A TWO-TIER RESOURCE MANAGEMENT MODEL FOR THE INTERNET

A. Terzis, L. Wang, J. Ogawa, L. Zhang
terzis,lanw,ogawa,lixia@cs.ucla.edu
UCLA Computer Science Department

Abstract

In this paper we propose a Two-Tier resource management model for the global Internet. Our solution resembles the current two-tier routing hierarchy and allows individual administrative domains to independently make their own decisions on strategies and protocols to use for internal resource management and QoS support. The aggregate traffic crossing domain borders is served according to relatively stable, long-lived bilateral agreements. End-to-end QoS support is achieved through the concatenation of such bilateral agreements.

We describe in detail a realization of this Two-Tier model, where a Bandwidth Broker (BB) acts as the resource manager for each administrative domain. Neighboring Bandwidth Brokers communicate with each other to establish Inter-domain resource agreements. As an illustrative example in this paper we used a simplified RSVP as an intra-domain resource allocation protocol for the aggregate traffic between border routers. Our simulation results show that this Two-Tier design can provide effective end-to-end QoS support for user applications.

1 Introduction

As the Internet evolves to become the ubiquitous communications infrastructure, there is a clear need for providing differentiated classes of service to network traffic, so that various applications and specific business requirements can be met with assurance in Quality of Service (QoS).

Providing QoS support over the Internet has been a research and engineering challenge for many years. Achieving QoS in a small, controlled environment seems simple: if adequate amount of bandwidth either is provisioned or otherwise can be reserved along the path of a specific data flow, all the packets of that flow can be delivered with minimal delay and no congestion loss. To *assure* such high performance over the global Internet, however, imposes a great challenge. Although one observes good performance from time to time when the network is lightly loaded, long delays and heavy packet losses are common when the network gets congested.

Fundamentally, differentiation of network services requires three simple steps:

1. Defining packet treatment classes,
2. Allocating adequate resource to each class at each router, and
3. Sorting packets to their corresponding classes and controlling the volume to be within the allocated amount.

The *Differentiated Services* architecture [BBC+98] has emerged over the last couple of years to address these three points in a scalable way. This architecture contains two main components. The first component includes the fairly well-understood behavior in the forwarding path (corresponding to points 1 and 3 above), which is moving quickly through the Internet standardization process. The second component, corresponding to point 2 in the above, involves more challenging issues regarding resource allocation policies and procedures to configure parameters used in the forwarding path; it largely remains as an open research topic.

This paper presents a framework that addresses the issues of resource allocation and management. We call this framework the *Two-Tier* resource management model. Our model is based on the fact that today's Internet is made of the interconnection of multiple administrative domains. Following the approach taken in the Internet routing architecture, we separate resource allocation control into a two level hierarchy, inter-domain allocation and intra-domain allocation. This separation allows each administrative domain to individually make its own decision on strategies and protocols to use for internal QoS support.

In our design, neighboring administrative domains make relatively stable, long-term *bilateral* agreements on the allocation of resources to different traffic classes for the aggregate traffic crossing domain borders. End-to-end QoS support is achieved through the concatenation of such bilateral agreements which install adequate Inter- and Intra-domain resource allocations. This is analogous to providing global IP delivery by hop-by-hop packet forwarding through intra and inter-domain routing protocols. Different from routing, however, we face the new challenge of assuring that the resource allocations at the two levels match each other, and that the allocations match the aggregated traffic demand.

The rest of the paper is structured as follows. Section 1.1 gives a brief overview of the Differentiated Services architecture. Section 2 presents the design and rationale behind the Two-Tier resource management architecture. A realization of our design for inter-domain resource allocation is described in section 3, and an example intra-domain management in section 4. To show a proof of concept, Section 5 presents some preliminary simulation results. Section 6 compares our work to other related proposals and finally we close with a summary in Section 7.

1.1 Differentiated Services

The Differentiated Services effort in IETF has developed a simple model to differentiate packet delivery qualities. The model assumes that each packet carries an appropriate value in the DS field value (previously called TOS byte) in its IP header. Each DS field value corresponds to a different forwarding treatment, called a Per Hop Behavior (PHB). Within the core of the network, routers sort incoming packets to different forwarding classes according to their DS values. For example, if the value a packet carries translates to an "Expedite Forwarding" treatment, routers will sort that packet into a class that has guaranteed bandwidth allocation. Since core routers only need to exam the DS field to decide how to service a packet, no complex classification or per-flow state is needed, leading to a simple implementation with increased scalability.

To ensure that network resources allocated to each forwarding class are not over subscribed, traffic entering the network is classified, and possibly shaped and policed. The traffic volume at network entry points is lower than that in the core, allowing packet classification and control with finer granularity and complexity to meet various policy requirements.

Routers in the Differentiated Services architecture are grouped by administrative boundaries to form Differentiated Services domains, for example an organization's Intranet or an ISP makes a DS domain. At domain boundaries, Agreements are made regarding the amount of resources allocated for data flows that cross domains. These agreements are called Service Level Agreements (SLAs), which represent business agreements between domains and are long lived.

The Differentiated Services approach differs from previous efforts in QoS provision in that it separates the mechanisms used to provide packet level service differentiation (such as queuing or buffering disciplines), collectively called forwarding path mechanisms, from the admission control and resource allocation mechanisms belonging in the management plane. This separation resembles the approach in the original Internet design that separates packet forwarding module from the routing module. In contrast, architectures such as the Integrated Services and ATM tie these two components together in an attempt to assure a flow's delivery quality at the connection establishment time.

2 Design Overview

In order to better understand our design, we first describe the current routing architecture in the Internet. We then present the rationale behind our design, followed by a description of the main components of our scheme and the interaction between them.

2.1 Routing in the Internet

The Internet today is made of the interconnection of multiple autonomous networks called autonomous systems (AS), or administrative domains, each under a separate administrative control. Each domain contracts its neighboring domain(s) for data delivery service; the neighbor domain, in turn, may pass the traffic to next neighbors, so on and so forth until packets are delivered to final destinations. For example, a campus contracts one ISP (or a few for redundancy) to deliver its traffic; the ISP delivers the campus' traffic either directly if the destinations are connected to the same ISP, or otherwise passes the packets to other ISPs for further forwarding.

Reflecting the AS-based network connectivity, today's Internet routing architecture follows a two-level hierarchical design. Each of the Autonomous Systems is free to choose whatever routing protocol it deems proper to use. To assure global connectivity, neighbor domains use the Inter-domain routing protocol BGP [RL95] to exchange network reachability information; reachability information can be aggregated when nearby networks share common address prefixes.

