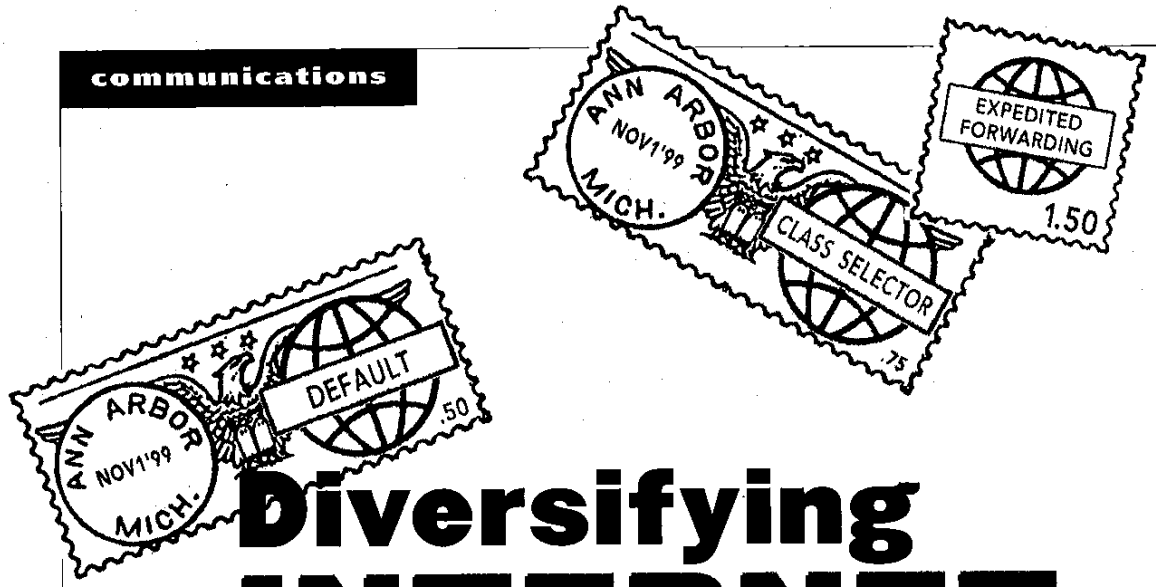
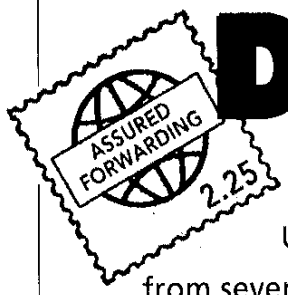
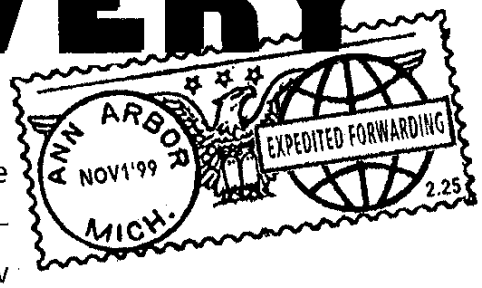


Diversifying INTERNET DELIVERY



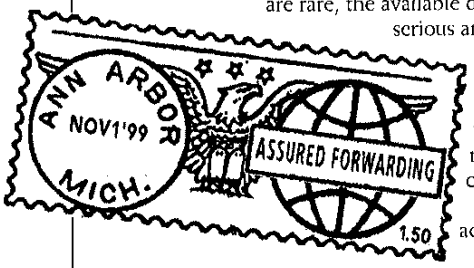


Users will be able to choose from several levels of performance—not one-size-fits-all — when a new service scheme replaces the Internet's existing infrastructure

Brian E. Carpenter &
Dilip D. Kandlur
IBM Corp.

Most people have encountered the Internet through the World Wide Web and e-mail. In fact, the popularity of these applications is what led to exuberant growth in the net's use. That expansion now poses a direct challenge: those responsible for the net's well-being must figure out how to make the technology scale up.

With billions of bits struggling to make it through the trunk lines of the Internet's backbone, performance has become a major issue. Although objective measurements are rare, the available data gives cause for concern: indications are that packet loss is serious and network congestion is at the root of the loss.



