Network Working Group
Request for Comments: 2638
Category: Informational

K.Nichols
V. Jacobson
Cisco
L. Zhang
UCLA
July, 1999

A Two-bit Differentiated Services Architecture for the Internet

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document was originally submitted as an internet draft in November of 1997. As one of the documents predating the formation of the IETF's Differentiated Services Working Group, many of the ideas presented here, in concert with Dave Clark's subsequent presentation to the December 1997 meeting of the IETF Integrated Services Working Group, were key to the work which led to RFCs 2474 and 2475 and the section on allocation remains a timely proposal. For this reason, and to provide a reference, it is being submitted in its original form. The forwarding path portion of this document is intended as a record of where we were at in late 1997 and not as an indication of future direction. In particular, this document uses the term "service" to denote a more primitive concept than in current use in the Differentiated Services Working Group.

The postscript version of this document includes Clark's slides as an appendix. The postscript version of this document also includes many figures that aid greatly in its readability.

1. Introduction

This document presents a differentiated services architecture for the internet. Dave Clark and Van Jacobson each presented work on differentiated services at the Munich IETF meeting [2,3]. Each explained how to use one bit of the IP header to deliver a new kind of service to packets in the internet. These were two very different kinds of service with quite different policy assumptions. Ensuing discussion has convinced us that both service types have merit and that both service types can be implemented with a set of very similar mechanisms.[1] We propose an architectural framework that permits the use of both of these service types and exploits their similarities in forwarding path mechanisms. The major goals of this architecture are each shared with one or both of those two proposals: keep the forwarding path simple, push complexity to the edges of the network to the extent possible, provide a service that avoids assumptions about the type of traffic using it, employ an allocation policy that will be compatible with both long-term and short-term provisioning, make it possible for the dominant Internet traffic model to remain best-effort.

The major contributions of this document are to present two distinct service types, a set of general

1. The underlying similarity is not an accident -- Dave Clark first presented parts of his architecture in the early 90s and Jacobson was heavily influenced by the simplicity and scalability of Clark's model.

mechanisms for the forwarding path that can be used to implement a range of differentiated services and to propose a flexible framework for provisioning a differentiated services network. It is precisely this kind of architecture that is needed for expedient deployment of differentiated services: we need a framework and set of primitives that can be implemented in the short-term and provide interoperable services, yet can provide a "sandbox" for experimentation and elaboration that can lead in time to more levels of differentiation within each service as needed.

At the risk of belaboring an analogy, we are motivated to provide services tiers in somewhat the same fashion as the airlines do with first class, business class and coach class. The latter also has tiering built in due to the various restrictions put on the purchase. A part of the analogy we want to stress is that best effort traffic, like coach class seats on an airplane, is still expected to make up the bulk of internet traffic. Business and first class carry a small number of passengers, but are quite important to the economics of the airline industry. The various economic forces and realities combine to dictate the relative allocation of the seats and to try to fill the airplane. We don't expect that differentiated services will comprise all the traffic on the internet, but we do expect that new services will lead to a healthy economic and service environment.

This document is organized into sections describing service architecture, mechanisms, the bandwidth allocation architecture, how this architecture might interoperate with RSVP/int-serv work, and gives recommendations for deployment.

2. Architecture

2.1 Background

The current internet delivers one type of service, best-effort, to all traffic. A number of proposals have been made concerning the addition of enhanced services to the Internet. We focus on two particular methods of adding a differentiated level of service to IP, each designated by one bit [1,2,3]. These services represent a radical departure from the Internet's traditional service, but they are also a radical departure from traditional "quality of service" architectures which rely on circuit-based models. Both these proposals seek to define a single common mechanism that is used by interior network routers, pushing most of the complexity and state of differentiated services to the network edges. Both use bandwidth as the resource that is being requested and allocated. Clark and Wroclawski defined an "Assured" service that follows "expected capacity" usage profiles that are statistically provisioned [3]. The assurance that the user of such a service receives is that such traffic is unlikely to be dropped as long as it stays within the expected capacity profile. The exact meaning of "unlikely" depends on how well provisioned the service is. An Assured service traffic flow may exceed its Profile, but the excess traffic is not given the same assurance level. Jacobson defined a "Premium" service that is provisioned according to peak capacity Profiles that are strictly not oversubscribed and that is given its own high-priority queue in routers [2]. A Premium service traffic flow is shaped and hard-limited to its provisioned peak rate and shaped so that bursts are not injected into the network. Premium service presents a "virtual wire" where a flow's bursts may queue at the shaper at the edge of the network, but thereafter only in proportion to the indegree of each router. Despite their many similarities, these two approaches result in fundamentally different services. The former uses buffer management to provide a "better effort" service while the latter creates a service with little jitter and queueing delay and no need for queue management on the Premium packets' queue.

An Assured service was introduced in [3] by Clark and Wroclawski, though we have made some alterations in its specification for our architecture. Further refinements and an "Expected Capacity" framework are given in Clark and Fang [10]. This framework is focused on "providing different levels of best-effort service at times of network congestion" but also mentions that it is possible to have a separate router queue to implement a "guaranteed" level of assurance. We believe this framework and our Two-bit architecture are compatible but this needs further exploration. As Premium service has not been documented elsewhere, we describe it next and follow this with a description of the two-bit architecture.

2.2 Premium service

In [2], a Premium service was presented that is fundamentally different from the Internet's current best effort service. This service is not meant to replace best effort but primarily to meet an emerging demand for a commercial service that can share the network with best effort traffic. This is desirable economically, since the same network can be used for both kinds of traffic. It is expected that Premium traffic would be allocated a small percentage of the total network capacity, but that it would be priced much higher. One use of such a service might be to create "virtual leased lines", saving the cost of building and maintaining a separate network. Premium service, not unlike a standard telephone line, is a capacity which the customer expects to be there when the receiver is lifted, although it may, depending on the household, be idle a good deal of the time. Provisioning Premium traffic in this way reduces the capacity of the best effort internet by the amount of Premium allocated, in the worst case, thus it would have to be priced accordingly. On the other hand, whenever that capacity is not being used it is available to best effort traffic. In contrast to normal best effort traffic which is bursty and requires queue management to deal fairly with congestive episodes, this Premium service *by design* creates very regular traffic patterns and small or nonexistent queues.

Premium service levels are specified as a desired peak bit-rate for a specific flow (or aggregation of flows). The user contract with the network is not to exceed the peak rate. The network contract is that the contracted bandwidth will be available when traffic is sent. First-hop routers (or other edge devices) filter the packets entering the network, set the Premium bit of those that match a Premium service specification, and perform traffic shaping on the flow that smooths all traffic bursts before they enter the network. This approach requires no changes in hosts. A compliant router along the path needs two levels of priority queueing, sending all packets with the Premium bit set first. Best-effort traffic is unmarked and queued and sent at the lower priority. This results in two "virtual networks": one which is identical to today's Internet with buffers designed to absorb traffic bursts; and one where traffic is limited and shaped to a contracted peak-rate, but packets move through a network of queues where they experience almost no queueing delay.

In this architecture, forwarding path decisions are made separately and more simply than the setting up of the service agreements and traffic profiles. With the exception of policing and shaping at administrative or "trust" boundaries, the only actions that need to be handled in the forwarding path are to classify a packet into one of two queues on a single bit and to service the two queues using simple priority. Shaping must include both rate and burst parameters; the latter is expected to be small, in the one or two packet range. Policing at boundaries enforces rate compliance, and may be implemented by a simple token bucket. The admission and set-up procedures are expected to evolve, in time, to be dynamically configurable and fairly complex while the mechanisms in the forwarding path remain simple.

A Premium service built on this architecture can be deployed in a useful way once the forwarding path mechanisms are in place by making static allocations. Traffic flows can be designated for special treatment through network management configuration. Traffic flows should be designated by the source, the destination, or any combination of fields in the packet header. First-hop (or leaf) routers will filter flows on all or part of the header tuple consisting of the source IP address, destination IP address, protocol identifier, source port number, and destination port number. Based on this classification, a first-hop router performs traffic shaping and sets the designated Premium bit of the precedence field. End-hosts are thus not required to be "differentiated services aware", though if and when end-systems become universally "aware", they might do their own shaping and first-hop routers merely police.

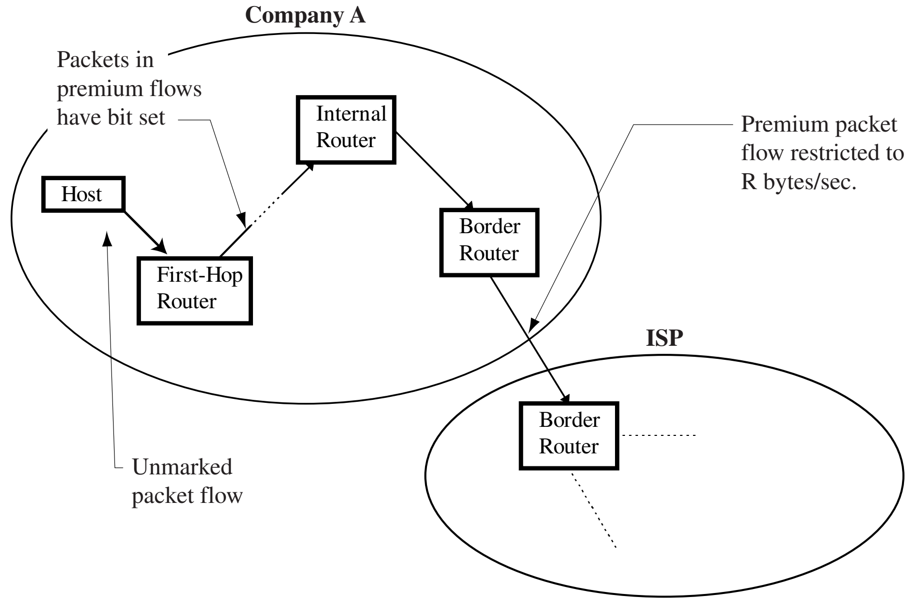
Adherence to the subscribed rate and burst size must be enforced at the entry to the network, either by the end-system or by the first-hop router. Within an intranet, administrative domain, or "trust region" the packets can then be classified and serviced solely on the Premium bit. Where packets cross a boundary, the policing function is critical. The entered region will check the prioritized packet flow for conformance to a rate the two regions have agreed upon, discarding packets that exceed the rate[1]. It is thus in the best interests of a region to ensure conformance to the agreed-upon rate at the egress. This requirement means that Premium traffic is burst-free and, together with the no oversubscription rule, leads directly to the observation that Premium queues can easily be sized to prevent the need to drop packets and thus the need for a queue management policy. At each router, the largest queue size is related to the in-degree of other routers and is thus quite small, on the order of ten packets.

