Differentiated Services

With the tremendous increase in Internet traffic volume and the introduction of new real-time, multimedia, and multicasting applications, the traditional IP-based services are woefully inadequate. The differentiated services architecture supports a range of network services that are differentiated on the basis of performance.

By William Stallings

The first attempt at fixing the Internet's lack of support for multimedia and other applications requiring real-time data delivery was the introduction of the integrated services architecture (ISA) and the related reservation protocol (RSVP). ISA and RSVP support quality of service (QoS) offerings on the Internet and in private internets. Although ISA and RSVP are useful tools, their features are relatively complex to deploy. Furthermore, they may not be able to handle large volumes of traffic due to the amount of control signaling required to coordinate integrated QoS offerings and the maintenance of state information required at routers.

As the burden on the Internet grows, and as the variety of applications grows with it, providing differing levels of QoS to different traffic flows is imperative. The differentiated services (DS) architecture (detailed in the IETF's RFC 2475) is designed to provide a simple, easy-to-implement, low-overhead tool that can support a range of network services differentiated on the basis of performance.

Several key characteristics of DS contribute to its efficiency and ease of deployment:

- IP packets are labeled for differing QoS treatment using the existing IPv4 type of service (TOS) octet or IPv6 traffic class octet (see Figure 1). Thus, no change is required to IP.
- A service-level agreement (SLA) is established between the service provider (Internet domain) and the customer prior to the use of DS. This avoids the need to incorporate DS mechanisms in applications. Therefore, existing applications need not be modified to use DS.
- DS provides a built-in aggregation mechanism. All traffic with the same DS octet is treated the same by the network service. For example, multiple voice connections are not handled individually but in the aggregate. This provides for good scaling to larger networks and traffic loads.
- DS is implemented in individual routers by queuing and forwarding packets based on the DS octet. Routers deal with each packet individually and do not have to save state information on packet flows.

Services

The DS TOS is provided within a DS domain, which is defined as a contiguous portion of the Internet over which a consistent set of DS policies is administered. Typically, a DS domain is under the control of one administrative entity.

The services provided across a DS domain are defined in an SLA, which is a service contract between a customer and the service provider that specifies the forwarding service that the customer should receive for various classes of packets. A customer may be a user organization or another DS domain.

Once the SLA is established, the customer submits packets with the DS octet marked to indicate the packet class. The service provider must assure that the customer gets at least the agreed QoS for each packet class. To provide that QoS, the service provider must configure the appropriate forwarding policies at each router (based on DS octet value) and must measure the performance being provided each class on an ongoing basis.

If a customer submits packets intended for destinations within the DS domain, the DS domain is expected to provide the agreed service. If the destination is beyond the customer's DS domain, the DS domain will attempt to forward the packets through other domains, requesting the appropriate service to match the requested service.

A draft DS framework document lists the following detailed performance parameters that might be included in an SLA:

- Detailed service performance parameters, such as expected throughput, drop probability, and latency.
- Constraints on the ingress and egress points at which the service is provided, indicating the scope of the service.
- Traffic profiles that must be adhered to for the requested service to be provided, such as token bucket parameters.
- Disposition of traffic submitted in excess of the specified profile.

The framework document also gives some examples of services that might be provided:

- Traffic offered at service level A will be delivered with low latency.
- Traffic offered at service level B will be delivered with low loss.
- 90% of in-profile traffic delivered at service level C will experience no more than 50-ms latency.
- 95% of in-profile traffic delivered at service level D will be delivered.
- Traffic offered at service level E will be allotted twice the bandwidth of traffic delivered at service level F.
- Traffic with drop precedence X has a higher probability of delivery than traffic with drop precedence Y.

The first two examples are qualitative and are valid only in comparison to other traffic, such as default traffic that gets a best-effort service. The next two examples are quantitative and provide a specific guarantee that can be verified by measurement on the actual service without comparison to any other services offered at the same time. The final two examples are a mixture of quantitative and qualitative.

DS octet

Packets are labeled for service handling by means of the DS octet, which is placed in the TOS field of an IPv4 header or the traffic class field of the IPv6 header. RFC 2474 defines the DS octet as having the following format: the left-most 6 bits form a DS codepoint, and the right-most 2 bits are currently unused. The DS codepoint is the DS label used to classify packets for differentiated services.

With a 6-bit codepoint, sixty-four different classes of traffic can be defined. These sixty-four codepoints are allocated across three pools of codepoints, as follows:

- Codepoints of the form xxxxx0, where x is either 0 or 1, are reserved for assignment as standards.
- Codepoints of the form xxxx11 are reserved for experimental or local use.
- Codepoints of the form xxxx01 are also reserved for experimental or local use, but may be allocated for future standards action as needed.