Over time, it has become harder to deploy end-to-end solutions across the entire network. The emergence of the company intranet, attended as it is by gateways, proxies, firewalls, address translators, and other forms of obstruction, has marred the network's transparency, that is, the ability to see the status and configuration of systems from source to destination.

Further, there is a clear demand for a variety of types of service across the Internet. New applications—for telephony, video

0018-9235/99/$10.00©1999 IEEE

broadcasting, remote performance of medical operations, letting people telecommute, and more—will ask even greater predictability of Internet performance. Companies and individual customers will want service providers always to guarantee a known level, or quality, of service. Service providers can thus compete on the basis of the types of services they can deliver reliably and at an attractive price.

To stabilize and improve Internet performance enough for the network to carry mission-critical services, it is necessary to control packet loss and hence congestion. A solution that can achieve this control on a massive scale and in the absence of network layer transparency will have to be very simple. In particular, it must not require routers in the network to stop and store information about the state of each and every passing communication session. With these goals in view, the Internet Engineering Task Force (IETF), the voluntary body of engineers who develop standards for the Internet, is now developing a differentiated services model.

What follows describes this new services model and the associated work on service policy mechanisms. It also explains the impact of these services and mechanisms on network servers and routers, shows how they tackle the problems of congestion and loss, and discusses how they may be used to provide a variety of grades of service.

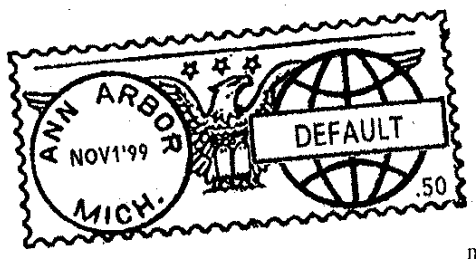
As things stand

It is a commonplace that the Internet is growing by leaps and bounds. Estimates of the annual rate of increase in demand run from 300 to 700 percent. Whatever the true figure, clearly Internet technology has to anticipate a network with several billion nodes, many connected at speeds a good deal higher than today's telephone modem connections can handle.

Even on a conservative basis, backbone trunks in the carrier networks may be expected to have capacities in the tens of gigabits per second, so that they can cope with traffic for millions of simultaneous communications. While the numbers may be less daunting in subscriber networks or inside corporate or campus networks, network upscaling issues remain a first priority for Internet technology.

Growth in demand for Internet access by another two or three orders of magnitude is feasible. After all, over some 25 years the Internet has expanded from the 100 or so hosts, or nodes, of the Arpanet (the early U.S. government network that was its progenitor) to around 60 million nodes today. However, the next stage of scaling will have to take the network past four significant milestones: addressing, routing, security, and quality.

The addressing milestone is based on the fact that, while the original 32-bit Internet Protocol (IP) address space is not yet exhausted, it can only be used rather sparingly. For several years, the registries that dole out blocks of address space have applied strict rules about how and to whom addresses are distributed to ward off satiation. Granted, a new 128-bit address space is available in the next version of IP (IP version 6, or IPv6), but it has yet to be widely deployed.



The second milestone has to do with the Internet's routing tables. As the Internet grows, so do the tables of addresses used by routers to move traffic. The tables map the logical topology of the address space into the physical topology of the links between routers. But there is no mathematical relationship between these two topologies, so if nothing is done, the mapping tables tend to grow without bound. Even if the tables do fit into the available memory, they must be recalculated whenever the logical or physical topology changes, and the time spent on that job also tends to grow without bound, most especially if multicast routing is involved.

As to the security milestone, it is no longer acceptable to deploy insecure technology on the Internet. If that were not so, correspondence would become vulnerable to forgery, and unsanctioned wiretapping or theft and denial of service would be all too easy.

Finally, there is the quality milestone. As the Internet penetrates deeper and deeper into the economy, it needs to be ruggedly reliable and its performance highly predictable. It is no longer acceptable for the Internet to offer poor quality.