Premium bandwidth allocations must not be oversubscribed as they represent a commitment by the network and should be priced accordingly. Note that, in this architecture, Premium traffic will also experience considerably less delay variation than either best effort traffic or the Assured data traffic of [3]. Premium rates might be configured on a subscription basis in the near-term, or on-demand when dynamic set-up or signaling is available.

Figure 1 shows how a Premium packet flow is established within a particular administrative domain, Company A, and sent across the access link to Company A's ISP. Assume that the host's first-hop router has been configured to match a flow from the host's IP address to a destination IP address that is reached through ISP. A Premium flow is configured from a host with a rate which is both smaller than the total Premium allocation Company A has from the ISP, *r* bytes per second, and smaller than the amount of that allocation which has been assigned to other hosts in Company A. Packets are not marked in any special way when they leave the host. The first-hop router clears the Premium bit on all arriving packets, sets the Premium bit on all packets in the designated flow, shapes packets in the Premium flow to a configured rate and burst size, queues best-effort unmarked packets in the low priority queue and shaped Premium packets in the high priority queue, and sends packets from those two queues at simple priority. Intermediate routers internal to Company A enqueue packets in one of two output queues based on the Premium bit and service the queues with simple priority. Border routers perform quite different tasks, depending on whether they are processing an egress flow or an ingress flow. An egress border router *may* perform some reshaping on the aggregate Premium traffic to conform to rate *r*, depending on the number of Premium flows aggregated. Ingress border routers only need to perform a simple policing function that can be implemented with a token bucket. In the example, the ISP accepts all Premium packets from A as long as the flow does not exceed *r* bytes per second.

1. An alternative strategy is to downgrade the priority of non-compliant packets. This has the effect of undermining the disincentives for Premium flows to stay within profile, causing out-of-order packet delivery, and leads to performance ambiguity, thus this strategy is not compatible with this service.

Figure 1. Premium traffic flow from end-host to organization's ISP



2.3 Two-bit differentiated services architecture

Clark's and Jacobson's proposals are markedly similar in the location and type of functional blocks that are needed to implement them. Furthermore, they implement quite different services which are not incompatible in a network. The Premium service implements a guaranteed peak bandwidth service with negligible queueing delay that cannot starve best effort traffic and can be allocated in a fairly straightforward fashion. This service would seem to have a strong appeal for commercial applications, video broadcasts, voice-over-IP, and VPNs. On the other hand, this service may prove both too restrictive (in its hard limits) and overdesigned (no overallocation) for some applications. The Assured service implements a service that has the same delay characteristics as (undropped) best effort packets and the firmness of its guarantee depends on how well individual links are provisioned for bursts of Assured packets. On the other hand, it permits traffic flows to use any additional available capacity without penalty and occasional dropped packets for short congestive periods may be acceptable to many users. This service might be what an ISP would provide to individual customers who are willing to pay a bit more for internet service that seems unaffected by congestive periods. Both services are only as good as their admission control schemes, design of which can be more difficult for traffic which is not peak-rate allocated.

There may be some additional benefits of deploying both services. To the extent that Premium service is a conservative allocation of resources, unused bandwidth that had been allocated to Premium might provide some "headroom" for underallocated or burst periods of Assured traffic or for best effort. Network elements that deploy both services will be performing RED queue management on all non-Premium traffic, as suggested in [4], and the effects of mixing the Premium streams with best effort might serve to reduce burstiness in the latter. A strength of the Assured service is that it allows bursts to happen in their natural fashion, but this also makes the provisioning, admission control and allocation problem more difficult so it may take more time and experimentation before this admission policy for this service is completely defined. A Premium service could be deployed that employs static allocations on peak rates with no statistical sharing.

As there appear to be a number of advantages to an architecture that permits these two types of service and because, as we shall see, they can be made to share many of the same mechanisms, we propose designating two bit-patterns from the IP header precedence field. We leave the explicit designation of these bit-patterns to the standards process thus we use the shorthand notation of denoting each pattern by a bit, one we will call the Premium or P-bit, the other we call the assurance or A-bit. It is possible for a network to implement only one of these services and to have network elements that only look at the one applicable bit, but we focus on the two service architecture. Further, we assume the case where no changes are made in the hosts, appropriate packet marking all being done in the network, at the first-hop, or leaf, router. We describe the forwarding path architecture in this section, assuming that the service has been allocated through mechanisms we will discuss in section 4.

In a more general sense, Premium service denotes packets that are enqueued at a higher priority than the ordinary best-effort queue. Similarly, Assured service denotes packets that are treated preferentially with respect to the dropping probability within the "normal" queue. There are a number of ways to add more service levels within each of these service types [7], but this document takes the position of specifying the base-level services of Premium and Assured.

The forwarding path mechanisms can be broken down into those that happen at the input interface, before packet forwarding, and those that happen at the output interface, after packet forwarding. Intermediate routers only need to implement the post packet forwarding functions, while leaf and border routers must perform functions on arriving packets before forwarding. We describe the mechanisms this way for illustration; other ways of composing their functions are possible.

Leaf routers are configured with a traffic profile for a particular flow based on its packet header. This functionality has been defined by the RSVP Working Group in RFC 2205. Figure 2 shows what happens to a packet that arrives at the leaf router, before it is passed to the forwarding engine. All arriving packets must have both the A-bit and the P-bit cleared after which packets are classified on their header. If the header does not match any configured values, it is immediately forwarded. Matched flows pass through individual Markers that have been configured from the usage profile for that flow: service class (Premium or Assured), rate (peak for Premium, "expected" for Assured), and permissible burst size (may be optional for Premium). Assured flow packets emerge from the Marker with their A-bits set when the flow is in conformance to its Profile, but the flow is otherwise unchanged. For a Premium flow, the Marker will hold packets when necessary to enforce their configured rate. Thus Premium flow packets emerge from the Marker in a shaped flow with their P-bits set. (It is possible for Premium flow packets to be dropped inside of the Marker as we describe below.) Packets are passed to the forwarding engine when they emerge from Markers. Packets that have either their P or A bits set we will refer to as Marked packets.

Figure 2. Block diagram of leaf router input functionality

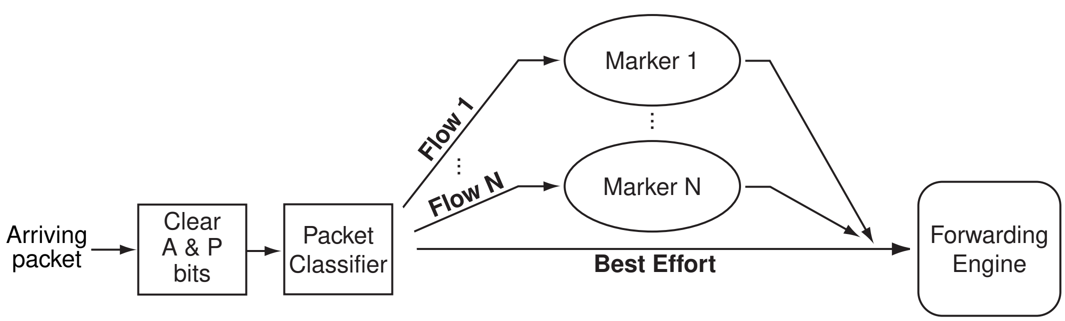
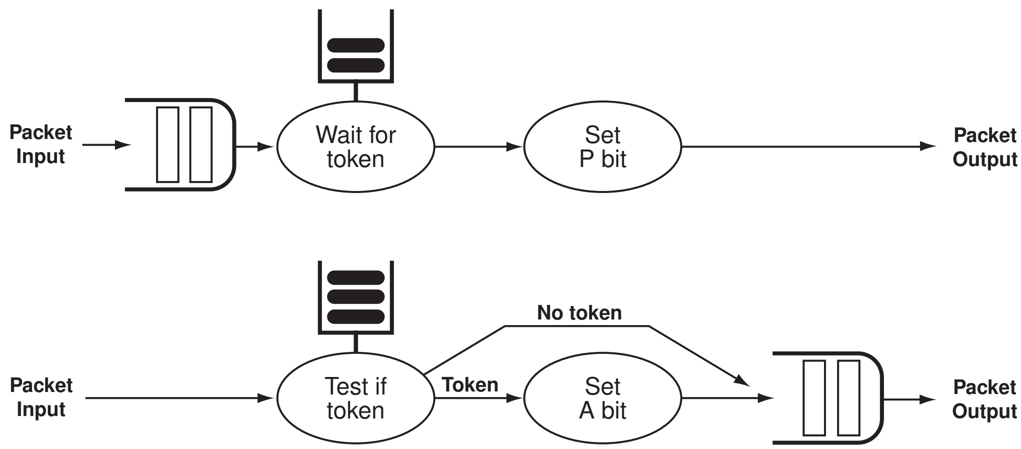


Figure 3 shows the inner workings of the Marker. For both Assured and Premium packets, a token bucket "fills" at the flow rate that was specified in the usage profile. For Assured service, the token bucket depth is set by the Profile's burst size. For Premium service, the token bucket depth must be limited to the equivalent of only one or two packets. (We suggest a depth of one packet in early deployments.) When a token is present, Assured flow packets have their A-bit set to one, otherwise the packet is passed to the forwarding engine. For Premium-configured Marker, arriving packets that see a token present have their P-bits set and are forwarded, but when no token is present, Premium flow packets are held until a token arrives. If a Premium flow bursts enough to overflow the holding queue, its packets will be dropped. Though the flow set up data can be used to configure a size limit for the holding queue (this would be the meaning of a "burst" in Premium service), it is not necessary. Unconfigured holding queues should be capable of holding at least two bandwidth-delay products, adequate for TCP connections. A smaller value might be used to suit delay requirements of a specific application.