We would like to make a few observations here. First and foremost, to get data delivered to external destinations, each domain makes a bilateral agreement with its directly connected neighbor domains, rather than multi-lateral agreement involving each of all ISPs along the paths to all possible destinations. The concatenation of bilateral traffic delivery agreements through transit ISPs results in the global IP delivery service.

Secondly, each individual domain makes simple delivery commitments externally, while it retains freedom in choosing its own routing approach internally. One may choose a preferred IGP (Interior Gateway Protocol) from multiple candidates, such as OSPF, RIP, or even manual router table configuration if the local domain is small enough to manage. One's choice of IGP does not impact routing function between domains. By keeping inter-domain and intra-domain routing independent, the system allows routing to be independently administered by the various Autonomous Systems.

Lastly, forwarding entries to all destinations are pre-computed, based on routing protocol message processing, rather than being computed in real time upon packet arrival. Furthermore, the pre-computed routing database is also dynamically adjusted to account for changes in topology or policy. This separation of routing computation and packet forwarding allows network service to be quickly deployed while the routing protocol continuing to evolve, and allows routing adjustments to be made on much larger time scales independent from individual flows' duration.

2.2 Two-Tier Resource Management

The tenet of our design is what we call *Two-Tier* resource management. By this term we mean that resource allocation should be done in two levels. The first level is resource allocation inside each administrative domain while the second level is resource management across neighboring domains. Following the paradigm of Internet Routing, each domain is free to choose whatever mechanism it deems proper for internal resource management as long as its bilateral resource agreements with neighboring domains are met.

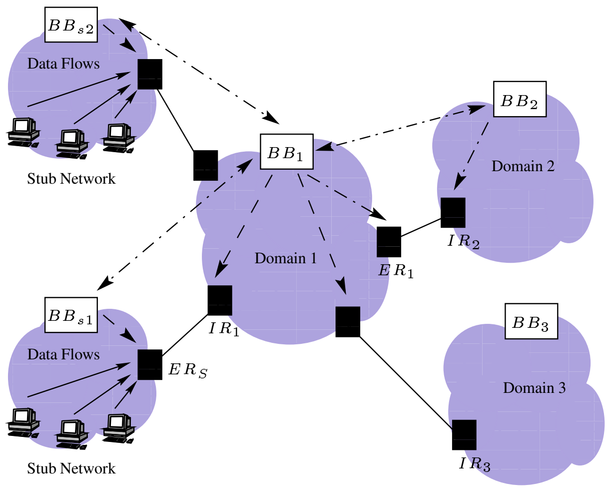

Figure 1: Two-Tier Resource Management

While Intra-domain resource allocation can be fined grained (per flow), we require that Inter-domain resource agreements are made for the *aggregate* traffic crossing domains. Furthermore, Inter-domain agreements should change infrequently at a larger time-scale than that of individual applications. These two requirements on Inter-domain agreements provide substantial scaling characteristics by decoupling Inter-domain allocations from individual end-to-end flows.

Figure 1 shows how resource allocation information is distributed in the Two-Tier model. Leaf Domains contact their service providers to request certain about of resources to cover for the aggregate high quality traffic leaving the domain. Once the agreement is in place, individual applications can request and use portions of the aggregate allocated amount. When and if the allocated resources are exhausted, the leaf domain may be able to re-negotiate the agreement with its provider, allocating a larger amount of resources.

Note that in our design the leaf domain contacts only its immediate neighbor for all its traffic, although the traffic may head toward various final destinations far away. It is the responsibility of the downstream domain, after agreeing to carry the client traffic, to both allocate resources internally as well as request allocation from the downstream neighbors for the portions of the traffic that exit the domain.

The challenges introduced by the desire to make resource allocations for aggregate traffic are twofold:

- The domain that receives traffic has to *predict* to where traffic flows and to make appropriate allocations both internally and externally.

- In the event of a failure, or when sufficient resources are not available to serve the total amount of traffic, the affected leaf domains should be notified. Given that those domains do not make explicit requests for traffic going to specific destinations it becomes harder to notify those domains.

In Sections 3 and 4 that follow, we elaborate on the details of Inter- and Intra-domain resource allocation respectively.

3 Inter-domain Resource Allocation

As we argued in Section 2, end-to-end QoS is provided by the concatenation of Intra-domain resource allocation and bilateral resource agreements between neighboring domains. These agreements specify the amount of traffic belonging to different classes, that crosses links connecting adjacent domains. To ensure that the level of actual traffic is always lower than the negotiated limit, the receiving domain *polices* incoming traffic, dropping or demoting[1] excess traffic. Knowing that offending traffic will be policed, the sending domain in turn, *shapes* traffic so that it always remains in profile.

In what follows, we explain how Inter-domain agreements are established and maintained and introduce the concept of the *Bandwidth Broker*, the entity responsible for administering Inter-domain agreements.

3.1 Bandwidth Broker

The idea of a Bandwidth Broker (BB for short) was first introduced by Van Jacobson et al in [NJZ97] as the *logical* entity in charge of resource management in an administrative domain. Being the locus of control for a domain's resource management, the Bandwidth Broker has a dual role:

- **Manage the domain's internal resources.** The BB can be responsible for resource allocation itself or it can delegate resource allocation to an *internal resource management protocol* (we describe one in Section 4) and be responsible only for special events and policy decisions (e.g. the admission of a new flow).
- **Allocate Inter-domain resources.** Each BB maintains bilateral agreements with its neighboring Bandwidth Brokers to allocate resources for the aggregate amount of traffic crossing domains.

If Inter-domain resource agreements were on a per application-flow basis, the amount of state that would have to be kept by border routers and Bandwidth Brokers would increase linearly with the number of flows crossing domains. Moreover agreements between domains would have to change very frequently to accommodate for arrivals and departures of individual flows.

Such a behavior would seriously affect the scalability and stability of the Inter-domain mechanism and of the resource allocation model in general. We therefore require that resource allocations between domains are for the *total aggregate* amount of Inter-domain traffic belonging to each service class. For simplicity in this paper we talk about only one service class other than Best Effort, namely the *Expedited Forwarding*[JNP99] (EF) class, but the mechanisms provided here can be used with multiple service classes.

Since Bandwidth Brokers are responsible for resource agreements and resource agreements are associated with monetary cost it becomes important to protect Bandwidth Broker communications from malicious attacks. IPsec [KA98] can be used to provide authentication and confidentiality to messages exchanged between Bandwidth Brokers.

