP for backup or load-sharing). As more businesses come to rely on the network, the number of multihomed sites is expected to grow dramatically, with a serious estimate of 10 million by 2050. A routing table of this size is not considered feasible. Even if Moore’s law solves the storage and processing challenge at reasonable cost, the rate of

BETTER, FASTER, MORE SECURE

change in a table of that size could greatly exceed the rate at which routing updates could be distributed worldwide.

Although we have known about this problem for more than 10 years, we are still waiting for the breakthrough ideas that will solve it.

MULTIPLE UNIVERSES?

The telecommunications industry was fundamentally surprised by the Internet's success in the 1990s and then fundamentally shaken by its economic consequences. Only now is the industry delivering a coherent response, in the form of the ITU's NGN (Next Generation Networks) initiative launched in 2004. NGN is to a large extent founded on IETF standards, including IP, MPLS, and SIP (Session Initiation Protocol), which is the foundation of standardized VoIP and IMS (IP Multimedia Subsystem). IMS was developed for third-generation cellphones but is now the basis for what ITU calls "fixed-mobile convergence." The basic principles of NGN are:³

- IP packet-based transport using MPLS
- QoS-enabled
- Embedded service-related functions—layered on top of transport or based on IMS
- User access to competing service providers
- Generalized mobility

At this writing, the standardization of NGN around these principles is well advanced. Although it is new for the telecommunications industry to layer services on top rather than embedding them in the transport network, there is still a big contrast with the Internet here: Internet services are by definition placed at the edges and are not normally provided by ISPs as such. The Internet has a history of avoiding monopoly deployments; it grows by spontaneous combustion, which allows natural selection of winning applications by the end users. Embedding service functions in the network has never worked in the past (except for directories). Why will it work now?

COME JOIN THE DANCE

It should be clear from this superficial and partial personal survey that we are still having fun developing the

technology of the Internet, and that the party is far from over. The Internet technical community has succeeded by being open—and open-minded. Any engineer who wants to join in can do so. The IETF has no membership requirements; anyone can join the mailing list of any working group, and anyone who pays the meeting fee can attend IETF meetings. Decisions are made by rough consensus, not by voting. The leadership committees in the IETF are drawn from the active participants by a community nomination process. Apart from meeting fees, the IETF is supported by the Internet Society.

Any engineer who wants to join in can do so in several ways: by supporting the Internet Society (<http://www.isoc.org>), by joining IETF activities of interest (<http://www.ietf.org>), or by contributing to research activities (<http://www.irtf.org> and, of course, ACM SIGCOMM at <http://www.acm.org/sigs/sigcomm/>). ☐

REFERENCES

1. Berners-Lee, T. 1999. *Weaving the Web*. San Francisco: HarperCollins.
2. Gross, P., Almquist, P. 1992. IESG deliberations on routing and addressing, RFC 1380 (November). DDN Network Information Center; <http://www.rfc-archive.org/getrfc.php?rfc=1380>.
3. Based on a talk by Keith Knightson. 2005. Basic NGN architecture principles and issues; <http://www.itu.int/ITU-T/worksem/ngn/200505/program.html>.

ACKNOWLEDGMENTS

Thanks to Bernard Aboba and Stu Feldman for valuable comments on a draft of this article.

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

BRIAN E. CARPENTER is an IBM Distinguished Engineer working on Internet standards and technology. Based in Switzerland, he became chair of the IETF (Internet Engineering Task Force) in March 2005. Before joining IBM, he led the networking group at CERN, the European Laboratory for Particle Physics, from 1985 to 1996. He served from March 1994 to March 2002 on the Internet Architecture Board, which he chaired for five years. He also served as a trustee of the Internet Society and was chairman of its board of trustees for two years until June 2002. He holds a first degree in physics and a Ph.D. in computer science, and is a chartered engineer (UK) and a member of the IBM Academy of Technology.

© 2006 ACM 1542-7730/06/1200 \$5.00