Differentiated services—or DiffServ, as it has become known in the Internet community—is an aspect of the quality milestone. But like any new technology developed for the Internet today, it affects, and is affected by, the other milestones. In other words, it must be independent of the address space and the IP version in use, it must have no negative impact on routing overhead, and it must work securely.

Fast and slow service needed

In almost all cases, today's Internet, as well as private networks based on Internet technology, such as corporate intranets, treats data packets exactly alike. As a packet hops through the network, it receives the same treatment whether it is part of an overnight file transfer, a request to or a reply from a Web server, an audio segment from an IP telephone call, or part of a live videocast. Furthermore, on a shared public network a packet is treated the same regardless of how much the user concerned is paying for Internet service.

If Internet applications were homogeneous, this might not be of great consequence. For example, what if all network traffic were telephony at a fixed bit-rate? Then the network designer would know that each call required (say) 64 kb/s of capacity, and that a given path on the network will be designed to handle a certain maximum number of calls. Multiplying the number of calls, N, by the 64-kb/s call requirement would determine the digital bandwidth of the path.

Each call could then simply send packets adding up to 64 kb/s in real time. (The nontransmission of silent packets is of course an available optimization, and telephony can adapt to lower capacity by reducing voice quality. But this does not change the essential argument, since there must be a maximum data rate per call on the average.)

With such a homogeneous system, it would be easy to compute the necessary capacity. Furthermore, since the packets will be transmitted immediately, in real time, the known electrical length of the path makes it easy to compute the maximum delay any voice packet could suffer. If the computation were imprecise, a few voice packets might be lost; but that is acceptable, because voice signals contain a great deal of extraneous information.

The situation would be different if all traffic were overnight file transfers. The total number of transfers, the speeds at which disks at each end-point could accept data, and the size of the files to be transferred would be unknown. While no individual transfer would be especially urgent (it is, after all, overnight transmission), any lost packets would have to be retransmitted to avoid file corruption. In this case the file transfers could run on top of the transmission control protocol (TCP), which has two important characteristics. First, any packets lost to congestion are retransmitted in due course. Second, sensing the packet loss caused by congestion, TCP slows down the rate at which each packet in a given file transfer is sent. Then no transfer takes more than its fair share of the total capacity, relieving congestion.

Now consider what happens when these two types of applications (telephony and file transfer) are mixed together in the same part of the network. Regardless of traffic congestion and packet loss, the telephony system design will preserve the transmission rate at 64 kb/s for each of the N calls. Meanwhile, the file transfer traffic using the same

path and also facing congestion and packet loss, will slow down in an attempt to share the available capacity fairly. As the telephony traffic cannot slow down, it will obtain an unfair share of the capacity; yet it will still suffer from high loss, and thus poor voice quality, because file transfer traffic is still being pushed into the path, albeit at a slower rate.

In general, if traffic slowed down by congestion is mixed in with real-time traffic that keeps going despite congestion, both sides lose. This problem has been studied by the Internet Research Task Force for some time, but remains intractable for currently deployed networks, which treat all types of traffic in the same way. To solve it, the next-generation Internet must be able to treat different types of traffic differently as they flow through its paths.

Another requirement, reported by Internet service providers, is the need to treat traffic from or to different subscribers differently. For instance, Company A may be willing to pay more to ensure that its overnight file transfers receive the highest possible priority. Or Company X may be prepared to pay more for Web hosting than Company Y if its Web pages are served up faster as a result. Or an Internet service provider may want to offer two grades of service to private domestic consumers, with two price levels and two distinct response times to Web requests.

In a sense, none of the requirements mentioned is new, and some of them may be met in part today. By reserving bandwidth for individual subscribers, perhaps in the form of a virtual private network, an Internet service provider could allow all of Company A's intra-company traffic to be isolated from other traffic. However, this arrangement does nothing to separate Company A's voice traffic from its file transfer traffic, nor does it improve Company A's response time to Web hits from the public Internet.