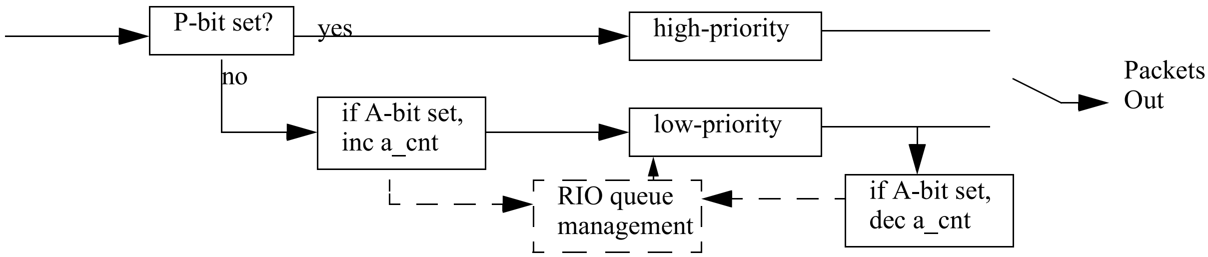
Figure 3. Markers to implement the two different services



In practice, the token bucket should be implemented in bytes and a token is considered to be present if the number of bytes in the bucket is equal or larger to the size of the packet. For Premium, the bucket can only be allowed to fill to the maximum packet size; while Assured may fill to the configured burst parameter. Premium traffic is held until a sufficient byte credit has accumulated and this holding buffer provides the only real queue the flow sees in the network. For Assured, traffic, we just test if the bytes in the bucket are sufficient for the packet size and set A if so. If not, the only difference is that A is not set. Assured traffic goes into a queue following this step and potentially sees a queue at every hop along its path.

Each output interface of a router must have two queues and must implement a test on the P-bit to select a packet's output queue. The two queues must be serviced by simple priority, Premium packets first. Each output interface must implement the RED-based RIO mechanism described in [3] on the lower priority queue. RIO uses two thresholds for when to begin dropping packets, a lower one based on total queue occupancy for ordinary best effort traffic and one based on the number of packets enqueued that have their A-bit set. This means that any action preferential to Assured service traffic will only be taken when the queue's capacity exceeds the threshold value for ordinary best effort service. In this case, only unmarked packets will be dropped (using the RED algorithm) unless the threshold value for Assured service is also reached. Keeping an accurate count of the number of A-bit packets currently in a queue requires either testing the A-bit at both entry and exit of the queue or some additional state in the router. Figure 4 is a block diagram of the output interface for all routers.

Figure 4. Router output interface for two-bit architecture

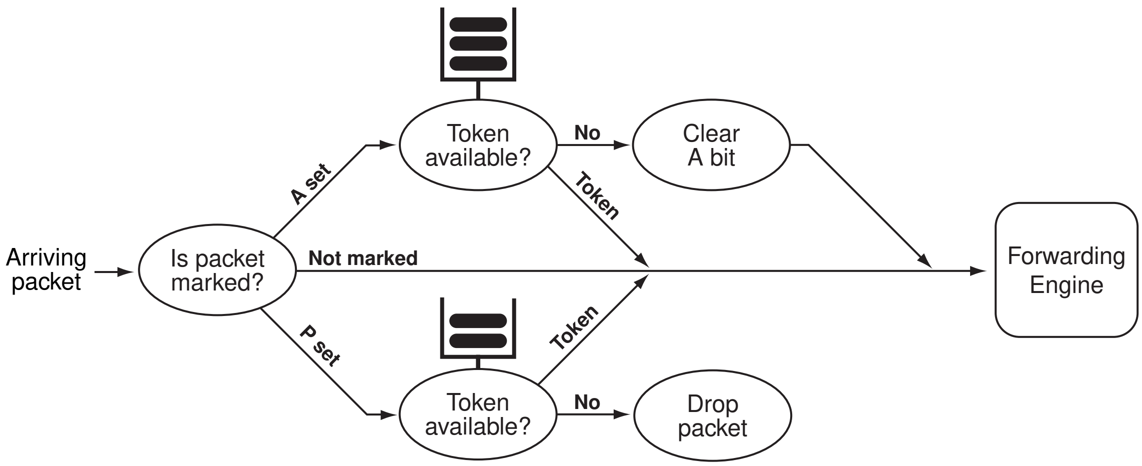


The packet output of a leaf router is thus a shaped stream of packets with P-bits set mingled with an unshaped best effort stream of packets, some of which may have A-bits set. Premium service clearly cannot starve best effort traffic because it is both burst and bandwidth controlled. Assured service might rely only on a conservative allocation to prevent starvation of unmarked traffic, but bursts of Assured traffic might then close out best-effort traffic at bottleneck queues during congestive periods.

After [3], we designate the forwarding path objects that test flows against their usage profiles "Profile Meters". Border routers will require Profile Meters at their input interfaces. The bilateral agreement between adjacent administrative domains must specify a peak rate on all P traffic and a rate and burst for A traffic (and possibly a start time and duration). A Profile Meter is required at the ingress of a trust region to ensure that differentiated service packet flows are in compliance with their agreed-upon rates. Non-compliant packets of Premium flows are discarded while non-compliant packets of Assured flows have their A-bits reset. For example, in figure 1, if the ISP has agreed to supply Company A with *r* bytes/sec of Premium service, P-bit marked packets that enter the ISP through the link from Company A will be dropped if they exceed *r*. If instead, the service in figure 1 was Assured service, the packets would simply be unmarked, forwarded as best effort.

The simplest border router input interface is a Profile Meter constructed from a token bucket configured with the contracted rate across that ingress link (see figure 5). Each type, Premium or Assured, and each interface must have its own profile meter corresponding to a particular class across a particular boundary. (This is in contrast to models where *every flow* that crosses the boundary must be separately policed and/or shaped.) The exact mechanisms required at a border router input interface depend on the allocation policy deployed; a more complex approach is presented in section 4.

Figure 5. Border router input interface Profile Meters



3. Mechanisms

3.1 Forwarding Path Primitives

Section 2.3 introduced the forwarding path objects of Markers and Profile Meters. In this section we specify the primitive building blocks required to compose them. The primitives are: general classifier, bit-pattern classifier, bit setter, priority queues, policing token bucket and shaping token bucket. These primitives can compose a Marker (either a policing or a shaping token bucket plus a bit setter) and a Profile Meter (a policing token bucket plus a dropper or bit setter).

General Classifier. Leaf or first-hop routers must perform a transport-level signature matching based on a tuple in the packet header, a functionality which is part of any RSVP-capable router. As described above, packets whose tuples match one of the configured flows are conformance tested and have the appropriate service bit set. This function is memory- and processing-intensive, but is kept at the edges of the network where there are fewer flows.

Bit-pattern classifier. This primitive comprises a simple two-way decision based on whether a particular bit-pattern in the IP header is set or not. As in figure 4, the P-bit is tested when a packet arrives at a non-leaf router to determine whether to enqueue it in the high priority output queue or the low priority packet queue. The A-bit of packets bound for the low priority queue is tested to 1) increment the count of Assured packets in the queue if set and 2) determine which drop probability will be used for that packet. Packets exiting the low priority queue must also have the A-bit tested so that the count of enqueued Assured packets can be decremented if necessary.

Bit setter. The A-bits and P-bits must be set or cleared in several places. A functional block that sets the appropriate bits of the IP header to a configured bit-pattern would be the most general.

Priority queues. Every network element must include (at least) two levels of simple priority queueing. The high priority queue is for the Premium traffic and the service rule is to send packets in that queue first and to exhaustion. Recall that Premium traffic must never be oversubscribed, thus Premium traffic should see little or no queue.

Shaping token bucket. This is the token bucket required at the leaf router for Premium traffic and shown in figure 3. As we shall see, shaping is also useful at egress points of a trust region. An arriving packet is immediately forwarded if there is a token present in the bucket, otherwise the packet is enqueued until the bucket contains tokens sufficient to send it. Shaping requires clocking mechanisms, packet memory, and some state block for each flow and is thus a memory and computation-intensive process.

Policing token bucket. This is the token bucket required for Profile Meters and shown in figure 5. Policing token buckets never hold arriving packets, but check on arrival to see if a token is available for the packet's service class. If so, the packet is forwarded immediately. If not, the policing action is taken, dropping for Premium and reclassifying or unmarking for Assured.

3.2 Passing configuration information

Clearly, mechanisms are required to communicate the information about the request to the leaf router. This configuration information is the rate, burst, and whether it is a Premium or Assured type. There may also need to be a specific field to set or clear this configuration. This information can be passed in a number of ways, including using the semantics of RSVP, SNMP, or directly set by a network administrator in some other way. There must be some mechanisms for authenticating the sender of this information. We expect configuration to be done in a variety of ways in early deployments and a protocol and mechanism for this to be a topic for future standards work.