[1] demoting is the process of changing the DS codepoint of packets to some value that requires *lower* service, such as best effort

A related issue is how Bandwidth Brokers discover securely neighboring Bandwidth Brokers and border routers belonging to their domain. While manual configuration is a first step towards this direction, if the set of border routers and neighboring Bandwidth Brokers is large and varying, some discovery mechanism will be required.

For reasons similar to those mentioned about security, robustness is equally important to the operation of Bandwidth Brokers. As we mentioned before, the Bandwidth Broker is a logical entity that can map to a single or multiple *physical* entities. If the Bandwidth Broker is materialized by multiple physical entities then robustness is increased (one can imagine a system with one primary and multiple backup systems implemented BB functionality). This increased robustness though introduces the problem of *consistency* between the multiple physical entities playing the role of the Bandwidth Broker.

An important question related to relationship between neighboring Bandwidth Brokers, is that of *state (and fate) sharing*. Depending on the granularity of resource allocation and negotiation time scales, the amount of state information shared between two BBs may vary. For the sake of robust, fault tolerant operation, we believe that any sharing of state between BBs should be based on the *soft state* model, so that necessary state can be re-established and recovered quickly when a BB recovers from a crash or a BB is replaced by another one as part of fault recovery. Therefore, we stipulate that any interaction among BBs that requires establishment of shared state must involve periodic timeout and refresh of shared state for robust operation.

3.2 Inter-domain Protocol

We assume that initial resource agreements among neighboring domains are bootstrapped via some configuration configuration. For example, network managers, based on past observed network usage, can install some initial resource agreements among neighboring domains. After this initial phase is over, Inter-domain agreements, can either remain static, be changed manually by network operators or adjust dynamically by an automated procedure. We believe that initially Inter-domain agreements will be static or seldomly change, but as experience with Differentiated Services mounts, agreements will become more dynamic, adjusting to changing traffic demands. In the rest of this section we describe a mechanism that dynamically adjusts the level of Inter-domain allocated resources according to the aggregate level of traffic crossing domain boundaries.

Figure 2 shows the message exchanges between the different entities involved in Inter-domain resource allocation. In this figure there are two neighboring domains A and B. BB_1, which is the Bandwidth Broker of domain A, is re-

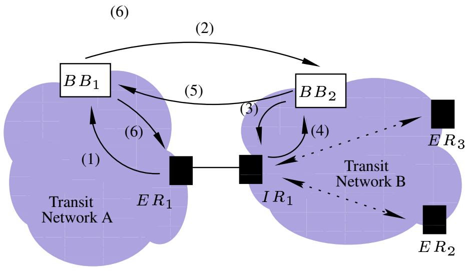

Figure 2: InterDomain message exchange

sponsible for allocating resources for the aggregate traffic crossing domain A to domain B. Let's assume that an agreement of level L (for simplicity we use L to be bandwidth) pre-exists between domain A and B. BB_1 informs egress router ER_1 about this limit, which in turns tunes its shaper parameters so traffic exiting the domain conforms to L.

When ER_1 detects[2] that the traffic volume r exiting the domain, exceeds a *High Watermark* value of $w \times L$ ($w \leq 1$ e.g. $w = 0.8$), it notifies BB_1 by sending it a message containing the current rate r and the address of IR_1. BB_1 in turn sends a message to its downstream neighbor BB_2 requesting $I \times r$ amount of resources ($I > 1$).

When BB_2 receives this increase request, it forwards a request to IR_1 querying whether there are sufficient internal resources to service this increase in entering traffic. The mechanism used by IR_1 to make this decision is a matter of the Intra-domain resource management mechanism which we describe in Section 4.3. If enough internal resources are available, IR_1 replies positively to BB_2, at the same time adjusting the parameters of it's policer for incoming traffic from domain A based on the contents of the request. When BB_2 receives the positive reply from IR_1 it sends a reply to BB_1 accepting the requested increase in traffic. As a last step then BB_1 informs ER_1 to update its L value to $L' = I \times r$.

Parameters w and I deserve a little more discussion. The *High Watermark* value w, provides the Inter-domain mechanism a *cushion* that protects domain-crossing traffic during the re-negotiation interval. The lower the value of w, the higher the cushion which in turn means that larger spikes in network traffic can be absorbed and longer delays in the re-negotiation process can be tolerated, without any loss in traffic performance.

The existence of cushions not only protects the system against sudden surges in traffic but also reduces the number of cascading changes in resource allocation through the domain structure. If domains delayed increasing resource allocations up to the point where real traffic reached the allocated levels, then resource allocation changes would *ripple* through the chain of domains carrying the traffic as each domain would try to adjust its inter-domain allocation. On the other hand, if domains applies a different high watermark then adjustments in one point may not lead to further adjustments downstream.

The multiplicative increase factor I, is used by the BB to dampen the frequency of increase requests. When the value of I increases, the new level L' of allocated resources increases also, thereby making the increase in inter-domain traffic necessary to trigger another renegotiation, even bigger.

There is a simple relation that links the two parameters w and I. Since we require the new level of allocated resources L' to be larger than L we have:

$$L' > L \Rightarrow I \times r > L \Rightarrow I \times w \times L > L \Rightarrow I \times w > 1 \quad (1)$$

Using Equation 1, we can compute the value of one of the parameters, given the value of the other.

Given that Inter-domain resources are going to be associated with some sort of cost, the amount of Inter-domain resources allocated should not exceed the current needs by a large margin. Therefore, a mechanism is required to detect the existence of *considerable* and *persistent* gaps between the levels of allocated and actual resources and then to reduce the amount of allocated resources.

The mechanism we designed works as follows: when the current load r that ER_1 measures on the link to the neighboring domain becomes, $r \leq l \times L, (l < 1)$, ER_1 notifies BB_1 about this condition. BB_1 applies a *hysteresis* process to the decrease requests from ER_1. BB_1 keeps a hysteresis counter H for each egress router. Each time BB_1 receives a decrease request, it decreases H. When the value of H becomes zero, it sends a request to the downstream Bandwidth Broker BB_2 to decrease the amount of Inter-domain resources to $L' = D \times L, D \leq 1$ and instructs ER_1 to adjust its shaper to the new value L'. When BB_2 receives a decrease request it also instructs IR_1 to adjust the parameters of its policer to L'. Hysteresis is applied so that Inter-domain allocations are decreased only when the level of traffic is consistently lower than the level of allocated resources. The hysteresis interval H regulates the frequency of Inter-domain resource requests. Large values of H will result in larger intervals where r must be lower than L and therefore will lead to less frequent changes in Inter-domain allocations.