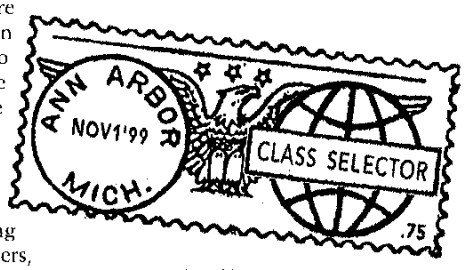
A method of differentiating traffic at any and every point in the Internet seems to be called for. Needless to say, it must scale up to multiple billions of nodes as the Internet grows, be deployable globally, and be manageable by network operators.

Earlier work

Since the problem is as old as the concept of packet switching, it has already been worked on. The nature of a viable solution is hinted at by four earlier approaches: IP precedence and Type of Service (TOS), IBM's System Network Architecture class of service, the Internet's Stream Protocol, and Integrated Services.

The IP precedence and TOS approach was part of the original IP design. In essence, a byte known as the type-of-service octet was reserved in the header of every packet. The byte was defined to contain two important fields: a 3-bit precedence value, and three type-of-service bits. The precedence was intended as a simple priority marker, where priority 0 got the worst treatment and priority 7 got the best. High precedence has often been used to transmit updates of the routing tables mentioned earlier. The type-of-service bits were more ambitious, being identified as low delay, high throughput, and high reliability. Unfortunately, how these properties could be implemented across a network was never explained. Some later work expanded the definitions, but still gave no recipe for implementation. In practice, although some of these bits have been set in some software, they have been of little use and generally ignored.

System Network Architecture (SNA) has long distinguished among several classes of service by, for example, giving real-time interactive transactions priority over batch operations. Having the network's control points schedule traffic on the links has enforced these classes of service. However, unlike the Internet, SNA was designed for use in organizations where deployment could be mandated by a central authority.



The Internet Stream Protocol was an attempt to accommodate real-time traffic streams in parallel with TCP/IP traffic by adding a second, connection-oriented network protocol in parallel with IP. Known as ST, it has failed to grow outside a limited experimental community, largely because it must store state information in every router involved.

Integrated Services (IntServ) is a major effort within the Internet Engineering Task Force. Its aim is to specify a mechanism for supporting those end-to-end sessions across the Internet that require a specific quality of service (such as a given peak capacity and transmission delay). The IntServ model requires that a software or firmware module reside in every IP router along the path capable of reserving resources for each session and checking that each data packet receives the resources to which it is entitled. Resource reservations are requested using a special protocol known as the Resource Reservation Protocol (RSVP).

IntServ introduces an important notion called admission control. Admission control is a process whereby a network can refuse to admit traffic when resources are insufficient. If the RSVP request fails, the session will not start (or will do so in a degraded mode). IntServ also employs the notion of traffic shaping; at the input to the network, packets will be spaced out in time so as to correspond to the resources reserved by RSVP.

Offsetting these and other attractive attributes of IntServ are two disadvantages. IntServ requires new software or firmware to be added to all routers along the network path concerned. Also, if it were to be used on major Internet service provider trunk connections, which carry millions of packets per second, the overhead per packet of implementing the necessary checks and resource management is widely believed to be unacceptable. For these reasons, it is expected that IntServ and RSVP will initially be limited to campus and small corporate networks.

Constraints on design

These earlier attempts at differentiated service make clear the design constraints on a standard for the Internet. They are:

- Deployability in small steps, with a degree of backward compatibility, including interoperability with IP precedence and with the IntServ/RSVP model.
- Minimal overhead on backbone and trunk routers. Specifically, the standard must not require these routers to store information about an individual traffic flow or treat a particular flow in a special way. (Of course, it must allow different types of flows to be handled in different ways.)
- Separation of real-time traffic from the TCP-like traffic that reacts to congestion by slowing down and retransmitting undelivered packets.
- The ability for Internet service providers to offer different grades of service to different customers.
- Inclusion of management facilities, not least the ability of network operators to assign and monitor the use of resources.

The current and future Internet protocols already have a mechanism by which a scheme can be implemented. Recall that every IP packet carries a byte called the type-of-service octet. In all but a few percent of traffic, this byte is set to zero; clearly it is an under-utilized feature of IP. In the new 128-bit IPv6 there is an equivalent byte, called the traffic class octet, whose use has not yet been specified. Both bytes can therefore be called into service for a differentiated-services scheme.