3.3 Discussion

The requirements of shapers motivate their placement at the edges of the network where the state per router can be smaller than in the middle of a network. The greatest burden of flow matching and shaping will be at leaf routers where the speeds and buffering required should be less than those that might be required deeper in the network. This functionality is *not* required at every network element on the path. Routers that are internal to a trust region will not need to shape traffic. Border routers may need or desire to shape the aggregate flow of Marked packets at their egress in order to ensure that they will not burst into non-compliance with the policing mechanism at the ingress to the other domain (though this may not be necessary if the in-degree of the router is low). Further, the shaping would be applied to an aggregation of all the Premium flows that exit the domain via that path, not to each flow individually.

These mechanisms are within reach of today's technology and it seems plausible to us that Premium and Assured services are all that is needed in the Internet. If, in time, these services are found insufficient, this architecture provides a migration path for delivering other kinds of service levels to traffic. The A- and P-bits would continue to be used to identify traffic that gets Marked service, but further filter matching could be done on packet headers to differentiate service levels further. Using the bits this way reduces the number of packets that have to have further matching done on them rather than filtering every incoming packet. More queue levels and more complex scheduling could be added for P-bit traffic and more levels of drop priority could be added for A-bit traffic if experience shows them to be necessary and processing speeds are sufficient. We propose that the services described here be considered as "at least" services. Thus, a network element should at least be capable of mapping all P-bit traffic to Premium service and of mapping all A-bit traffic to be treated with one level of priority in the "best effort" queue (it appears that the single level of A-bit traffic should map to a priority that is equivalent to the best level in a multi-level element that is also in the path).

On the other hand, what is the downside of deploying an architecture for both classes of service if later experience convinces us that only one of them is needed? The functional blocks of both service classes are similar and can be provided by the same mechanisms, parameterized differently. If Assured service is not used, very little is lost. A RED-managed best effort queue has been strongly recommended in [4] and, to the extent that the deployment of this architecture pushes the deployment of RED-managed best effort queues, it is clearly a positive. If Premium service goes unused, the two-queues with simple priority service is not required and the shaping function of the Marker may be unused, thus these would impose an unnecessary implementation cost.

4. The Architectural Framework for Marked Traffic Allocation

Thus far we have focused on the service definitions and the forwarding path mechanisms. We now turn to the problem of allocating the level of Marked traffic throughout the Internet. We observe that most organizations have fixed portions of their budgets, including data communications, that are determined on an annual or quarterly basis. Some additional monies might be attached to specific projects for discretionary costs that arise in the shorter term. In turn, service providers (ISPs and NSPs) must do their planning on annual and quarterly bases and thus cannot be expected to provide differentiated services purely "on call". Provisioning sets up static levels of Marked traffic while call set-up creates an allocation of Marked traffic for a single flow’s duration. Static levels can be provisioned with time-of-day specifications, but cannot be changed in response to a dynamic message. We expect both kinds of bandwidth allocation to be important. The purchasers of Marked services can generally be expected to work on longer-term budget cycles where these services will be accounted for similarly to many information services today. A mail-order house may wish to purchase a fixed allocation of bandwidth in and out of its web-server to give potential customers a "fast" feel when browsing their site. This allocation might be based on hit rates of the previous quarter or some sort of industry-based averages. In addition, there needs to be a dynamic allocation capability to respond to particular events, such as a demonstration, a network broadcast by a company’s CEO, or a particular network test. Furthermore, a dynamic capability may be needed in order to meet a precommitted service level when the particular source or destination is allowed to be "anywhere on the Internet". "Dynamic" covers the range from a telephoned or e-mailed request to a signalling type model. A strictly statically allocated scenario is expected to be useful in initial deployment of differentiated services and to make up a major portion of the Marked traffic for the forseeable future.

Without a "per call" dynamic set up, the preconfiguring of usage profiles can always be construed as "paying for bits you don’t use" whether the type of service is Premium or Assured. We prefer to think of this as paying for the level of service that one expects to have available at any time, for example paying for a telephone line. A customer might pay an additional flat fee to have the privilege of calling a wide local area for no additional charge or might pay by the call. Although a customer might pay on a "per call" basis for every call made anywhere, it generally turns out not to be the most economical option for most customers. It’s possible similar pricing structures might arise in the internet.

We use Allocation to refer to the process of making Marked traffic commitments anywhere along this continuum from strictly preallocated to dynamic call set-up and we require an Allocation architecture capable of encompassing this entire spectrum in any mix. We further observe that Allocation must follow organizational hierarchies, that is each organization must have complete responsibility for the Allocation of the Marked traffic resource within its domain. Finally, we observe that the only chance of success for incremental deployment lies in an Allocation architecture that is made up of bilateral agreements, as multilateral agreements are much too complex to administer. Thus, the Allocation architecture is made up of agreements across boundaries as to the amount of Marked traffic that will be allowed to pass. This is similar to "settlement" models used today.

4.1 Bandwidth Brokers - Allocating and Controlling Bandwidth Shares

The goal of differentiated services is *controlled* sharing of some organization’s Internet bandwidth. The control can be done independently by individuals, i.e., users set bit(s) in their packets to distinguish their most important traffic, or it can be done by agents that have some knowledge of the organization's priorities and policies and allocate bandwidth with respect to those policies. Independent labeling by individuals is simple to implement but unlikely to be sufficient since it's unreasonable to expect all individuals to know all their organization's priorities and current network use and always mark their traffic accordingly. Thus this architecture is designed with agents called bandwidth brokers (BB) [2], that can be configured with organizational policies, keep track of the current allocation of marked traffic, and interpret new requests to mark traffic in light of the policies and current allocation.

We note that such agents are inherent in any but the most trivial notions of sharing. Neither individuals nor the routers their packets transit have the information necessary to decide which packets are most important to the organization. Since these agents must exist, they can be used to allocate bandwidth for end-to-end connections with far less state and simpler trust relationships than deploying per flow or per filter guarantees in all network elements on an end-to-end path. BBs make it possible for bandwidth allocation to follow organizational hierarchies and, in concert with the forwarding path mechanisms discussed in section 3, reduce the state required to set up and maintain a flow over architectures that require checking the full flow header at every network element. Organizationally, the BB architecture is motivated by the observation that multilateral agreements rarely work and this architecture allows end-to-end services to be constructed out of purely bilateral agreements. BBs only need to establish relationships of limited trust with their peers in adjacent domains, unlike schemes that require the setting of flow specifications in routers throughout an end-to-end path. In practical technical terms, the BB architecture makes it possible to keep state on an administrative domain basis, rather than at every router and the service definitions of Premium and Assured service make it possible to confine per flow state to just the leaf routers.

BBs have two responsibilities. Their primary one is to parcel out their region's Marked traffic allocations and set up the leaf routers within the local domain. The other is to manage the messages that are sent across boundaries to adjacent regions' BBs. A BB is associated with a particular trust region, one per domain[1]. A BB has a policy database that keeps the information on who can do what when and a method of using that database to authenticate requesters. Only a BB can configure the leaf routers to deliver a particular service to flows, crucial for deploying a secure system. If the deployment of Differentiated Services has advanced to the stage where dynamically allocated, marked flows are possible between two adjacent domains, BBs also provide the hook needed to implement this. Each domain's BB establishes a secure association with its peer in the adjacent domain to negotiate or configure a rate and a service class (Premium or Assured) across the shared boundary and through the peer's domain. As we shall see, it is possible for some types of service and particularly in early implementations, that this "secure association" is not automatic but accomplished through human negotiation and subsequent manual configuration of the adjacent BBs according to the negotiated agreement. This negotiated rate is a capability that a BB controls for all hosts in its region.

When an allocation is desired for a particular flow, a request is sent to the BB. Requests include a service type, a target rate, a maximum burst, and the time period when service is required. The request can be made manually by a network administrator or a user or it might come from another region's BB. A BB first authenticates the credentials of the requester, then verifies there exists unallocated bandwidth sufficient to meet the request. If a request passes these tests, the available bandwidth is reduced by the requested amount and the flow specification is recorded. In the case where the flow has a destination outside this trust region, the request must

1. Initially. This can be expanded to a hierarchy of BBs within a domain but only the top level BB would be responsible for communicating accross domain boundaries.

fall within the class allocation through the "next hop" trust region that was established through a bilateral agreement of the two trust regions. The requester's BB informs the adjacent region's BB that it will be using some of this rate allocation. The BB configures the appropriate leaf router with the information about the packet flow to be given a service at the time that the service is to commence. This configuration is "soft state" that the BB will periodically refresh. The BB in the adjacent region is responsible for configuring the border router to permit the allocated packet flow to pass and for any additional configurations and negotiations within and across its borders that will allow the flow to reach its final destination.

At DMZs, there must be an unambiguous way to determine the local source of a packet. An interface's source could be determined from its MAC address which would then be used to classify packets as coming across a logical link directly from the source domain corresponding to that MAC address. Thus with this understanding we can continue to use figures illustrating a single pipe between two different domains.

In this way, all agreements and negotiations are performed between two adjacent domains. An initial request might cause communication between BBs on several domains along a path, but each communication is only between two adjacent BBs. Initially, these agreements will be prenegotiated and fairly static. Some may become more dynamic as the service evolves.