The multiplicative decrease constant D regulates, how conservative or aggressive is the Bandwidth Broker in trying to *match* the Inter-domain resource allocation to the current

[2] we describe in Section 3.3 the mechanism that ER_1 uses to measure traffic exiting the domain

load. If D is close to one then the Bandwidth Broker is conservative, making only gradual decreases in the amount of allocated resources, while if D is lower then the Bandwidth tries to closely match the Inter-domain resources allocated to the current load. The reader may notice that while allocation increases are a function of the current traffic level, decreases are not. The reason for this choice is that we want to be cautious decreasing only gradually from the current level of allocation. If we had taken the level of actual traffic in consideration, a steep decrease in traffic followed by a sudden increase would lead to two inter-domain adjustments, while with the current scheme the second adjustment (the increase) is possibly avoided.

Messages between all the nodes involved in the Inter-domain mechanism are delivered reliably. This requirement however, is not in contrast with our earlier suggestion that a *soft state* protocol should be used for the BB communications. Indeed, soft state protocols such as RSVP are enhanced with acknowledgment mechanisms that can provide reliable delivery of messages without sacrificing the soft-state character of the protocol [WTZ99].

3.3 Estimation Process

The purpose of the estimation process is to measure the network load attributed to EF packets. This estimate, which we refer to as r, is computed by a domain's edge routers and is used for two purposes: (1) to estimate the aggregate amount of EF traffic that follows the path from one ingress router to one egress router and (2) to estimate the amount of EF traffic crossing domains.

Our traffic estimation model uses a time window measurement process borrowed from [JDSZ97], [JB97]. The time window measurement process uses two parameters, T and S. T is the measurement window and S is the sampling period, with T being a multiple of S. During every sampling period, S, an average load is computed. This average load is simply the sum of bits in EF packets received by the length of the sampling period divided by the length of the sampling period. The load estimate, r, is updated as follows:

1. If a newly computed average load for a given sampling period S is larger than the current value of r, r is set to the newly computed average.

2. At the end of every measurement window, T, r is set to the highest average load computed for any S during the previous window.

Increasing T increases the amount of history remembered by the measurement process. For a fixed T, decreasing S makes this measurement process more sensitive to bursts of data. Appropriate values of S are likely to be on the order of thousands of packet transmission times.

3.4 Reject Behavior

While up to now we have described the behavior of the Inter-domain protocol when adequate amount of resources exist, there are going to be cases when increase requests must be rejected either because no physical resources are available or for policy reasons. Our goals in this situation are the following:

- Convey the reject information to upstream neighbors affected by this shortcoming and eventually to leaf domains. It is up to each individual domain how to react to this information.

- Protect traffic from leaf domains that do not contribute to this anomaly and therefore should not be affected by it.

- Allow the Bandwidth Broker of the domain where the failure occurred to decide what upstream domains should be notified and affected according to some local policy.

With these goals in mind, we present, using Figure 3, the propagation of reject information to upstream domains from the point of failure.

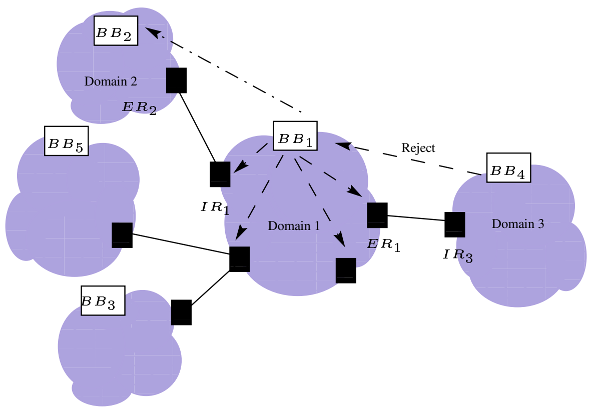

Figure 3: Propagation of Reject Information

The process begins when Bandwidth Broker BB_1 receives a negative response for one of it's requests to downstream Domain 3. BB_1 inquires all the domain's border routers about which ones are sending traffic towards that particular egress. As we will see in Section 4, border routers keep this information for internal resource management.

Each border router that forwards data towards the problematic egress point ER_1 reports back to the domain's Bandwidth Broker including in it's response the measured amount of data towards that egress.

Once BB_1 receives the list of ingress routers, it makes a policy decision regarding which of the upstream domains will be affected from the current situation. This policy decision depends on the type of SLAs that each upstream domain has with Domain 1. For example, an SLA may specify that traffic may be dropped as long as the customer is reimbursed for the lost traffic. Then a domain having such an SLA will be selected to face the problem of (transient) over-allocation rather than a domain which has an SLA requiring traffic to be carried at all times (such an SLA would of course would be costlier to establish).

In our example, IR_1 is the only ingress router sending to ER_1. So BB_1 informs BB_2 about the problem downstream. BB_1 includes in its message to BB_2 the δ increase in resource request that caused the rejection. BB_2 can use the information provided by BB_1 to inform its own affected upstream domains using the same mechanism. Eventually this information propagates all the way back to the affected leaf domains. Each leaf domain then individually makes a policy decision about how to react to this problem. Depending on the Intra-domain mechanism and the amount of flow information every domain keeps, it can decide to selectively terminate some application flows, to shape the total aggregate to the new value at its border or to demote some of its traffic at the domain border.

4 Intra-domain Resource Allocation

The purpose of Intra-domain resource allocation mechanisms is to check whether sufficient network resources are available for traffic flowing through each network and if so to allocate domain resources for this traffic. Each domain is responsible for allocating resources internally using any mechanism that seems reasonable, as long as Inter-domain agreements with neighboring domains are met.

There are two variations in the Inter-domain resource allocation mechanism. The first deals with allocation at *leaf* domains, that is domains which contain sources or receivers of traffic, while the second is used in *transit* domains that carry traffic from its source to its final destination. The fundamental difference between the two versions of the internal allocation mechanism is that in leaf domains flow information is available to aid in resource allocation decisions, while in transit domains no such information is available. Transit domains, therefore have to discover the directions traffic is headed through the domain and allocate resources accordingly.

In the sections that follow we present sample realizations of the leaf and transit domain internal resource allocation mechanisms. Both mechanisms are based on RSVP, but we do not advocate RSVP as the *only* solution to this problem. We expect in the future other mechanisms will arise; for now RSVP presents a solution which is available today and can be used to build a working system.

4.1 Allocation in Leaf Domains

As we have seen earlier, leaf domains contract their providers to carry their traffic downstream with appropriate QoS provisions. However, for applications to receive the full benefits of the network QoS, the complete network path from the source host to the destination host has to provide the requested service.