The differentiated-services model uses the most-significant 6 bits (0–5) from the type-of-service or traffic class octet, defined identically for the old IPv4 and the new IPv6. Known as the Differentiated Services Code Point (DSCP), this 6-bit field indicates how each router should

treat the packet. To emphasize the fact that the router need not store information about what the ultimate provider and consumer of the data are doing (so-called session information), this treatment is known as a per-hop behavior (PHB). On the Internet, the transmission of a data packet between two routers is only one leg, or hop, in its journey, and a per-hop behavior defines how an individual router will treat an individual packet when sending it over the next hop through the network.

Being 6 bits long, the differentiated-services code point can have one of 64 different binary values and each one can be defined as calling for a unique per-hop behavior. Many experts believe that 64 different behaviors are more than will ever be needed but, to allow for all eventualities, some of the 64 possible values are reserved for local or experimental use.

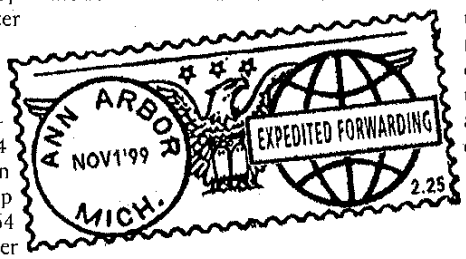
When a packet arrives, a router has a new job to do in addition to deciding which output port to send the packet to. In concept, the router will use the code point to select one of 64 possible subroutines that will manage the handling of the packet at its output port. What the subroutine actually does will depend on the definition of the per-hop behavior for the particular code point. For example, the subroutine might instruct the router to put the packet at the front of the queue at the output port, thereby giving it highest priority, or at the back of the queue, giving it the lowest. Or the packet might be placed in a queue where its transmission depends on special circumstances, such as link usage. In some cases, the subroutine will execute an algorithm that doles out the traffic capacity of the outgoing link among the different types of per-hop behaviors. What share of capacity a particular type of per-hop behavior gets would be defined by some network management mechanism.

It is a basic feature of DiffServ that every packet must be classified, that is, it must have a suitable value inserted in its differentiated-services code point field. The value can be inserted in either of two places: the original source of the traffic or at a router.

Inserting the code point at the original source of the traffic, such as a Web server or IP telephony gateway, has a distinct advantage. The server or gateway in question can have explicit knowledge of the application in use, and can therefore mark packets in an application-dependent way.

One alternative is to have the traffic classified and marked by a router—say, the first encountered by the traffic or the one at the boundary between customer and Internet service provider. The advantage here is that no change is needed to servers. But the router requires some extra "smarts." Fortunately, many routers have a very similar capability already, for use with IntServ/RSVP. DiffServ needs this extra logic only in the boundary routers, and thereby avoids the performance overhead suffered by IntServ on backbone trunks.

A third option is to combine DiffServ with IntServ/RSVP. In this situation, the intranet server would use the IntServ/RSVP model for communication with a boundary router, which stands between a private network and the Internet, and the boundary router would use the DiffServ model for wide-area communication across the open Internet.



All of these options have the same result: traffic that flows out onto the broad expanses of the Internet is marked with a specific differentiated-services code point value at or near its source. In the Internet service provider's backbones and trunk routers, no further processing is required except for the rapid choice of queuing mechanism mentioned above.

The kind of service the user sees will depend on the per-hop behavior selected and on the amount of network capacity that Internet service providers along the path have assigned to that per-hop behavior. It will also depend upon statistics: unless the behavior is a very special one, an element of congestion and packet loss will remain.

Four overall types of per-hop behaviors have been defined as standard so far. They are default, class-selector, expedited forwarding, and assured forwarding. For default behavior, the code point value is zero and the service to be expected is exactly what is provided by today's Internet service, with its completely uncontrolled traffic congestion and packet loss.