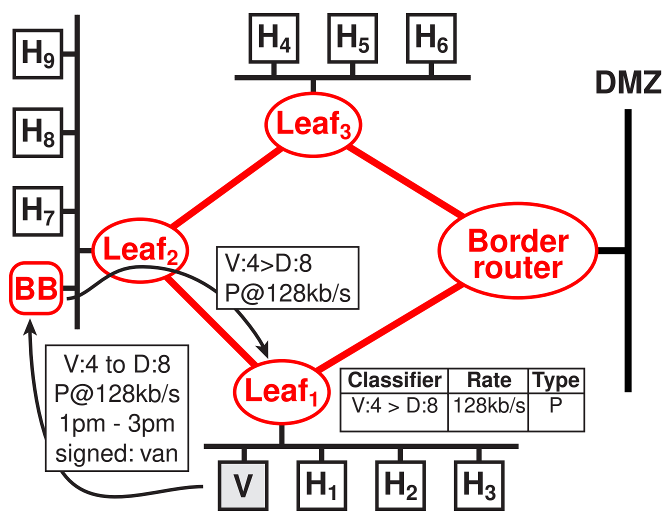
4.2 Examples

This section gives examples of BB transactions in a non-trivial, multi-transit-domain Internet. The BB framework allows operating points across a spectrum from "no signalling across boundaries" to "each flow set up dynamically". We might expect to move across this spectrum over time, as the necessary mechanisms are ubiquitously deployed and BBs become more sophisticated, but the statically allocated portions of the spectrum should always have uses. We believe the ability to support this wide spectrum of choices simultaneously will be important both in incremental deployment and in allowing ISPs to make a wide range of offerings and pricings to users. The examples of this section roughly follow the spectrum of increasing sophistication. Note that we assume that domains contract for some amount of Marked traffic which can be requested as either 'Assured' or 'Premium' in each individual flow setup transaction. The examples say "Marked" although actual transactions would have to specify either Assured or Premium.

A statically configured example with no BB messages exchanged. Here all allocations are statically preallocated through purely bilateral agreements between users (individual TCPs, individual hosts, campus networks, or whole ISPs) [6]. The allocations are in the form of usage profiles of rate, burst, and a time during which that profile is to be active. Users and providers negotiate these Profiles which are then installed in the user domain BB and in the provider domain BB. No BB messages cross the boundary; we assume this negotiation is done by human representatives of each domain. In this case, BBs only have to perform one of their two functions, that of allocating this Profile within their local domain. It is even possible to set all of this suballocations up in advance and then the BB only needs to set up and tear down the Profile at the proper time and to refresh the soft state in the leaf routers. From the user domain BB, the Profile is sent as soft state to the first hop router of the flow during the specified time. These Profiles might be set using RSVP, a variant of RSVP, SNMP, or some vendor-specific mechanism. Although this static approach can work for all Marked traffic, due to the strictly not oversubscribed requirement, it is only appropriate for Premium traffic as long as it is kept to a small percentage of the bottleneck path through a domain or is otherwise constrained to a well-known behavior. Similar restrictions might hold for Assured depending on the expectation associated with the service.

In figure 6, we show an example of setting a Profile in a leaf router. A usage profile has been negotiated with the ISP for the entire domain and the BB parcels it out among individual flows as requested. The leaf router mechanism is that shown in figure 3, with the token bucket set to the parameters from the usage profile. The ISP's BB would configure its own Profile Meter at the ingress router from that customer to ensure the Profile was maintained. This mechanism was shown in figure 5. We assume that the time duration and start times for any Profile to be active are maintained in the BB. The Profile is sent to the ingress device or cleared from the ingress device by messages sent from the BB. In this example, we assume that van@lbl wants to talk to ddc@mit. The LBL-BB is sent a request from Van asking that premium service be assigned to a flow that is designated as having source address "V:4" and going to destination address "D:8". This flow should be configured for a rate of 128kb/sec and allocated from 1pm to 3pm. The request must be "signed" in a secure, verifiable manner. The request might be sent as data to the LBL-BB, an e-mail message to a network administrator, or in a phone call to a network administrator. The LBL-BB receives this message, verifies that there is 128kb/sec of unused Premium service for the domain from 1-3pm, then sends a message to Leaf1 that sets up an appropriate Profile Meter. The message to Leaf1 might be an RSVP message, or SNMP, or some proprietary method. All the domains passed must have sufficient reserve capacity to meet this request.

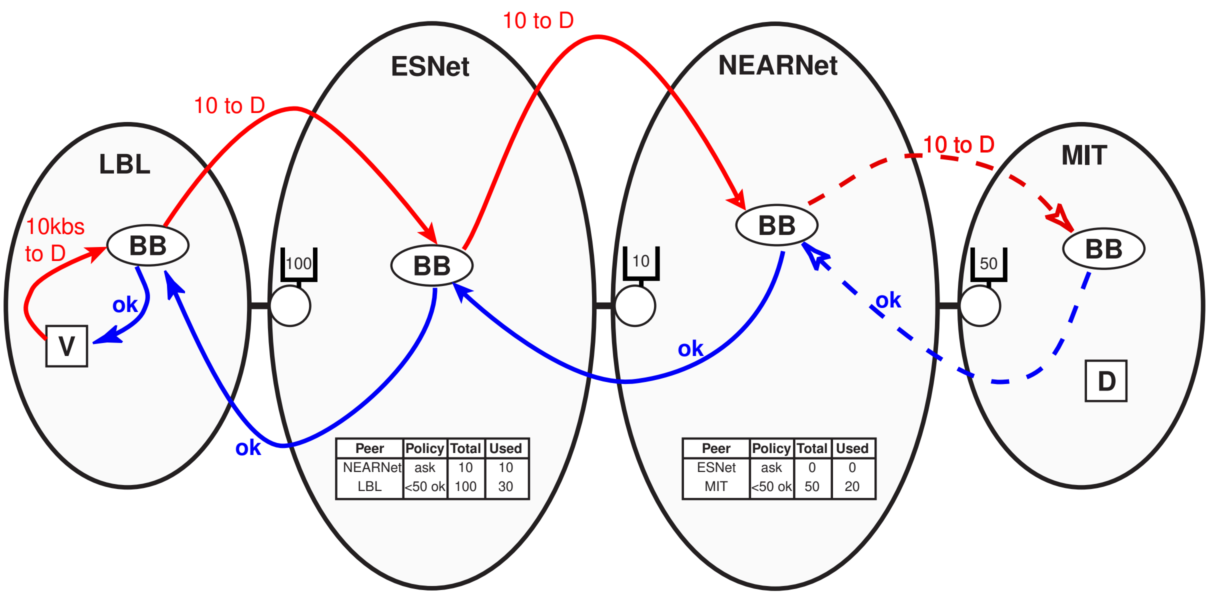
Figure 6. Bandwidth Broker setting Profiles in leaf routers



A statically configured example with BB messages exchanged. Next we present an example where all allocations are statically preallocated but BB messages are exchanged for greater flexibility. Figure 7 shows an end-to-end example for Marked traffic in a statically allocated internet. The numbers at the trust region boundaries indicate the total statically allocated Marked packet rates that will be accepted across those boundaries. For example, 100kbps of Marked traffic can be sent from LBL to ESNet; a Profile Meter at the ESNet egress boundary would have a token bucket set to rate 100kbps. (There MAY be a shaper set at LBL's egress to ensure that the Marked traffic conforms to the aggregate Profile.) The tables inside the transit network "bubbles" show their policy databases and reflect the values after the transaction is complete. In Figure 7, V wants to transmit a flow from LBL to D at MIT at 10 Kbps. As in figure 6, a request for this profile is made of LBL's BB. LBL's BB authenticates the request and checks to see if there is 10kbps left in its Marked allocation going in that direction. There is, so the LBL-BB passes a message to the ESNet-BB saying that it would like to use 10kbps of its Marked allocation for this flow. ESNet authenticates the message, checks its database and sees that it has a 10kbps Marked allocation to NEARNet (the next region in that direction) that is being unused. The policy is that

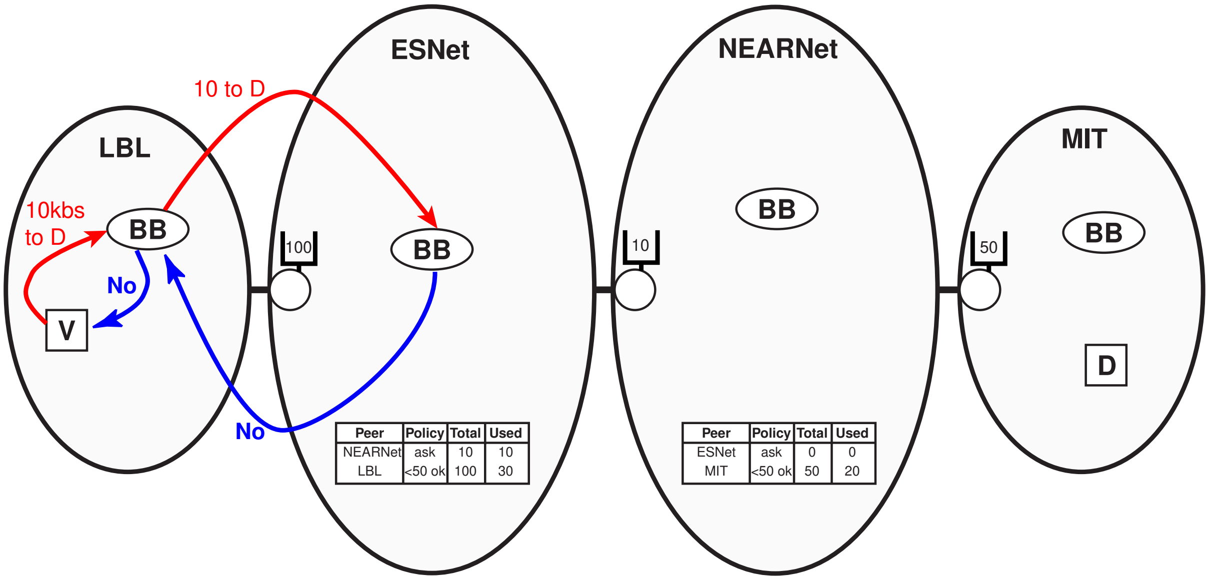
ESNet-BB must always inform ("ask") NEARNet-BB when it is about to use part of its allocation. NEARNET-BB authenticates the message, checks its database and discovers that 20kbps of the allocation to MIT is unused and the policy at that boundary is to not inform MIT when part of the allocation is about to be used ("<50 ok" where the total allocation is 50). The dotted lines indicate the "implied" transaction, that is the transaction that would have happened if the policy hadn’t said "don’t ask me". Now each BB can pass an "ok" message to this request across its boundary. This allows V to send to D, but not vice versa. It would also be possible for the request to originate from D.