Providing QoS to the portion of the path not covered by the Inter-domain agreement, that is from the source host to the egress of the source domain and from the ingress of the destination domain to the destination host is the task of Intra-domain resource allocation in leaf domains. Implementing QoS in leaf domains is not only a technical issue but also an issue of policy, but fundamentally such a task involves three issues:

1. Communicating the QoS requirements between the sending and receiving application(s).

2. Getting authorization and allocating adequate resources at the source domain and

3. Getting authorization and allocating adequate resources at the destination domain.

The simplest way to allocate resources inside a leaf domain, is through static configurations. Network managers at leaf domains statically allocate adequate resources for applications that require higher levels of service. The level of required resources is estimated or derived empirically.

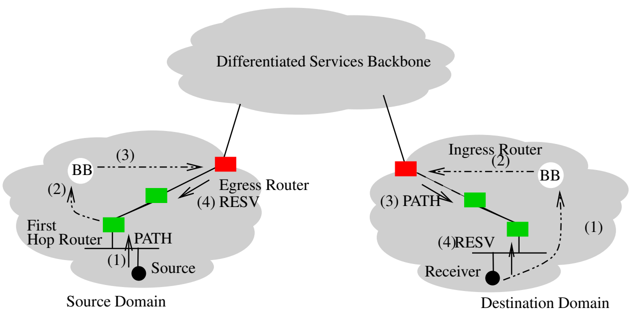

Figure 4: Intra-domain Resource Management

An extension to this mechanism is to replace the static configuration with automated communications between the applications (or some proxy) and the BB of each domain. Figure 4 shows how this scheme works. At the source domain, a source starts sending RSVP PATH messages towards the flow's receiver. The first-hop router intercepts the PATH message and informs the domain's BB of the source host's request. The BB then checks if the source is allowed to send the requested amount of traffic according to the domain's policy and whether there are enough resources allocated on the border link to the downstream domain that the traffic will be sent to. If the source is not permitted to use the resources then an error message notifies the sender of the failure. In the case where all Inter-domain resources are exhausted, the source request may trigger a resource renegotiation with the downstream domain. Otherwise, if both checks are successful, the BB informs the ingress router to forward the PATH message as well as the egress router that is the domain's exit point towards that specific destination, to send RESV messages to the back to source host.

Next, the egress router starts sending RESV messages, reserving resources on the local path towards the source. If the domain does not have enough internal resources to support the flow, a reservation error is returned to the egress which in turn notifies the source. The RESV message delivered to the source host also contains the mapping (inside a DCLASS [Ber99] object) between the Intserv service contained in the PATH message and the DS codepoint that should be used for packets sent. PATH messages sent by the source are not forwarded beyond the leaf domain but instead are terminated at the source domain's egress router.

Since PATH messages from the source are not delivered to the destination domain, receivers need an out-of-band mechanism to know the traffic profile of the source. This can be achieved by advertising the source's traffic profile at the application level. Once the receiver knows this information it sends a request to the destination domain's BB requesting the specified resources. The BB examines the receiver's eligibility to receive the requested traffic and then checks if it has contracted adequate receiving capacity on its border link. If any one of the tests fails, BB informs the receiver of the failure. Otherwise, BB instructs the ingress router to send PATH messages towards the receiver on behalf of the source. After receiving the PATH message, the receiver sends RESV messages towards the source which allocate resources on the local path, thereby completing the coordinated process of allocating resources at the source and destination domains.

4.2 Allocation in Transit Domains

Given that upstream domains do not provide any information about the destinations or the current level of injected traffic, domains that have agreed to carry this traffic have to discover where traffic is headed, check resource availability and allocate resources on the paths from the point where traffic enters up to the point where traffic exits the transit network.

In our proposed solution, each ingress router measures the amount of traffic sent towards each egress router and uses RSVP to inquire about resource availability and allocate resources on the domain paths towards each egress router. Border routers have an enhanced forwarding table, where for each known destination not only the next hop is listed, but also a counter is kept. This counter counts the number of bytes contained in EF packets[3] sent towards that specific destination. Each time that a packet arrives at an ingress router, the routers looks up the packet's destination address in its forwarding table to properly forward the packet towards its next hop. Additionally, for EF packets the ingress router increases the counter associated with that destination.

Once the amount of traffic sent towards each destination in the forwarding table is measured, the next step is to map destinations to egress routers. For this task, we assume that each of the border routers in the transit network participates in the BGP routing protocol [RL95]. Each BGP router in an Autonomous System, advertises the destinations learned by exterior peers to all others BGP routers in that AS. Using this information, an ingress router can map each destination to the egress router used to reach it.

Having this information, ingress routers periodically execute the following algorithm to compute the amount of traffic sent towards each egress router:

```
for (k=0;k < num_of_BRs;k++) {
  egress_router = BR[k];
  counter[k] = 0;
  for (i=0;i < num_of_destinations;i++) {
    if(egress(dest[i]) == egress_router) {
             counter[k] += dest[i].EF_counter;
    }
  }
}
```

`num_of_destinations` is the number of destinations in the router's forwarding table, `num_of_BRs` is the number of the domain's border routers while the table `BR[...]` holds the border routers of the domain. The function `egress()` gives the border router towards a destination (by looking at the BGP routing table). The table `dest[i]` contains the i-th destination in the forwarding table, while `counter[k]` holds the counter values for egress router k.

[3] alternatively the counter may measure number of packets

Having the per-egress counters, ingress routers can apply the procedure shown in Section 3.3, to compute the r_I values, of traffic sent towards egress router I. Using these values, each ingress router sends RSVP PATH messages towards each egress router it has seen traffic for ($r_I > 0$). Egress routers respond by sending RESV messages, reserving resources inside the transit domain. In the event that a reservation is not successful due to transient over-allocation, a reservation error message is returned to the egress router that issued the reservation request. The egress router in turn, notifies the domain's BB initiating the mechanism described in Section 3.4.

From our description of the mechanism above, one can see that for a domain that has N border routers, the number of RSVP sessions carried by the domains' routers is in the worst case $N(N-1)/2 \approx N^2$. While such scaling behavior may prove burdensome to domains with a large number of border routers there are two comments to make: (1) this mechanism presents a significant improvement over per-flow state, since all flows between a ingress-egress pair are aggregated to a single flow, (2) this mechanism presents only a early solution to the problem of Intra-domain allocation in transit networks and is provided only as an example of how this task could be accomplished and not as a final solution.

Indeed, we are currently investigating mechanisms where the state kept at a domain's interior routers, is independent from the number of flows crossing the domain or the number of border routers.