For class-selector behaviors, there are seven code point values, running from 001000 to 111000 and selecting up to seven ranked behaviors. Each behavior has at least as good a probability of timely forwarding as its predecessor in the ranking, if not a better one. Note that the default behavior plus the class selectors exactly mirror the original eight IP precedence values, thereby providing compatibility with that scheme.

Expedited forwarding (EF) behavior has a recommended differentiated-services code point value of 101110. The departure rate of EF traffic is defined as necessarily equaling or exceeding a configurable rate. EF is intended to allow the creation of real-time services with a configured throughput rate for the services' data packets.

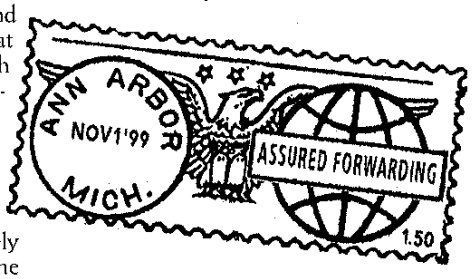
Assured forwarding (AF) behavior actually consists of three sub-behaviors, which for convenience may be called AF_1, AF_2, and AF_3. When the network is congested, packets marked for AF_1 have the lowest probability of being discarded by any router, and packets marked for AF_3 have the highest. Thus, within the AF class, differential drop probabilities are available; otherwise, the class represents a single type of per-hop behavior. The standard actually defines four independent AF classes. Quite complex service offerings can be constructed using AF behaviors, and much remains to be understood about them.

To complete the picture, note that, in addition to a classifier, the entry point to a DiffServ network will require some form of admission control and traffic shaping. Otherwise, too much traffic for a given class of behavior may enter the network, or momentary excesses of traffic may occur. It is expected that these components will be re-used from IntServ/RSVP implementations. As they will be needed only at the edges of the network, however, DiffServ is not expected to penalize end-to-end performance.

A matter of policy

A DiffServ network will have many parameters: what per-hop behaviors are implemented? How are the routers to share capacity among those behaviors? Which type of traffic, from which sources, is allowed to use which per-hop behaviors? What parameters control the classifiers and traffic shapers? How does the admission control mechanism know the available capacity?

These are complex questions, many of which are still under study. They all point to a common requirement: the need for a network management system to deliver DiffServ policy to the routers and servers involved. At one level of abstraction, a policy statement might be "allow server X to generate a maximum of 6 Mb/s of video traffic during prime time, but there is no restriction overnight." At another level, it might be "allow customer A 10 Mb/s of AF_1 traffic at all times, and outside prime time allow unrestricted AF_2 and AF_3 as well." At the lowest level of detail, policy statements



would more likely refer to IP addresses and protocol numbers, and to specific code point values and network capacity values.

The work on a standard policy framework is still at an early stage, and it would be misleading to go into real detail. It seems likely that DiffServ policy, along with policy for other network resources, will be stored in a repository using the lightweight directory-access protocol (LDAP), an open protocol that enables all types of computer to access the directory's data. The policy would be communicated to the servers and routers by one of several protocols. Of course, adequate operator interfaces with this policy management system will have to be provided, so that the system can be integrated with generic network management tools.

Yet to come

The DiffServ standards developed so far meet the constraints outlined previously. Specifically, DiffServ is backwardly compatible with IP precedence and can interoperate with IntServ/RSVP. It requires no per-flow state in backbone and trunk routers. TCP-like traffic may be classified into a different per-hop behavior from real-time traffic (assured and expedited forwarding, for instance). Internet service providers can assign different assured forwarding classes to different customers.

In the broader context of the Internet as a whole, DiffServ is a fundamental part of the Internet's approach to the quality milestone. Nore, because it relies on only on a single byte in the IP header, will DiffServ interfere with the addressing, routing, and security milestones.

Yet much remains to be done. The Internet Engineering Task Force is currently developing the mechanisms necessary for policy management and related low-level network management. More important is the need to gain experience of DiffServ technology by deploying it.