Figure 7. End-to-end example with static allocation.



Consider the same example where the ESNet-BB finds all of its Marked allocation to NEARNet, 10 kbps, in use. With static allocations, ESNet must transmit a "no" to this request back to the LBL-BB. Presumably, the LBL-BB would record this information to complain to ESNet about the overbooking at the end of the month! One solution to this sort of "busy signal" is for ESNet to get better at anticipating its customers needs or require long advance bookings for every flow, but it’s also possible for bandwidth brokerage decisions to become dynamic.

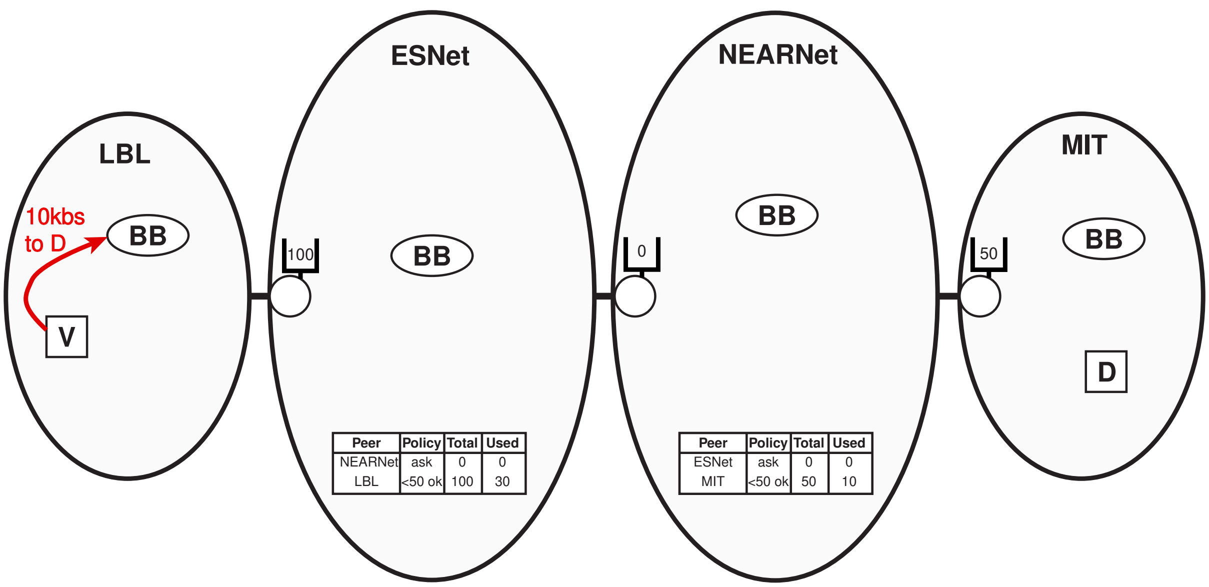
Figure 8. End-to-end static allocation example with no remaining allocation



Dynamic Allocation and additional mechanism. As we shall see, dynamic allocation requires more complex BBs as well as more complex border policing, including the necessity to keep more state. However, it enables an important service with a small increase in state.

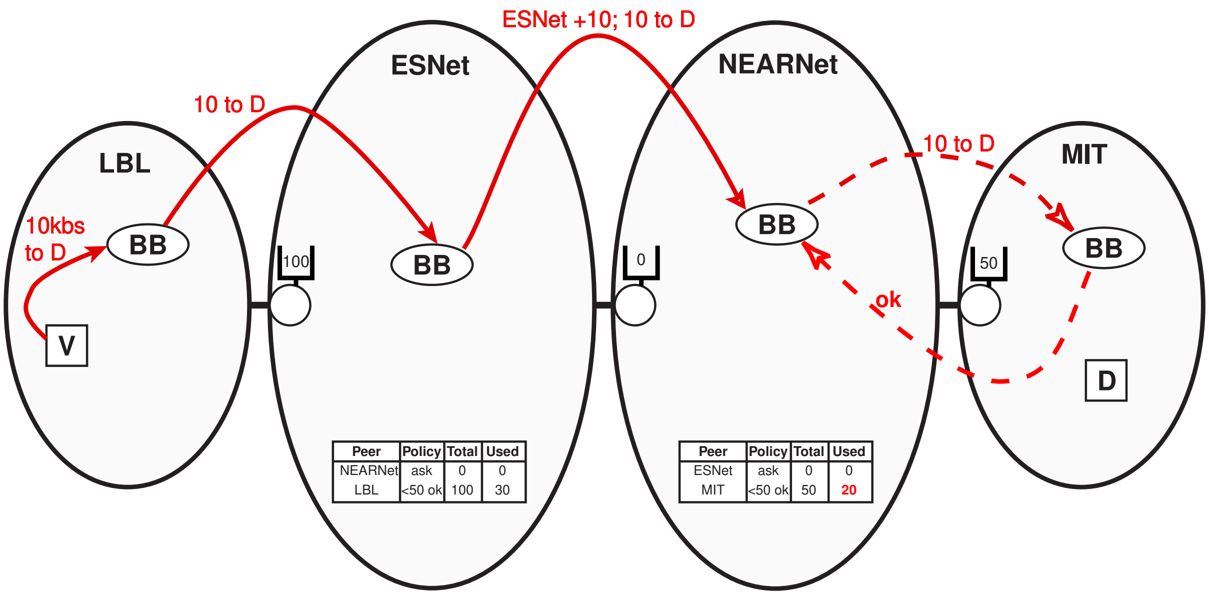
The next set of figures (starting with figure 9) show what happens in the case of dynamic allocation. As before, V requests 10kbps to talk to D at MIT. Since the allocation is dynamic, the border policers do not have a preset value, instead being set to reflect the current peak value of Marked traffic permitted to cross that boundary. The request is sent to the LBL-BB.

Figure 9. First step in end-to-end dynamic allocation example.



In figure 10, note that ESNet has no allocation set up to NEARNet. This system is capable of dynamic allocations in addition to static, so it asks NEARNet if it can "add 10" to its allocation from ESNet. As in the figure 7 example, MIT's policy is set to "don't ask" for this case, so the dotted lines represent "implicit transactions" where no messages were exchanged. However, NEARNet does update its table to indicate that it is now using 20kbps of the Marked allocation to MIT.

Figure 10. Second step in end-to-end dynamic allocation example



In figure 11, we see the third step where MIT's "virtual ok" allows the NEARNet-BB to tell its border router to increase the Marked allocation across the ESNet-NEARNet boundary by 10 kbps.

Figure 11. Third step in end-to-end dynamic allocation example

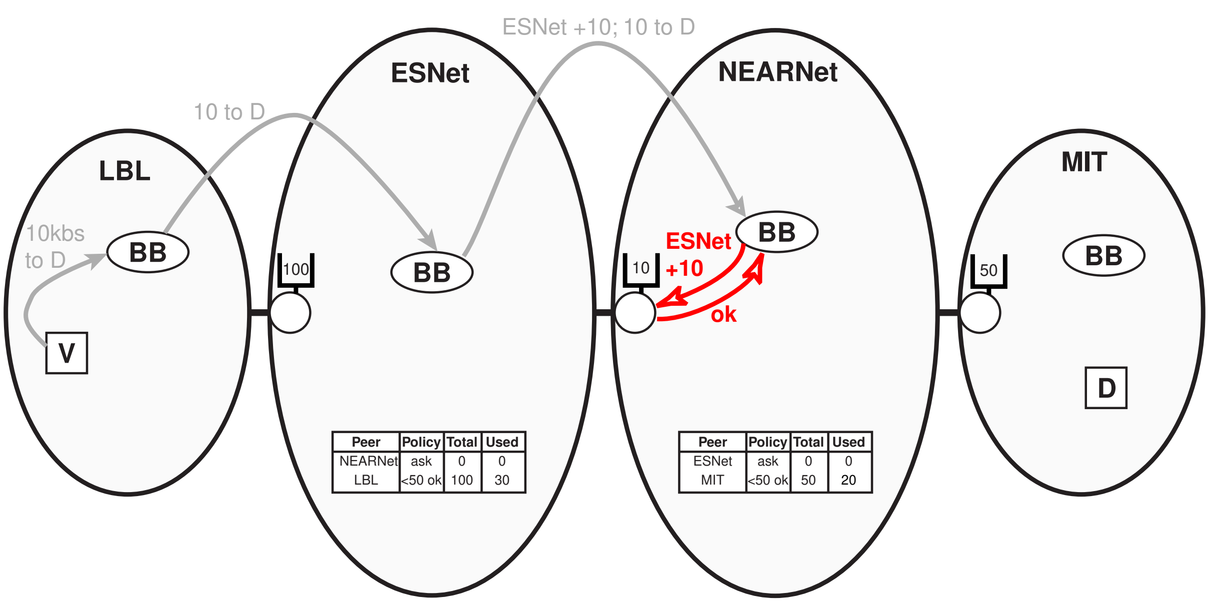
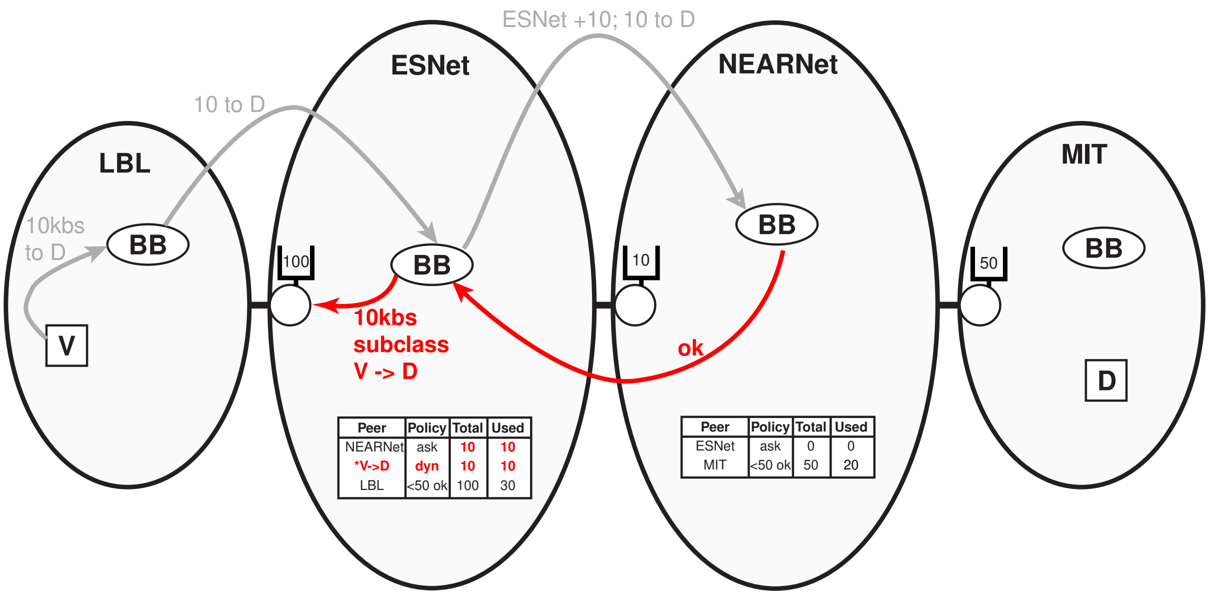