Within the first pool, several assignments are made in RFC 2474. The codepoint 000000 is the default packet class. The default class is the best-effort forwarding behavior in existing routers.

Such packets are forwarded in the order that they are received as soon as link capacity becomes available. If higher-priority packets in other DS classes are available for transmission, these are given preference over best-effort default packets.

Codepoints of the form xxx000 are reserved to provide backward compatibility with the IPv4 precedence service. To explain this requirement, we need to digress to an explanation of the IPv4 precedence service. The IPv4 TOS field includes two subfields: a 3-bit precedence subfield and a 4-bit TOS subfield. These subfields serve complementary functions. The TOS subfield provides guidance to the IP entity (in the source or router) on selecting the next hop for this datagram, and the precedence subfield provides guidance about the relative allocation of router resources for this datagram.

The precedence field is set to indicate the degree of urgency or priority to be associated with a datagram. If a router supports the precedence subfield, there are three approaches to responding:

- *Route selection.* A particular route may be selected if the router has a smaller queue for that route or if the next hop on that route supports network precedence or priority (such as a Token Ring network supports priority).
- *Network service.* If the network on the next hop supports precedence, then that service is invoked.

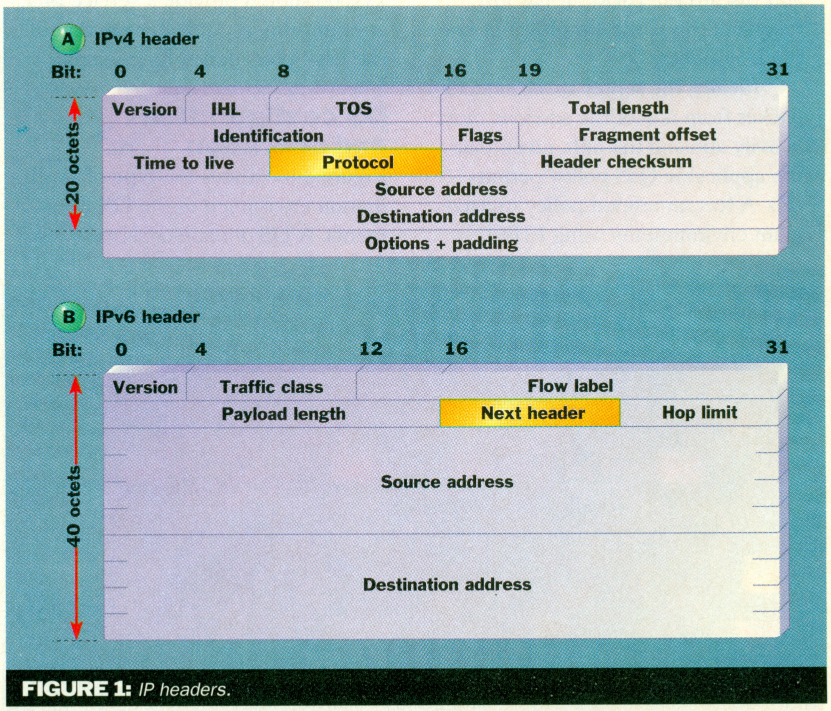

FIGURE 1: *IP headers.*

- *Queuing discipline.* A router may use precedence to affect how queues are handled. For example, a router may give preferential treatment in queues to datagrams with higher precedence.

RFC 1812, "Requirements for IP Version 4 Routers," provides recommendations for queuing discipline that fall into two categories. (As with many RFCs, the words *must, may,* and *should* are used here with very specific meanings: *must* means that a certain action is required; *may* means that an action is permissible, but not required; *should* means that an action is highly recommended, but not an absolute requirement.)

- *Queue service.* Routers *should* implement precedence-ordered queue service. Precedence-ordered queue service means that when a packet is selected for output on a (logical) link, the packet of highest precedence that has been queued for that link is sent. Any router *may* implement other policy-based throughput management procedures that result in other than strict precedence ordering, but it *must* be configurable to suppress them (such as use strict ordering).
- *Congestion control.* When a router receives a packet beyond its storage capacity, it *must* discard it or some other packet or packets. A router *may* discard the packet it has just received; this is the simplest but not the best policy.

Ideally, the router *should* select a packet from one of the sessions most heavily abusing the link, given that the applicable QoS policy permits this. A recommended policy in datagram environments using FIFO queues is to discard a packet randomly selected from the queue. An equivalent algorithm in routers using fair queues is to discard from the longest queue. A router *may* use these algorithms to determine which packet to discard.