An early example of this is the Qbone, which was launched in October 1998. The goal of this initiative by the Internet2 coalition (a group of over 150 U.S. universities and research institutions) is to build a testbed for new IP quality-of-service (QoS) technologies using the differentiated services approach. If Qbone can meet the demands placed on the network by such new applications as remote instrument control and virtual classrooms, then universities in the United States will have the tools to teach and do research in the coming century. In the words of the Qbone group, the differentiated services approach "has great potential to overcome some of the complexities of earlier IP QoS architectures." They also recognize that DiffServ will require a great deal of implementation experience, engineering, and study before becoming mature enough to offer production-quality performance, and they are willing to put in the effort needed.

A related experiment within the European Union is the Quantum Test Program (QTP), where Quantum stands for quality network technology for user-oriented multimedia. The program aims to test and validate new technologies, products, and services with a view to introducing them into the TEN-155 service at some future date. Operational since 11 December 1998, TEN-155 connects 16 national research networks and one regional research network in Eastern Europe. It replaces the previous TEN-34 network.

Hopefully, these projects will be successful precursors to much wider deployment of differentiated services across the Internet. ◆

To probe further

The history of the Internet's technical development can be found in the requests-for-comments (RFC) library (http://www.ietf.org/rfc/) maintained on line by the Internet Engineering Task Force. The library's electronic documents record rationales, overviews, and specifications for implementing the various facilities and capabilities that compose the Internet.

The documents bulleted below can be accessed using http://www.ietf.org/rfc/rfcXXXX.txt, where XXXX stands for the RFC number. If a number contains fewer than four digits, leading zeros are used. The origin and working of the Internet Engineering Task Force itself can be found in "The Tao of IETF," RFC 1718, November 1994.

- "Internet Protocol," edited by J. Postel, RFC 791, September 1981, which gives the original approach to solving the problem of handling different types of packets.

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

- "Internet Stream Protocol version 2 (ST2) Protocol Specification," edited by L. Delgrossi and L. Berger, RFC 1819, August 1995.

- For limiting factors for IntServ/RSVP, "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement: Some Guidelines on Deployment," edited by Alison Mankin, RFC 2208, September 1997.

- "Recommendations on Queue Management and Congestion Avoidance in the Internet," by R. Braden and others, RFC 2309, April 1998, which tells of the IETF's work on the real-time versus congestion-slowed traffic problem.

Among the many documents relating specifically to describing and implementing the differentiated service scheme, some of basic ones are:

- "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," by K. Nichols, S. Blake, F. Baker, and D.Black, RFC 2474, December 1998.

- "An Architecture for Differentiated Services," by S. Blake and five others, RFC 2475, December 1998.

- "Assured Forwarding PHB Group," by J. Heinanen et al., RFC 2597, June 1999.

- "An Expedited Forwarding PHB," by V. Jacobson, K. Nichols, and K. Poduri, RFC 2598, also in June 1999.

- "A Two-bit Differentiated Services Architecture for the Internet," by K. Nichols, V. Jacobson, and L. Zang, RFC 2638, July 1999.

The latest specification for LDAP appears in "Lightweight Directory Access Protocol (v3)," by Y. Yaacovi, M. Wahl, and T. Genovese, RFC2589, May 1999.

Also worth visiting are the Qbone Project's home page, http://www.internet2.edu/qos/qbone and that of the Quantum test program, http://www.dante.net/tf-tant/.

About the authors

Brian E. Carpenter is program director, Internet standards and technology, in the Internet Division of IBM Corp., Hawthorne, N.Y. He is also chief architect of the IBM-sponsored International Center for Advanced Internet Research (iCAIR) at Northwestern University in Evanston,Ill., where he is currently based. In addition, Carpenter chairs the Internet Architecture Board and is an active member of the Internet Engineering Task Force, where he is a co-chair of the Differentiated Services working group. Previously he led the networking group at CERN, the European Laboratory for Particle Physics, in Geneva.

Dilip D. Kandlur is the manager of the open systems networking group at IBM's Thomas J. Watson Research Center, Yorktown Heights, N.Y., where he oversees efforts to advance Internet protocols. A native of India, he joined IBM in 1991 after receiving a doctorate in computer science from the University of Michigan-Ann Arbor. Both he and Carpenter are involved with the Qbone and Quantum projects.

Spectrum editor: Richard Comerford