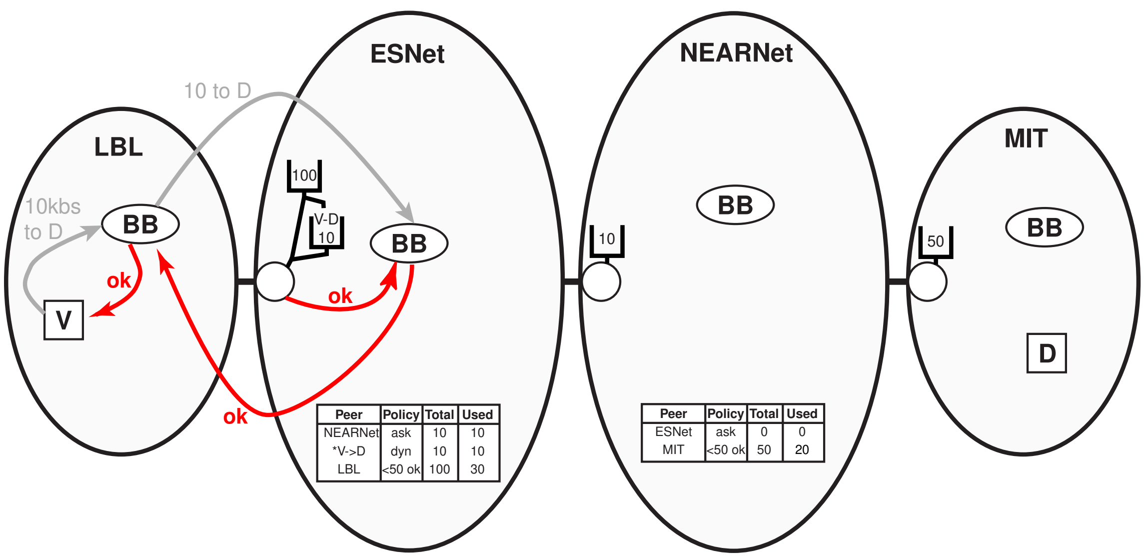
Figure 11 shows NEARNet-BB's "ok" for that request transmitted back to ESNet-BB. This causes ESNet-BB to send its border router a message to create a 10 kbps subclass for the flow "V->D". This is required in order to ensure that the 10kpbs that has just been dynamically allocated gets used only for that connection. Note that this does require that the per flow state be passed from LBL-BB to ESNet-BB, but this is the only boundary that needs that level of flow information and this further classification will only need to be done at that one boundary router and only on packets coming from LBL. Thus dynamic allocation requires more complex Profile Metering than that shown in figure 5.

Figure 12. Fourth step in end-to-end dynamic allocation example.



In figure 12, the ESNet border router gives the "ok" that a subclass has been created, causing the ESNet-BB to send an "ok" to the LBL-BB which lets V know the request has been approved.

Figure 13. Final step in end-to-end dynamic allocation example



For dynamic allocation, a basic version of a CBQ scheduler [5] would have all the required functionality to set up the subclasses. RSVP currently provides a way to move the TSpec for the flow.

For multicast flows, we assume that packets that are bound for at least one egress can be carried through a domain at that level of service to all egress points. If a particular multicast branch has been subscribed to at best-effort when upstream branches are Marked, it will have its bit settings cleared before it crosses the boundary. The information required for this flow identification is used to augment the existing state that is already kept on this flow because it is a multicast flow. We note that we are already "catching" this flow, but now we must potentially clear the bit-pattern.

5. RSVP/int-serv and this architecture

Much work has been done in recent years on the definition of related integrated services for the internet and the specification of the RSVP signalling protocol. The two-bit architecture proposed in this work can easily interoperate with those specifications. In this section we first discuss how the forwarding mechanisms described in section 3 can be used to support integrated services. Second, we discuss how RSVP could interoperate with the administrative structure of the BBs to provide better scaling.

5.1 Providing Controlled-Load and Guaranteed Service

We believe that the forwarding path mechanisms described in section 3 are general enough that they can also be used to provide the Controlled-Load service [8] and a version of the Guaranteed Quality of Service [9], as developed by the int-serv WG. First note that Premium service can be thought of as a constrained case of Controlled-Load service where the burst size is limited to one packet and where non-conforming packets are dropped. A network element that has implemented the mechanisms to support premium service can easily support the more general controlled-load service by making one or more minor parameter adjustments, e.g. by lifting the constraint on the token bucket size, or configuring the Premium service rate with the peak traffic rate parameter in the Controlled-Load specification, and by changing the policing action on out-of-profile packets from dropping to sending the packets to the Best-effort queue.

It is also possible to implement Guaranteed Quality of Service using the mechanisms of Premium service. From RFC 2212 [9]: "The definition of guaranteed service relies on the result that the fluid delay of a flow obeying a token bucket (r, b) and being served by a line with bandwidth R is bounded by b/R as long as R is no less than r. Guaranteed service with a service rate R, where now R is a share of bandwidth rather than the bandwidth of a dedicated line approximates this behavior." The service model of Premium clearly fits this model. RFC 2212 states that "Non-conforming datagrams SHOULD be treated as best-effort datagrams." Thus, a policing Profile Meter that drops non-conforming datagrams would be acceptable, but it’s also possible to change the action for non-compliant packets from a drop to sending to the best-effort queue.

5.2 RSVP and BBs

In this section we discuss how RSVP signaling can be used in conjunction with the BBs described in section 4 to deliver a more scalable end-to-end resource set up for Integrated Services. First we note that the BB architecture has three major differences with the original RSVP resource set up model:

1. There exist apriori bilateral business relations between BBs of adjacent trust regions before one can set up end-to-end resource allocation; real-time signaling is used only to activate/confirm the availability of pre-negotiated Marked bandwidth, and to dynamically readjust the allocation amount when necessary. We note that this real-time signaling across domains is not required, but depends on the nature of the bilateral agreement (e.g., the agreement might state "I’ll tell you whenever I’m going to use some of my allocation" or not).

2. A few bits in the packet header, i.e. the P-bit and A-bit, are used to mark the service class of each packet, therefore a full packet classification (by checking all relevant fields in the header) need be done only once at the leaf router; after that packets will be served according to their class bit settings.

3. RSVP resource set up assumes that resources will be reserved hop-by-hop at each router

along the entire end-to-end path.

RSVP messages sent to leaf routers by hosts can be intercepted and sent to the local domain's BB. The BB processes the message and, if the request is approved, forwards a message to the leaf router that sets up appropriate per-flow packet classification. A message should also be sent to the egress border router to add to the aggregate Marked traffic allocation for packet shaping by the Profile Meter on outbound traffic. (It's possible that this is always set to the full allocation.) An RSVP message must be sent across the boundary to adjacent ISP's border router, either from the local domain's border router or from the local domain's BB. If the ISP is also implementing the RSVP with a BB and diff-serv framework, its border router forwards the message to the ISP's local BB. A similar process (to what happened in the first domain) can be carried out in the ISP domain, then an RSVP message gets forwarded to the next ISP along the path. Inside a domain, packets are served solely according to the Marked bits. The local BB knows exactly how much Premium traffic is permitted to enter at each border router and from which border router packets exit.

6. Recommendations

This document has presented a reference architecture for differentiated services. Several variations can be envisioned, particularly for early and partial deployments, but we do not enumerate all of these variations here. There has been a great market demand for differentiated services lately. As one of the many efforts to meet that demand this draft sketches out the framework of a flexible architecture for offering differential services, and in particular defines a simple set of packet forwarding path mechanisms to support two basic types of differential services. Although there remain a number of issues and parameters that need further exploration and refinement, we believe it is both possible and feasible at this time to start deployment of differentiated services incrementally. First, given that the basic mechanisms required in the packet forwarding path are clearly understood, both Assured and Premium services can be implemented today with manually configured BBs and static resource allocation. Initially we recommend conservative choices on the amount of Marked traffic that is admitted into the network. Second, we plan to continue the effort started with this draft and the experimental work of the authors to define and deploy increasingly sophisticated BBs. We hope to turn the experience gained from in-progress trial implementations on ESNet and CAIRN into future proposals to the IETF.