If precedence-ordered queue service is implemented and enabled, the router *must not* discard a packet whose IP precedence is higher than that of a packet that is not discarded.

A router *may* protect packets whose IP headers request the maximize reliability TOS, except where doing so would be in violation of the previous rule.

A router *may* protect fragmented IP packets on the theory that dropping a fragment of a datagram may increase congestion by causing all fragments of the datagram to be retransmitted by the source.

To help prevent routing perturbations or disruption of management functions, the router *may* protect packets used for routing control, link control, or network management from being discarded. Dedicated routers (routers that are not also general-purpose hosts, and terminal servers) can achieve an approximation of this rule by protecting packets whose source or destination is the router itself.

The DS codepoints of the form xxx000 should provide a service that, at minimum, is equivalent to that of the IPv4 precedence functionality.

DS configuration and operation

Figure 2 illustrates the type of configuration envisioned in the DS documents. A DS domain consists of a set of contiguous routers; it is possible to get from any router in the domain to any other router in the domain by a path that does not include routers outside the domain. Within a domain, the interpretation of DS codepoints is uniform, so that a uniform, consistent service is provided.

Routers in a DS domain are either boundary nodes or interior nodes. Typically, the interior nodes implement simple mechanisms for handling packets based on their DS codepoint values. This includes queuing discipline to give preferential treatment depending on codepoint value, and packet dropping rules to dictate which packets should be dropped first in the event of buffer saturation. The DS specifications refer to the forwarding treatment provided at a router as per-hop behavior (PHB).

This PHB must be available at all routers. PHB is typically the only part of DS implemented in interior routers. The boundary nodes include PHB mechanisms but also more sophisticated traffic conditioning mechanisms required to provide the desired service. Thus, interior routers have minimal functionality and minimal overhead in providing the DS service, while most of the complexity is in the boundary nodes. The boundary node function can also be provided by a host system attached to the domain, on behalf of the applications at that host system.

The traffic conditioning function consists of five elements:

- *Classifier.* This element separates submitted packets into different classes. This is the foundation for differentiated services. A classifier may separate traffic only on the basis of the DS codepoint (behavior aggregate classifier) or based on multiple fields within the packet header or even the packet payload (making it a multifield classifier).
- *Meter.* A meter element measures submitted traffic for conformance to a profile. It determines whether a given packet stream class is within or exceeds the service level guaranteed for that class.
- *Marker.* This element polices traffic by re-marking packets with a different codepoint as needed. This may be done for packets that exceed the profile. For example, if

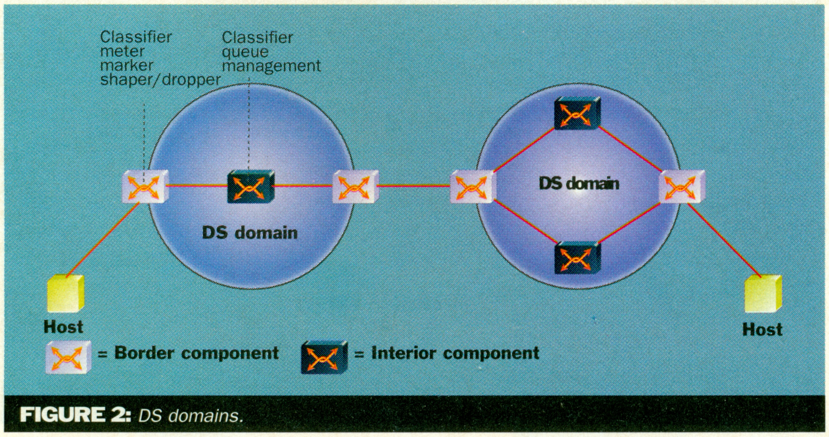

FIGURE 2: *DS domains.*

a given throughput is guaranteed for a particular service class, any packets in that class that exceed the throughput in some defined time interval may be re-marked for best-effort handling. Also, re-marking may be required at the boundary between two DS domains. For example, if a given traffic class is to receive the highest supported priority — and this corresponds to a value of three in one domain and seven in the next domain — then packets with a priority three value traversing the first domain are remarked as priority seven when entering the second domain.

- *Shaper*: Polices traffic by delaying packets as necessary so that the packet stream in a given class does not exceed the traffic rate specified in the profile for that class.
- *Dropper*: Drops packets when the packet rates of a given class exceed that specified in the profile for that class.

Figure 3 illustrates the relationship between the elements of traffic conditioning. After a flow is classified, its resource consumption must be measured. The metering function measures the volume of packets over a particular time interval to determine a flow's compliance with the traffic agreement. If the host is bursty, a simple data rate or packet rate may not be sufficient to capture the desired traffic characteristics. A token bucket scheme is an example of a way to define a traffic profile to take into account both packet rate and burstiness.

A token bucket traffic specification consists of two parameters: a token replenishment rate R and a bucket size B. The token rate R specifies the continually sustainable data rate. Over a long period of time, the average data rate to be supported for this flow is R. The bucket size B specifies the amount by which the data rate can exceed R for short periods of time. The exact condition is as follows: during any time period T, the amount of data sent cannot exceed $RT + B$.

Figure 4 illustrates this scheme and explains the use of the term "bucket." The bucket represents a counter that indicates the allowable number of octets of IP data that can be sent at any time. The bucket fills with octet tokens at the rate of R (the counter is incremented R times per second), up to the bucket capacity (up to the maximum counter value). IP packets arrive and are queued for processing. An IP packet may be processed if there are sufficient octet tokens to match the IP data size. If so, the packet is processed and the bucket is drained of the corresponding number of tokens. If a packet arrives and there are insufficient tokens available, then the packet exceeds the limit for this flow.

Over the long run, the rate of IP data allowed by the token bucket is R. However, if there is an idle or slow period, the bucket capacity builds up so that, at most, an additional B octets above the stated rate can be accepted. Thus, B is a measure of the degree of burstiness of the data flow that is allowed.

If a traffic flow exceeds some profile, several approaches can be taken. Individual packets in excess of the profile may be re-marked for lower-quality handling and allowed to pass into the domain. A traffic shaper may absorb a burst of packets in a buffer and pace the packets over a longer period of time. A dropper may drop packets if the buffer used for pacing becomes saturated.

Enabling QoS

As the Internet and private internets grow in scale, a host of new demands march steadily into view. Low-volume Telnet conversations are leapfrogged by high-volume client/server applications. Recently, tremendous volumes of Web traffic, which is increasingly graphics intensive, have been generated. Now, real-time voice and video applications add to the burden on the Internet.

To cope with these demands, it is not enough to increase Internet capacity. Sensible and effective methods for managing the traffic and controlling congestion are needed. Differentiated services provide a simple tool for delivering different levels of QoS to different groups of applications and users.

William Stallings is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. This article is based on material in the author's latest book, Data and Computer Communications, Sixth Edition *(Prentice-Hall, 2000). He can be reached at ws@shore.net.*

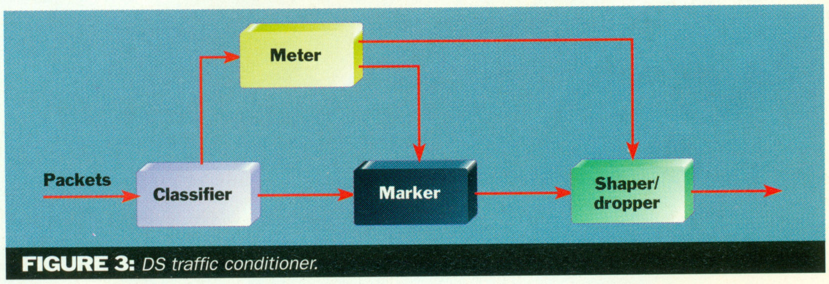

FIGURE 3: *DS traffic conditioner.*

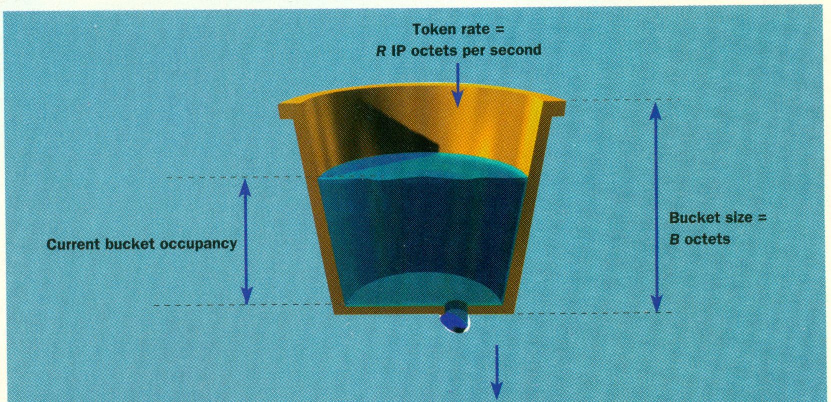

FIGURE 4: *Token bucket scheme.*