4.3 Coupling of Inter- and Intra-domain Protocols

As we saw in Section 3.2, when a BB receives a request for increasing the amount of traffic entering its domain, it queries the Intra-domain mechanism to check whether the increase in incoming traffic can be supported. Via this mechanism, Inter-domain commitments are transformed to demands for internal resources and therefore the effectiveness of Intra-domain resource management scheme becomes crucial to the establishment and maintenance of Inter-domain agreements.

Following Figure 2, when BB_2 sends the increase query to IR_1, this ingress router has to check whether this request can be accepted. Assuming that the increase request is for δ more units of bandwidth, then IR_1 distributes this load on the paths towards the domain's egress routers in proportion to the amount of traffic currently sent towards each egress router. That is, if $\alpha_1, \ldots, \alpha_k, (\alpha_1 + \ldots + \alpha_k = 1,\ \alpha_i \leq 1)$, are the current percentages of total traffic sent towards egress routers $ER_1, \ldots, ER_k$ respectively, then IR_1 increases its request towards egress ER_i by $\alpha_i \times \delta$. Each of the contacted egress routers, will respond to the increase by sending a RESV message for the updated amount. If all RESV requests are successful, then IR_1 replies positively to BB_2 and the request is accepted, otherwise the request is denied.

This method is based on the assumption that newly admitted traffic will follow the same traffic distribution that current traffic has. Such an assumption is of course a heuristic that can fail, but the point is that without any explicit information from the upstream domain regarding the destinations traffic will follow, some heuristic has to be applied. In the event where due to traffic shifts, resources on some part of the internal network are exhausted, the mechanism described in Section 3.4 is invoked.

5 Simulation Results

The purpose of the simulations we performed, is twofold: First we want to show that resource allocation information can be effectively propagated among numerous domains without the need for explicit, per-flow signaling. Second, we want to investigate the levels of service that applications can receive from the Two-Tier architecture in order to compare the utility of our architecture with existing QoS provision architectures.

We have therefore divided our simulations to two major categories: those that investigate how Inter- and Intra-domain allocation are performed and those that examine services received by user applications. After briefly presenting our simulator in the next section, we discuss our results in each category.

5.1 Simulator Description

Our simulator is written in PARSEC [Mey98] and is based on a RSVP simulator [TNW98] we had developed earlier. There are five entities in the simulator: *Senders, Border Routers, Interior Routers, Bandwidth Brokers* and *Receivers*. Senders are the sources of data and receivers are the final destinations. Interior and Border Routers forward packets according to their service class. In addition, Border Routers are responsible for measuring data, sending RSVP messages as well as shaping and policing traffic. Our routers implement the EF PHB using a priority queue. Following the guidelines in [JNP99], we have limited the maximum amount of link capacity that can potentially be used by EF to 50%. Bandwidth Brokers implement the scheme we described in Section 3.

We have implemented both UDP and TCP senders. UDP Senders are ON/OFF sources. The lengths of the ON and OFF periods are selected from a Pareto distribution with parameter β. During the ON period a sender sends packets at a constant configured rate. In our simulations, we have used three different UDP senders profiles similar to those used in [JDSZ97]. Table 1 shows the parameters of the different source types.

Sender Type	Sending Rate	ON Period	OFF Period	β	Packet Size
1	8 KB/s	300 ms	325 ms	1.2	125 bytes
2	8 KB/s	300 ms	3000 ms	1.2	125 bytes
3	64 KB/s	40 ms	360 ms	1.2	125 bytes

Table 1: Parameters of different UDP senders

To emulate the effects of traffic shifts a UDP sender can send to a small number of different receivers. At the end of an ON period the sender randomly chooses among its group of receivers, which the next receiver will be. All TCP senders are FTP sources. A FTP sender starts a TCP session to a particular receiver, sends 500 Kbytes of data and then picks a different sender to send to.

5.2 Inter-domain Results

Our goal here is to show how Inter-domain agreements can adjust over the time depending on the current traffic load and second to show how the tuning of the parameters presented in Section 3.2 can affect the shape of these Inter-domain agreements.

The topology we used is shown in Figure 5. There are 8 domains in total, 6 of them being leaf domains containing sources and destinations of traffic, while 2 of them are transit domains carrying traffic destined to some leaf domain. Links connecting senders or receivers to first hop routers have bandwidth of $1.5Mbps$ and delay of $10ms$. Intra-domain links have capacity of $10Mbps$ and delay of $10ms$ while Inter-domain links have capacity of $4.5Mbps$ and delay of $20ms$. While this topology does not cover the full heterogeneity of the Internet it contains links with different speeds and delays making our topology semi-realistic. In this simulation we used $w = 0.9$, $l = 2$, $I = 1.25$, $D = 0.95$, $S = 1sec$, $T = 100$ and $l = 2$.

Figures 6 and 7 show the traffic crossing the boundary between Border Routers 33 and 36. The *meas* line shows the current load on the link as measured by Border Router 33. The *r* line shows the estimation of the average load as described in Section 3.3. Lastly, the *inter* line shows the level of the Inter-domain agreement as set by the neighboring BBs. For the first figure we used $H = 100$ while for the

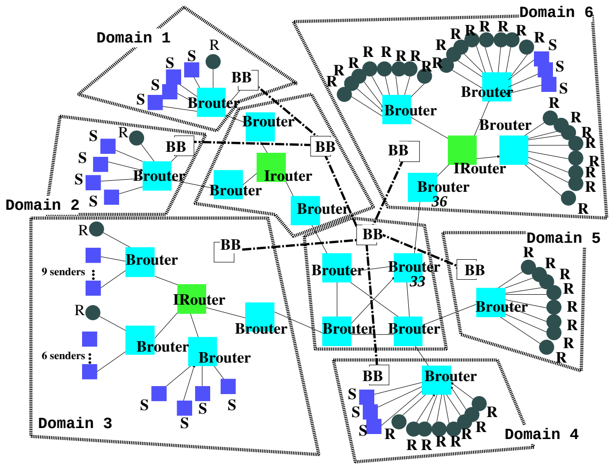

Figure 5: Simulation Topology

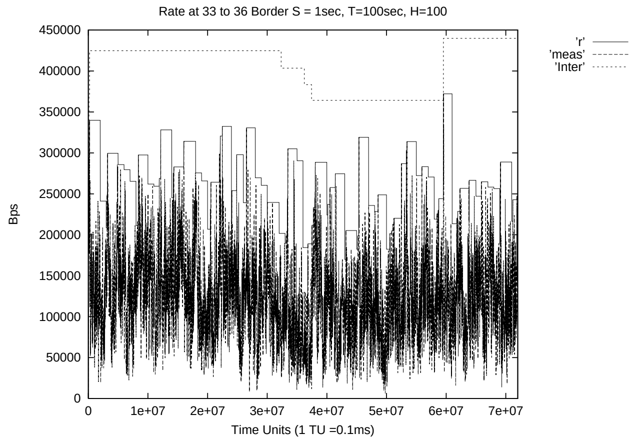

Figure 6: Resource Allocation at the 33-36 border, for H =100

second one we used $H = 500$.

We have two comments to make about these figures. First we see that the mechanisms described in Section 3.2 and 3.3 can effectively measure the amount of aggregate traffic crossing inter-domain links and consistently allocate sufficient resources for this traffic. Second, it is apparent that by increasing the value of H, the Inter-domain agreement changes less frequently. This is the trade-off between stability and closely following the current load we described in Section 3.2.

5.3 End-to-End Performance

This set of simulations was performed to estimate the performance that EF and best effort flows receive. The purpose of these simulations was twofold: (1) to show that EF flows

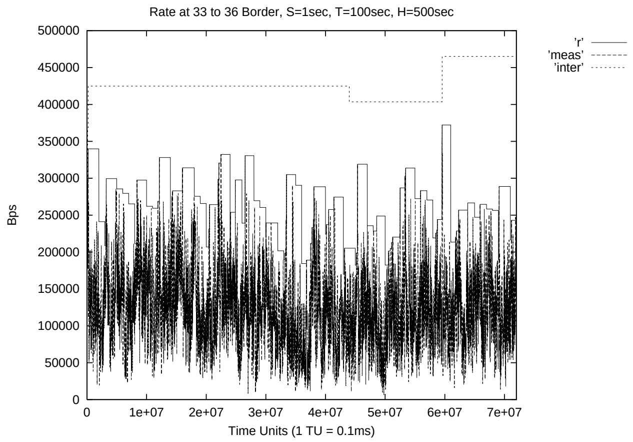

Figure 7: Resource Allocation at the 33-36 border, for H=500

get increased service over best effort flows and (2) that best effort traffic is not completely dominated by EF traffic. We have simulated both TCP and UDP flows and we present the results in the sections that follow.

5.3.1 UDP Performance

Figure 8 shows the topology we used to evaluate the performance of EF UDP traffic compared to best effort traffic. All links have $1Mbps$ capacity and delay of $1ms$. There are 8 domains in total. Senders 0 and 1 are sending to receivers 18 and 19 respectively, while sources from domain 1 and 2 send competing best effort UDP traffic to receivers at domain 3. Sender 0 is an EF sender while sender 1 uses best effort. All of the senders are identical Pareto ON/OFF sources with $\beta = 1.2$, sending packets with size 228 bytes [4] at $40KBps$ during ON periods.

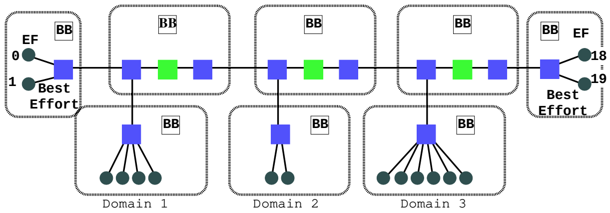

Figure 8: UDP Simulation Topology

Figures 9 and 10 show the delay and jitter distribution of EF and BE traffic respectively. We see that the delay of the EF flow is smaller than that of the best effort flow sharing the same path. The delay of EF traffic is dominated by the shaping done at the egresses of domains, while BE delay shows that BE traffic encounters large queuing delays. Note that the scale on the y axis is logarithmic.

For the jitter computation we used the same definition given in [JNP99] as a way of making our results comparable to those provided in that work. From that graph we see that EF traffic has very low jitter compared to best effort traffic. The reason is that once EF packets are shaped at the first egress router they observe virtually no queuing delay.

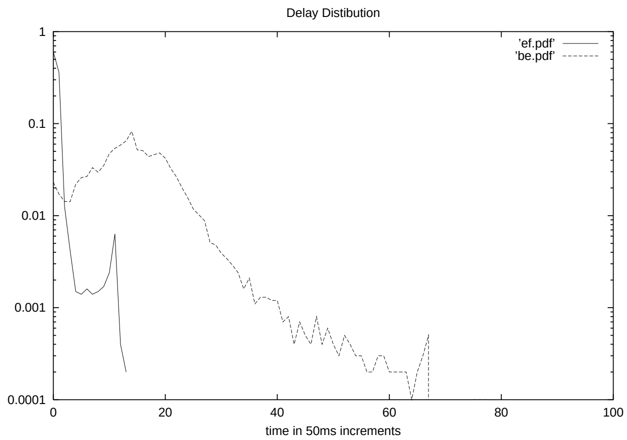

Figure 9: Delay distribution for best effort and EF UDP traffic

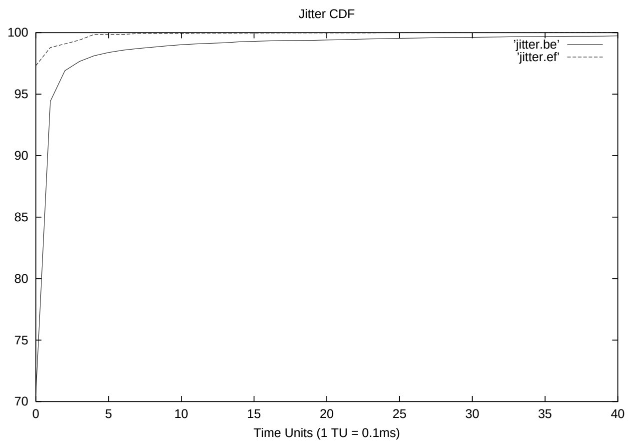

Figure 10: Jitter cumulative distribution for best effort and EF UDP traffic

5.3.2 TCP Performance

For the TCP simulations we have used the same topology shown in Figure 8 but we have replaced senders 0 and 1 with TCP sources sending FTP traffic. We also decreased the sending rate of UDP sources to be $20KBps$ to make

[4] 200 bytes of payload plus 28 bytes for IP and UDP header

them less aggressive. We ran the simulation for $1000ms$ and the results are taken from the later half of the simulation, to give TCP connections time to stabilize. Our simulation results show that the average throughput of the EF TCP connection is $426Kbps$, which is close to the bandwidth allocated to EF traffic, while the best effort TCP connection only gets $0.028Kbps$. We know from the characteristics of UDP sources that the maximum traffic rate of the UDP sources competing with the best-effort TCP connection is $960Kbps$, when all of them are in their ON-period at the same time. It is apparent that the EF TCP connection fully utilizes its share of the link, while the best-effort TCP traffic is dramatically affected by the best effort UDP traffic.

6 Related Work

A number of recent research efforts have addressed the issue of resource allocations for traffic aggregates. Guerin et al in [GBH97], proposed a mechanism where individual RSVP sessions are *hidden* over the backbone, instead RSVP sessions for the aggregate traffic are created at the backbone edges on their behalf. This work also covers the issues of aggregate scheduling requirements, admission control and path characterization to support both Guaranteed and Controlled Load Services. In [BV98], Berson et al proposed a similar scheme, though their work mostly focuses on the signaling aspect of aggregation. The main difference between these proposals and our work is that our work proposes a new resource management framework, namely the Two-Tier framework. This framework eliminates the constraint that all domains adopt the same reservation protocol, and replaces *multi-lateral* agreements with strict *bilateral* agreements between neighboring domains for the aggregate traffic crossing domain borders.

Recently, in the context of the Differentiated Services architecture, the concept of *Dynamic Packet State* or DPS has been proposed [SZ99]. In this work Stoica et al present a intra-domain signaling protocol that requires only aggregate traffic class state at internal routers. However, all border routers, including those of backbone domains, must keep per flow state; furthermore, the scheme relies on end-to-end, per flow signaling through transit domains. On other hand, our framework removes dependency on end-to-end Signaling, it supports end-to-end QOS by addressing the fundamental issue of *predicting* the resource needs of traffic aggregates through transit domains towards various destinations.

In the context of the MPLS architecture, Li et al have proposed in [LR98] the PASTE mechanism where aggregate reservations are are made between edges of an MPLS domain. Reservations are made between pairs of ingress and egress routers or on a sink tree towards each egress by using a modified version of RSVP. PASTE fits well into our Two-Tier resource management model as an alternative intra-domain resource allocation protocol.

Finally, the scheme we presented in Section 4.1 for resource allocation within leaf domains differs from a related proposal presented in [BYF+99] in a fundamental way. Our scheme makes no assumption about the mechanism used by the peering leaf domain, while [BYF+99] assumes that both leaf domains support RSVP; RSVP signals from one domain travel all the way to the other domain (even though those signals are *hidden* in the core are not used for resource allocation). This assumption is in sharp contrast with our design principle of the Two-Tier model, that is individual administrative domains be given the freedom of choosing its internal resource allocation mechanisms.

7 Summary

In this paper we described a Two-Tier resource management model for the global Internet. Our design resembles the current two-tier routing hierarchy and divide resource allocation control into a two level hierarchy, Inter-domain allocation and Intra-domain allocation. This division allows each administrative domain to individually make its own decision on strategies and protocols to use for internal QoS support. This division between intra- and inter-domain resource control also naturally leads to treating the aggregate traffic that cross domain borders as the basic unit for inter-domain resource allocation, a resource management design that scales with the size of the topology instead of the number of end-to-end application flows, and that matches well with the Differentiated Service architecture.

We pointed out two of the basic challenges in making reservations for aggregate traffic: the destinations of the aggregate data flow are not known in advance, and the sources of the flow are not explicitly given to deliver feedback information regarding resource condition changes. We use a traffic measurement based approach to handle both issues. Each administrative domain measures the amount of traffic entering and exiting the domain and adjusts it's Inter-domain allocations accordingly. Similarly, for internal resource allocation the amount of traffic traveling on each ingress-egress pair is measured, and appropriate resources are allocated for it.

Our preliminary simulation results showed that the concatenation of bilateral service agreements across domains, supported by a collection of simple building blocks including traffic shapers, policers, and a simple priority scheduler developed by Differentiated Services effort, can indeed provide effective end-to-end quality of service support for demanding users applications.

Acknowledgments

We like to thank Francis Reichmeyer, Lyndon Ong and Raj Yavatkar, our co-authors in the Internet-Draft [ROT+98] containing an early description of the Two-Tier resource management model. We also would like to thank the anonymous reviewers for their insightful comments that helped us improve the quality of this paper.

References

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *RFC 2475*, Dec 1998.

[Ber99] Y. Bernet. Usage and Format of the DCLASS Object With RSVP Signaling. *Internet-Draft, work in progress*, 1999.

[BV98] S. Berson and S. Vincent. Aggregation of Internet Integrated Service States. In *Proceedings of IWQoS 98*, May 1998.

[BYF+99] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, John Wroclawski, and E. Felstaine. A Framework for Use of RSVP with Diffserv Networks. *Internet-Draft, work in progress*, Sep 1999.

[GBH97] R. Guerin, S. Blake, and S. Herzog. Aggregating RSVP-based QoS Requests. *Internet-Draft, work in progress*, November 1997.

[JB97] S. Jamin and L. Breslau. A Measurement-based Admission Control Algorithms for Controlled-load Service. *Internet Draft, work in progress*, Oct 1997.

[JDSZ97] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks,. *IEEE/ACM Transactions in Networking*, Feb 1997.

[JNP99] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. *RFC 2598*, June 1999.

[KA98] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. *RFC 2401*, Nov 1998.

[LR98] T. Li and Y. Rekhter. Provider Architecture for Differentiated Services and Traffic Engineering (PASTE). *Internet-Draft, work in progress*, January 1998.

[Mey98] R. Meyer. PARSEC User Manual. Available at http://pcl.cs.ucla.edu/projects/parsec/manual, August 1998.

[NJZ97] K. Nichols, V. Jacobson, and L. Zhang. A Two-bit Differentiated Services Architercture for the Internet. *Internet-Draft, work in progress*, Nov 1997.

[RL95] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). *RFC1771*, March 1995.

[ROT+98] F. Reichmeyr, L. Ong, A. Terzis, L. Zhang, and R. Yavatkar. A Two-Tier Resource Management Model for Differentiated Services Networks. *Internet-Draft, work in progress*, 1998.

[SZ99] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per Flow Management. In *Proceeding of SIGCOMM'99*, 1999.

[TNW98] A. Terzis, C. Nikoludakis, and L. Wang. Simulation of the Resource ReSerVation Protocol (RSVP) in PARSEC. Available at http://irl.cs.ucla.edu/, Feb 1998.

[WTZ99] L. Wang, A. Terzis, and L. Zhang. A New Proposal for RSVP Refreshes . In *In Proceeding of ICNP 99*, Nov 1999.