Future revisions of this draft will present the receiver-based and multicast flow allocations in detail. After this step is finished, we believe the basic picture of an scalable, robust, secure resource management and allocation system will be completed. In this draft we described how the proposed architecture supports two services that seem to us to provide at least a good starting point for trial deployment of differentiated services. Our main intent is to define an architecture with three services, Premium, Assured, and Best effort, that can be determined by specific bit-patterns, but not to preclude additional levels of differentiation within each service. It seems that more experimentation and experience is required before we could standardize more than one level per service class. Our base-level approach says that everyone has to provide "at least" Premium service and Assured service as documented. We feel rather strongly about both 1) that we should not try to define, at this time, something beyond the minimalist two service approach and 2) that the architecture we define must be open-ended so that more levels of differentiation might be standardized in the future. We believe this architecture is completely compatible with approaches that would define more levels of differentiation within a particular service, if the benefits of doing so become well understood.

7. Acknowledgments

The authors have benefited from many discussions, both in person and electronically and wish to particularly thank Dave Clark who has been responsible for the genesis of many of the ideas presented here, though he does not agree with all of the content of this document. We also thank Sally Floyd for comments on an earlier draft. A comment from Jon Crowcroft was partially responsible for our including section 5. Comments from Fred Baker made us try to make it clearer that we are defining two base-level services, irrespective of the bit patterns used to encode them.

8. Security Considerations

There are no security considerations associated with this document.

9. References

[1] D. Clark, "Adding Service Discrimination to the Internet", Proceedings of the 23rd Annual Telecommunications Policy Research Conference (TPRC), Solomons, MD, October 1995.

[2] V. Jacobson, "Differentiated Services Architecture", talk in the Int-Serv WG at the Munich IETF, August, 1997.

[3] D. Clark and J. Wroclawski, "An Approach to Service Allocation in the Internet", Internet Draft draft-clark-diff-svc-alloc-00.txt, July 1997, also talk by D. Clark in the Int-Serv WG at the Munich IETF, August, 1997.

[4] Braden et. al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April, 1997.

[4] Braden R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September, 1997.

[5] S. Floyd and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, pp 365-386, August 1995.

[6] D. Clark, private communication, October 26, 1997

[7] "Advanced QoS Services for the Intelligent Internet", Cisco Systems White Paper, 1997.

[8] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, September 1997.

[9] Shenker, S., Partirdge, C. and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[10] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service", IEEE/ACM Transactions on Networking, August, 1998, Vol6, No 4, pp. 362-373. also at: http://diffserv.lcs.mit.edu/Papers/exp-alloc-ddc-wf.pdf

Authors' Addresses

Kathleen Nichols
Cisco Systems, Inc.
San Jose, CA 95134-1706
Phone: 408-525-4857
Email: kmn@cisco.com

Van Jacobson
Cisco Systems, Inc.
San Jose, CA 95134-1706
Email: van@cisco.com

Lixia Zhang
UCLA
4531G Boelter Hall
Los Angeles, CA 90095
Phone: 310-825-2695
Email: lixia@cs.ucla.edu

Appendix: A Combined Approach to Differential Service in the Internet by David D. Clark

After the draft-nichols-diff-svc-00 was submitted, the co-authors had a discussion with Dave Clark and John Wroclawski which resulted in Clark's using the presentation slot for the draft at the December 1997 IETF Integrated Services Working Group meeting. A reading of the slides shows that it was Clark's proposal on "mechanisms", "services", and "rules" and how to proceed in the standards process that has guided much of the process in the subsequently formed IETF Differentiated Services Working Group. We believe Dave Clark's talk gave us a solid approach for bringing quality of service to the Internet in a manner that is compatible with its strengths.

The slides presented at the December 1997 IETF Integrated Services Working Group follow.

A combined approach to differential service in Internet routers

David D. Clark
M.I.T. Laboratory for Computer Science
IETF
December 1997

The purpose of this talk

Describe mechanisms in routers (and elsewhere) that can form the basis of a range of application services.

(Specifically) describe a combination of two schemes presented at the last IETF -- “Van’s scheme” and “Dave’s scheme”.

Proposal for action -- what do we do now?

Select some suitable mechanisms.

- General
- Implementable

Provide illustrations of how these are used to provide services.

Define how bits are used to select the mechanisms.

DDC MIT Lab for Computer Science 12/97 Slide 3

Vocabulary

What is implemented in a specific “box” (router, profile meter) we will call a mechanism with certain behavior.

What the user obtains, overall, by the specific use of these mechanisms, we will call a service.

In order that a mechanism can be used to build services, there must be enforced rules that limit how it is used.

DDC MIT Lab for Computer Science 12/97 Slide 4

The three (important) places where mechanism is implemented

The router (the point of congestion)

- Mechanism controls scheduling during congestion, etc.

Boundaries between networks

- Mechanism verifies usage patterns, tags packets, shapes flows, logs usage, etc. Enforces the rules.

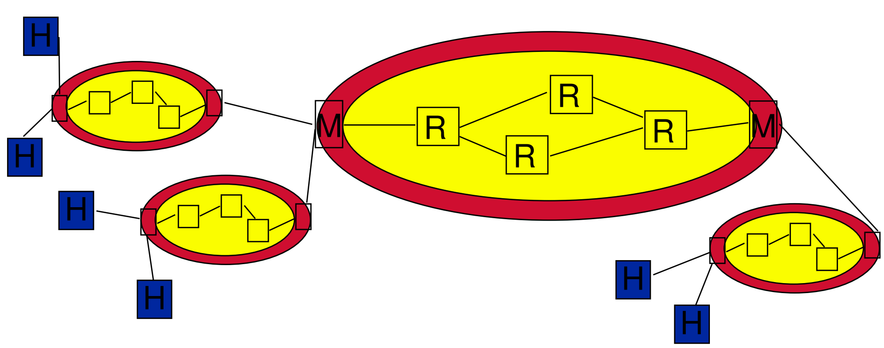
End node -- source and destination

- Generates/sinks traffic
- Uses the mechanisms in ways that realize the service.

These parts, working together, can make a range of services out of the core mechanisms.

DDC MIT Lab for Computer Science 12/97 Slide 5

The placement of mechanism



DDC MIT Lab for Computer Science 12/97 Slide 6

Specifics

Router mechanism	Profile meter mechanism	Resulting service
Queue with higher priority than other queues.	Token buffer with constant rate output.	Emulation of fixed capacity circuit
Queue with RIO dropper.	Tags packets *in* or *out* based on usage profile.	Allocated capacity profiles
	Host tagging of coded video (for example).	Application-controlled drop preference

DDC MIT Lab for Computer Science 12/97 Slide 7

A note on traffic meters

Meters enforce the rules.

Different rules at different points

- Between two ISPs -- enforce bulk contract for access to “mechanism”.

Meters exploit the mechanism to obtain the desired service.

- Applies to meter near (at) the end-point.
- Meters inside the network do not need to know what the application-visible service is.

DDC MIT Lab for Computer Science 12/97 Slide 8

Conclusion for mechanisms

These two mechanisms (priority and RIO dropping) are not in conflict. Both can be implemented at once.

There seems to be a demand for the services that can be built from both mechanisms.

So -- put in both, and allow both sorts of services.

I named a specific dropping mechanism -- RIO dropping.

Claim: if a mechanism is to be used with a range of meters to produce a range of services, the mechanism itself needs to be carefully defined.

DDC MIT Lab for Computer Science 12/97 Slide 9

How to use bits in header of packet

Option 1: bits describe service.

Option 2: bits select/control mechanism .

My proposal: use bits to select/control mechanism. Why?

- Many more services than mechanisms. Run out of bits sooner if select services.
- New services invented all the time. Don't want to standardize them.
- Don't need/want to know what service is in "middle" of net if mechanisms work properly.

The "aggregate and forward" model, rather than the "segregate and bill" model.

DDC MIT Lab for Computer Science 12/97 Slide 10

Drawbacks -- selecting mechanism using header bits

A service might be built from different mechanism in different parts of the net.

Some services may be so universal that it is helpful to have a name for them.

When hoarding bits, it is better to be thoughtful than dogmatic.

DDC MIT Lab for Computer Science 12/97 Slide 11

Using bits to select mechanism.

A framework:

- Two parts:
  - Mechanism selection.
  - Per-packet control over what mechanism does.
- Mechanism:
  - Priority Queue
  - Queue with RIO dropper
- Per-packet control:
  - Priority -- nothing
  - Drop preference -- is <u>this</u> packet *in* or *out*

DDC MIT Lab for Computer Science 12/97 Slide 12

Bits in the header

My proposal

- Use a TOS value to select the mechanism..
- Use one or more precedence bits to select the per-packet treatment within the mechanism.

DDC MIT Lab for Computer Science 12/97 Slide 13

How many drop levels?

I proposed two drop preferences: *in* and *out*.

- Don’t specify more unless it is proven that less will not do.

Why more?

- In military nets, where capacity may change wildly due to network attack, it may be important to shed load in several steps.
- For commercial nets, which are normally provisioned in a conservative way, no clear evidence that having more levels provides an increased range of interesting services.

Multiple drop levels, combined with **specific** rules for use, can create specialized services. See the Kaveli proposal.

DDC MIT Lab for Computer Science 12/97 Slide 14

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

ERROR: undefined
OFFENDING COMMAND: .

STACK: