A Distributed Traffic Control Scheme Based on Edge-centric Resource Management

Yingxin Jiang and Aaron Striegel
Dept. of Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46556 USA
yjiang3,striegel@nd.edu

ABSTRACT

The correct admission of flows in the Differentiated Services (DiffServ) environment is critical to provide stable and predictable quality of service (QoS) to the end user. Without a scalable and precise admission control scheme, the service provider is faced with either over-provisioning the network or accepting periods of best-effort like behavior. In this paper, we propose a novel approach for admission control that exploits the unique architectural aspects of DiffServ. Through the use of periodic heartbeats emanating from edge routers to probe the network state on the available egress paths, edge routers are able to quickly conduct admission control with a tunable degree of precision. In this paper, we detail our approach, Edge-centric Resource Management (ERM), and conduct detailed simulation studies regarding the effectiveness of the approach.

Categories and Subject Descriptors: C.2.3 [Network Operations]: Network monitoring

General Terms: Algorithms, Measurement

Keywords: Differentiated Services, QoS, Admission Control, Traffic Engineering

1. INTRODUCTION

Over the last decade, the concept of quality of service (QoS) has evolved dramatically. Starting from the unsuccessful origins in the type of service byte (ToS) in the initial IPv4 specification, the networking research community has explored a wide variety of techniques ranging from per-flow QoS in Integrated Services [17] to aggregate or per-class QoS in the Differentiated Services model [16]. A critical aspect of each of the QoS schemes beyond best effort is the notion of *traffic control*, controlling what is admitted (admission control) and verifying the traffic stays within its allocation (policing).

The dominant factors for an admission control scheme are that the scheme should be fast (low setup latency), scalable (large network, large number of flows), and precise (efficient and controllable resource allocation). Hence, a wide variety of proposals have emerged to address the issues of resource management (and implicitly admission control) in the context of both inter-domain and intra-domain traffic. Efforts in the inter-domain space have built on RSVP [9], the emerging work by the IETF NSIS (Next Step In Signaling) workgroup, and other protocols [18]. Specifically, this paper is interested in addressing the issue of intra-domain admission control within the context of one of the more promising QoS models, Differentiated Services (DiffServ).

While there has been a wide variety of efforts with regards to DiffServ QoS admission control [1, 4–7, 12–14], the majority of the efforts can be categorized into one of two categories, namely *distributed* approaches relying on probing and *centralized* approaches relying on the concept of a Bandwidth Broker (BB). In the distributed approach, nodes probe the network and make decisions for admission based on the QoS of the probe packets. Nodes act largely independently of one another relying on observed behavior rather than explicit reservation to assess the free capacity of the network. Thus, one must typically over-provision the network so as to allow appropriate safeguard regarding the imprecision of the probe data.

In contrast, centralized approaches rely on a Bandwidth Broker (BB) whereby all new connections must be approved through the BB. Provided that all policing is done correctly, no probing is necessary as the BB provides a centralized location for the domain-wise resources. Although the centralized approach can offer more precise resource allocation, it suffers from scalability constraints even when considered only on a domain-wise basis. Furthermore, both approaches can suffer from latency penalties due to either the probe/response delay or the edge router/BB delay. While such an impact may be minimal across a single domain or AS, the compound effect of multiple domains can be significant.

1.1 Motivation

Thus, a service provider is faced with a dilemma. Should the provider emphasize scalability and employ a probing approach or should the provider embrace a centralized approach for additional efficiency at the cost of scalability? We believe that a new approach that offers the scalability of probing with the efficiency of centralization would offer an extremely compelling approach to admission control. In this paper, we propose a distributed scheme, *ERM (Edge-based Resource Management)*, that offers a new approach for DiffServ intra-domain admission control.

Rather than edge routers operating independently, we introduce the notion of cooperative state gathering heartbeats that provide a relatively precise view of the network. ERM builds on the notion of a heartbeat with the idea of *effective remaining fair capacity* to govern the overall rate of admission control and keep the network core stable. Through simulation studies, we demonstrate how our scheme provides an intuitive and effective approach for admission control that meets the three primary goals of speed, scalability, and precision under a variety of network conditions.

The remainder of our paper is organized as follows. Section 2 comments on related work and differentiates the novelty of our work. Section 3 presents an overview of ERM and discusses the network intelligence (heartbeat) aspects of ERM. Section 4 continues by discussing the admission control aspects of ERM and presenting advanced issues regarding ERM. Next, Section 5 details our simulation studies on ERM and Section 6 concludes the paper with several remarks.

2. BACKGROUND AND RELATED WORK

To begin, we briefly introduce the key concepts of the Differentiated Services framework. In the DiffServ framework, routers are divided into two key types: simple, stateless *core* routers and intelligent, stateful *edge* routers. Edge routers are responsible for policing, shaping, and/or marking packets with a DSCP (DiffServ Codepoint) that determines the PHB (Per-Hop Behavior) that a packet will receive in the core. Packets enter at the *ingress* router and exit at the *egress* router. Policing, shaping, and/or marking are performed at the ingress router using a SLA (Service Level Agreement), thus providing a sender-driven QoS.

In the Bandwidth Broker (BB) architecture, when a new flow with the specific QoS requirement and traffic profile arrives at the ingress router, the ingress router will forward a request packet to the bandwidth broker to ask for admission to the network. If the QoS requirement of this flow can be satisfied, the bandwidth broker will acknowledge the ingress router with a positive feedback; otherwise, a negative feedback will be sent. While probing is not necessary due to precise resource allocation, this centralized scheme will inherently suffer from scalability and robustness issues. Hence, the alternative is to employ a distributed admission control scheme.

The most common distributed approach is a Probe Based Admission Control (PBAC) which employs end-to-end unicast probes via a pattern of packets to infer the path-wise QoS. PBAC schemes can be further subdivided into schemes that require core router involvement [1, 4, 7] and Endpoint Admission Control (EAC) schemes [5,6,12–14]. In the PBAC schemes where core routers are involved, each router on the path has to make a decision about whether the probes can continue to be forwarded to its destination. These schemes avoid the storage of per-flow states by using run-time link load estimation to do admission control. However, since the core routers need to actively participate in admission control for every flow going through them, the architectural and computational requirements for core routers can be high. Such requirements contradict the concepts of the DiffServ framework where core routers are simple.

In Endpoint Admission Control (EAC) schemes, before the new traffic flow is admitted into the network, the ingress router sends a probe packet stream with similar characteristics to the request flow to the egress router along the flow's path. According to the feedback(s) coming from the egress router, the ingress router can make the admission decision. In EAC, core routers are simple, since they are only responsible for forwarding probe packets as normal data packets. However, flow set-up delays can often be on the order of seconds [10]. Furthermore, there is a hazard of the thrashing problem: when many flows are probing at the same time, a large amount of link capacity may be consumed by probing packets.

In a classical sense, ERM could be viewed from the perspective of isarithmic traffic control where the distributed heartbeats of ERM operate similar to the permit operations of isarithmic traffic control. In contrast to a permit-based system, the heartbeats of ERM indicate a fair capacity of the links from the ingress versus an indication of permission to send additional packets. Other works related to ERM, although not explicitly used for admission control, include delay/loss probing via multicast [11], multicast heartbeat exchanges for fault tolerance [19], feedback based routing [20], and potential based routing [2]. While the works in [2,19,20] also utilize distributed feedback and control, the works focus primarily on routing rather than admission control.

In contrast to the previous work in DiffServ admission control, our paper makes several key contributions that include:

- *Coordinated heartbeat:* Rather than having each edge router act independently with regards to probing (as with all of the above distributed schemes), edge routers in ERM both cooperatively gather and reflect relevant information to other edge nodes. This cooperation allows ERM to more efficiently gather information and offers its concept of effective fair capacity.
- *Expedited flow/aggregate admission:* ERM utilizes the concept of a distributed effective fair capacity that allows an edge router to make immediate admissions providing it has remaining capacity. Rather than waiting for probing or a BB response, ERM offers a significantly reduced setup time while still remaining scalable.
- *Increased efficiency:* Since state information is continually gathered and transformed into an effective fair capacity to guard against over-allocation with a reasonable degree of precision, ERM is able to increase the maximal safe network utilization versus existing probing schemes.
- *Fixed overhead:*Unlike probe-based schemes where the control overhead is dependent on the number of flows, the overhead of ERM is independent of the number of flows. The overhead of ERM is directly fixed to the number of edge nodes in the network. Moreover, the percentage of overhead in ERM decreases with increasing link speed, making ERM an excellent admission scheme for high speed networks with large numbers of flows.

3. EDGE-BASED RESOURCE MANAGEMENT

The premise of Edge-based Resource Management (ERM) is to provide a controllable and predictable tool for admission control that leverages the unique aspects of the DiffServ architecture. In keeping with the initial principles of DiffServ, our goal is to keep the admission control mechanism both relatively simple and scalable. In short, each edge router in the domain works together to help gather information about the internal state of the network but yet operates independently for admission control (see Figure 1).

A key point to note is that ERM focuses on edge-to-edge QoS (across a single domain), not end-to-end QoS. Thus, we believe ERM can seamlessly interact with the wide body of work in end-to-end QoS negotiation [9,18]. A second aspect of an edge-to-edge focus is a significant reduction in scale

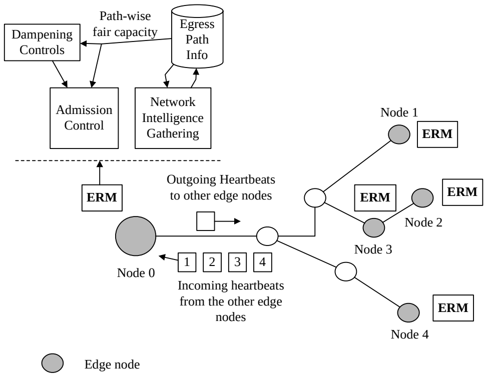


Figure 1: ERM Framework

of the number of edge nodes participating in the ERM process. ERM focuses on the notion of ingress (entrance) and egress (exit) edge nodes based on the DiffServ architecture. Specifically, we are interested in making an admission control decision at the ingress due to the sender-driven nature of DiffServ[1].

The primary concern of ERM is how to juxtapose the competing demands on the network between controllability and efficiency. On one hand, network state gathered extremely quickly and often will provide an extremely accurate view of the network. In such a case, one can safely allocate to maximal efficiency based on the excellent accuracy of the information. However, such rapid techniques will extract a high price on the network itself, thus taking a large toll on the maximal efficiency possible. Furthermore, the network state will always be a lagging factor versus the currently admitted flows, thus introducing a further level of inaccuracy that cannot be avoided when employing such an approach.

The lagging factor is important in that lagging in terms of over-estimating capacity can result in over-allocation and QoS violations, a serious concern. Although lagging in terms of under-estimating capacity will result in missed efficiency, the penalty associated with under-estimating is much less than from over-estimation. The goal of ERM is to avoid over-allocation in all circumstances but yet push the efficiency level as high as safely possible.

In the next subsection, we describe our approach for gathering network intelligence through the use of periodic heartbeat exchanges among edge nodes in the network. We build on the concept of the coordinated heartbeat in the next section by introducing methods for admission control that control the inaccuracy inherently associated with the heartbeat approach but yet achieve a relatively high level of maximal efficiency from the network.

3.1 ERM Heartbeat

The primary component of ERM is a periodic heartbeat message exchanged amongst the edge routers present in the domain. As the heartbeat crosses the domain, it gathers information with regards to QoS similar to what could be gathered using SNMP. By virtue of the heartbeat, edge nodes can gather the current state of the core of the network to make admission control decisions without contacting a central entity. Informally, the heartbeat problem can be stated as follows:

For each ingress router E_I in E where E is the set of all of edge nodes in the domain, gather the QoS information on all egress points accessible from E_I in a timely and scalable manner.

In essence, the system-wide goal of the heartbeats is to gather all of the relevant link-wise information for communications between each of the edge nodes. On the whole, the heartbeats themselves look like a series of many-to-many communications. A natural solution is to employ multicast in order to transport heartbeats across the domain. The use of multiple unicasts (as seen with existing probing approaches) is extremely inefficient on a larger scale and can potentially be extremely taxing on the computational load of the core routers. On the other hand, employing a traditional many-to-many multicast tree (*,G) may not cover the appropriate paths.

Thus, we propose to employ a restricted form of multicast for transport of the multicast heartbeat. First, the multicast group will be strictly limited to the domain, thus negating the need for inter-domain multicast routing or the need for end-user multicast support[2]. Second, multicast will be delivered using the PIM-SSM model [3] with multicast groups sourced at a given edge router and membership restricted to verifiable edge nodes in the domain. Multicast join/leave messages will be driven by out-of-band control signaling such that an edge node will drive other specific edge routers to join or leave a requested SSM group with itself as the source.

3.2 Heartbeat Transport

Before explaining heartbeat transport, we briefly describe the PIM-SSM model. PIM-SSM (Protocol Independent Multicast, Source Specific Multicast) [3] is similar to PIM-SM (Protocol-Independent Multicast, Sparse Mode) except that the multicast transport is limited to one-to-many. PIM-SSM performs Reverse Path Forwarding (RPF) for receiver joining and state refreshes. Thus, after a receiver joins a PIM-SSM tree, the multicast tree path from the source to this receiver will follow the reverse of the minimal cost path from the receiver to the source. The edge-to-edge minimal cost paths are provided by underlying routing protocols (such as OSPF [15]).

In order to more easily introduce the ERM heartbeat concept, we begin with strict assumptions regarding the heartbeat scope/relevance and relax the restrictions in a later subsection. We assume that all edge nodes are *greedy* such that all paths to all other edge nodes are verified on a single heartbeat. Thus, all other edge nodes will be a member of the SSM multicast group centered on node E_I (see Figure 2).

Each edge node in the network will send a heartbeat packet periodically at an interval of HBI (Heartbeat Interval) seconds. When a heartbeat arrives at a node, a node will append its respective QoS information to the packet and forward the packet appropriately on the multicast distribution tree. After a heartbeat arrives at a receiver, the receiver will use the information gathered in the heartbeat to do admission control. From the view of admission con-

[1] While ERM can be easily modified to handle a receiver-driven approach such as RSVP signaling, we focus on the intricacies of sender-driven operations in this paper.

[2] The domain-wise restriction of multicast can be further enforced through the use of secure digital signatures if necessary.

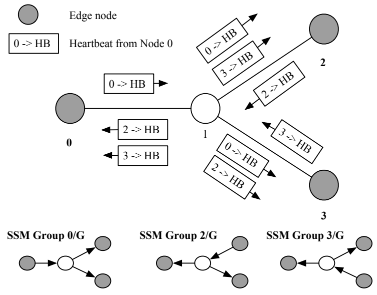


Figure 2: ERM Heartbeat

trol, the receivers on a multicast tree are ingress nodes to the egress node that is the multicast source.

To place the information in the packet, one must make several design decisions regarding the granularity, completeness, and relevance of the information to gather for the multicast receivers (edge nodes). First, what is the appropriate information to gather regarding a respective queue scheduler? Examples of various characteristics would include packet loss, instantaneous and weighted queue sizes, average queuing delay, and idle link bandwidth. A related question to ask is at what granularity and degree of completeness with regards to individual queue classes should such information be gathered. For instance, is it sufficient to gather statistics regarding only AF1x (Assured Forwarding 1, all loss classes) or it necessary to gather AF10, AF11, and AF12 (gold, silver, bronze loss priorities)? Depending upon the scheduler employed at the node, the relative load between various classes may have a distinctive effect on QoS.

Second, what queue information is relevant to the multicast receivers? For instance, while node 2 may be interested in the queuing information for the outgoing link from node 1 to node 2, node 3 may not be interested in such information. Furthermore, the edge-to-edge path information in the direction of node 1 to node 2 is of little relevance to node 2 as an ingress (see Figure 2) as the forward path may suffer from congestion while the reverse path does not. This is due to the fact that policing/admission control is done on a sender-wise basis and a forward-gathering approach typically gathers receiver-based QoS information.

RPF (reverse path forwarding) of PIM-SSM removes the need for reflection of the information back to the sender. If a heartbeat arrives on a specific interface, we know that the unicast traffic, which is routed by the underlying routing protocol, will take the outgoing queue on that interface in the reverse path. In Figure 2, the heartbeat received at node 1 from node 0 means that all other edge nodes (node 2 and node 3) will use the same path in reverse to transmit unicast traffic. Thus, the only relevant queue for all multicast receivers in this case would be the queue on that link from node 1 to node 0.

Figure 3 demonstrates this point in a relatively simple case. Normally, a probe packet gathering information (such as in [4,11]) from A to D arriving at D would be useless to D for admission control as it does not contain any information about the reverse direction. Thus, D would need to wait for a reflection of its heartbeat to A on A's heartbeat to deduce the status of its path from D to A. Beyond the additional network overhead of embedding reflection information, the staleness (age) of the information is also increased. With PIM-SSM, one can remove the need for reflection. By gathering information on the outgoing queue of the incoming interface, a heartbeat that arrives at D contains information about its reverse path to A.

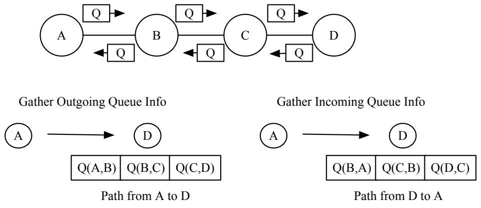
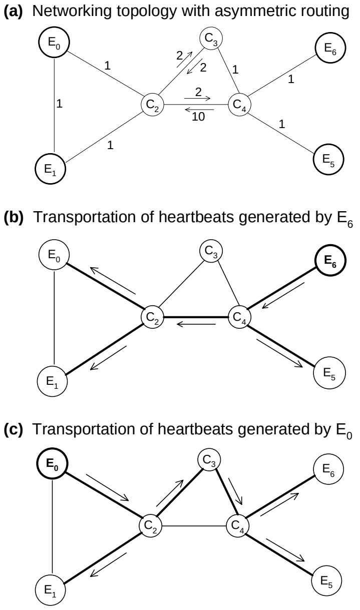


Figure 3: Reflection trade-off


(a) Networking topology with asymmetric routing

(b) Transportation of heartbeats generated by E_6

(c) Transportation of heartbeats generated by E_0

Figure 4: Heartbeat transportation for asymmetric routing

Figure 4 shows the heartbeat transportation for a networking topology under the condition of asymmetric routing. In this networking topology (Figure 4(a)), all links are assigned the same weight, 1, on both directions except that the weights of link $C_2 \rightarrow C_4$, $C_4 \rightarrow C_2$, $C_2 \rightarrow C_3$, $C_3 \rightarrow C_2$ are assigned 2, 10, 2, and 2 respectively. Hence, OSPF would create asymmetric routing between edge nodes. Figure 4(b) shows the paths of heartbeats emanating from E_6. The path from E_6 to E_0 in the multicast tree is the reverse of the minimal cost path from E_0 to E_6. Thus, when a heartbeat from E_6 arrives at E_0, E_0 can make use of the heartbeat for admission control as the heartbeat gathers the information for the unicast path from E_0 to E_6. In Figure 4(c), similarly, when a heartbeat from E_0 arrives at E_6, it gathers the information for the unicast path from E_6 to E_0.

To compute the staleness of the information gathered in heartbeats, for simplicity, we assume that the network employs a greedy heartbeat (all edges, all QoS info). Thus, the average case staleness for the picture of the internal network for edge node A going to edge node B can be stated as:

$HBI + EE_{Delay(B,A)} + EE_{QDelay(B,A)} + HB_{Delay(B,A)} + J_{B,A}$

where HBI is the heartbeat interval, $EE_{Delay(B,A)}$ is the minimum delay from B to A, $EE_{QDelay(B,A)}$ is the average queuing delay for a heartbeat packet, $HB_{Delay(B,A)}$ is the average transmission delay for the heartbeat packet, and $J_{B,A}$ is the delay jitter associated with the path from B to A. Nominally, both $EE_{QDelay(B,A)}$ and $J_{B,A}$ would approach zero given a high priority control marking and appropriate governing on the heartbeats themselves. The staleness of the core information is critical as it plays a role in determining how fast new flows can be admitted. For the admission control section, we assume that the maximum staleness of core information can be represented by βHBI.

3.3 Scalability & Processing Overhead

While the notion of greedy exploration provides a simple example for demonstrating the heartbeat operation, various practical issues such as scalability and network topology may preclude these assumptions. To start, we consider the impact of the greedy operation (probe all paths and all relevant class information on every single heartbeat):

- *Number of messages:* For a given link in the network that lies on a path between edge nodes, it will see at most $O(N)$ heartbeats where N is the number of edge routers. While the bandwidth itself is not zero, it will be negligible relative to the quantity of data traffic.
- *MTU Limitations:* At worst case, a heartbeat can only reach a size of $X \times InfoSize$ where X is the maximum number of hops between edge routers and $InfoSize$ is the amount of information gathered at each hop by the heartbeat. Reductions may be achieved through the use of probabilistic inclusion or round robin gathering of the respective class information. The reduction itself would be achieved at a cost of increased staleness and hence a larger βHBI.
- *Control processing:* While bandwidth will most likely not be an issue for the heartbeats in a wired networking environment, the impact on control processing is an extremely large concern. In keeping with the DiffServ principle of simple, per-flow stateless core routers, we mitigate the impact of control processing in several fashions. To start, core routers do not actively participate in signaling and are only responsible for appending information to the heartbeats themselves. We envisage that a heartbeat would simply append a segment of shared memory that contains the appropriate queue information onto the packet. The shared memory would be updated by the underlying queue scheduler and would not involve any significant processing for populating a heartbeat message beyond the memory copy itself. Furthermore, the router would have the option to ignore upper layer checksums (UDP) and rely on data layer checksums that are inherently more reliable to reduce the control processing impact. The HBI setting (and hence βHBI) can also be relaxed at the cost of efficiency to further reduce the control processing impact.
- *Multicast scalability:* A final aspect of scalability is the inclusion of multicast itself. While multicast deployment across the global Internet cannot be effectively relied upon in the near timeframe, ERM relies on multicast in a domain-wise sense only. Thus, the complex issues of inter-domain multicast routing and end user support for multicast are avoided. Second, the number of multicast groups is restricted to at most N where N is the number of edge routers in the domain.
- *Multi-pathing support:* Similar to the approach proposed for MTU limitations, a round robin approach can be employed to assess the availability of equal-cost multiple paths to an edge node.

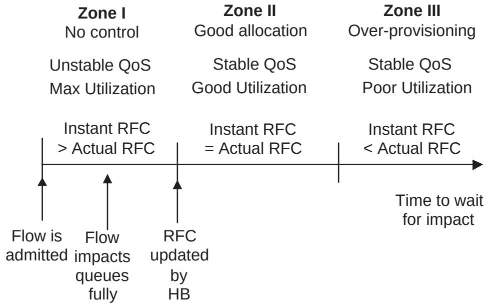


Figure 5: Effect of heartbeat lagging on system controllability

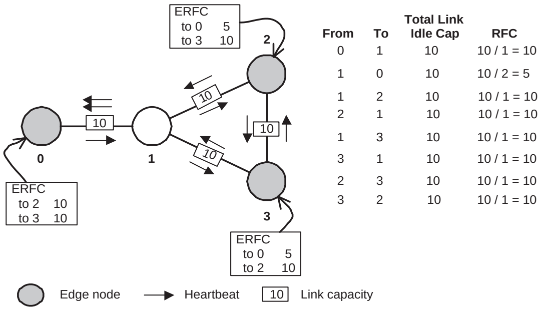
4. ERM ADMISSION CONTROL

As alluded earlier, the second key component of ERM is admission control. To allow for an appropriate abstraction of the network state picture provided by the heartbeat module, we assume that the staleness of the path from edge node A to edge node B can be at most βHBI seconds old. Unlike traditional approaches where a probe must first occur before admission control can proceed, an edge node has information regarding all paths with a maximum age of βHBI. Thus, admission can proceed immediately provided that appropriate resources exist.

In this relatively simple query lies the delicate balance for the entire system. In essence, one must quickly admit flows to reduce setup time but yet not admit too many flows so as to violate the overall QoS requirements for flows already existing in the network. As noted earlier, this problem is significantly complicated by the fact that heartbeats will always be a lagging factor versus the currently admitted flows in the network.

The net goal of the admission control scheme should be to operate in Zone II as identified in Figure 5 whereby a balance is struck versus speed of admission versus properly recognizing current flows in the network. If flows are admitted too quickly, the admission control quickly decays to an unstable state due to the fact that the effect of newly admitted traffic is not taken into proper account. In contrast, if flows are admitted too slowly, the utilization of the network will be poor due to the fact that scheme is waiting too long for the effect of new flows.

Furthermore, one must also ensure that each edge router has a 'fair' opportunity to compete for resources in the core. For admission control, we define *fairness* as being concerned with the allocation of remaining resources, not the actual treatment of flows or classes in the core of the network. For simplicity, we consider fairness as the fairness over a window (βHBI), not fairness in the long term.



From	To	Total Link Idle Cap	RFC
0	1	10	10 / 1 = 10
1	0	10	10 / 2 = 5
1	2	10	10 / 1 = 10
2	1	10	10 / 1 = 10
1	3	10	10 / 1 = 10
3	1	10	10 / 1 = 10
2	3	10	10 / 1 = 10
3	2	10	10 / 1 = 10

Figure 6: Example for RFC

To that end, we introduce a concept called *remaining fair capacity* or RFC. The goal of the RFC is to capture the maximum amount of resources available to an ingress point for a given link. By virtue of the heartbeats flowing between edge routers, a core router can deduce the remaining fair capacity (RFC) on a link through the following:

$$RFC_{X,Y} = \frac{IC_{X,Y}}{|HB|}$$

where IC is the weighted average idle capacity on the link from X to Y and $|HB|$ is the number of unique heartbeat sources seen on a given link. The RFC for each link is recalculated at an interval of βHBI which insures that new heartbeat information has been received in the interim. Intuitively, the RFC creates an equal opportunity for all of the ingress routers sharing the link on each βHBI interval to compete for the available free capacity. Without other governing mechanisms, RFC is an absolute maximum that would theoretically achieve a utilization of 100% if all edge nodes admitted up to the RFC onto the network. As will be discussed later, it will be highly desirable to govern the RFC so as to limit the overall network rate of change.

Although an instantaneous value of the average idle capacity could be used if βHBI is sufficiently small, the weighted fair capacity offers a better picture for larger (and more scalable) values of βHBI. Based on the link-wise value for RFC, the next natural step is to calculate the path-wise value at the ingress for RFC (see Figure 6):

$$RFC_{P(A,B)} = Min(RFC_{A,i_0}, \ldots, RFC_{i_{N-1},B})$$

where i_0 through i_{N-1} lie on the path between A and B. For a given ingress router with no recently admitted flows, RFC represents the net capacity available on a given path to an egress point that can be allocated in this βHBI. However, in the event that flows have been recently allocated, the RFC is a lagging factor as recently admitted flows have not yet had a chance to impact the RFC along the path. In fact, even the receipt of heartbeats after admission may not be sufficient either as flows may take time to reach their requested capacity (i.e. TCP). Thus, we introduce the concept of *effective remaining fair capacity* or $ERFC$ for the actual admission control decision:

$$ERFC_{P(A,B)} = RFC_{P(A,B)} - \sum_{i=0}^{N-1} RF_i$$

where N is the number of recently admitted flows affecting the path from A to B at this ingress router and RF_i is the influence of the ith recently admitted flow affecting the path, such as the request bandwidth of this flow. To qualify as recent, a flow must have been admitted within the last αHBI seconds. After ingress node A admits a flow, node A stores the information of this newly admitted flow (the egress point, the request bandwidth, etc.). Thus, $ERFC_{P(A,B)}$ can be computed at ingress node A easily as node A knows $RFC_{P(A,B)}$ as well. The intuition of αHBI is that a flow will have fully realized its impact on the network within αHBI. Figure 7 shows an example of the $ERFC$ calculation with RF_i fixed over the duration of αHBI.

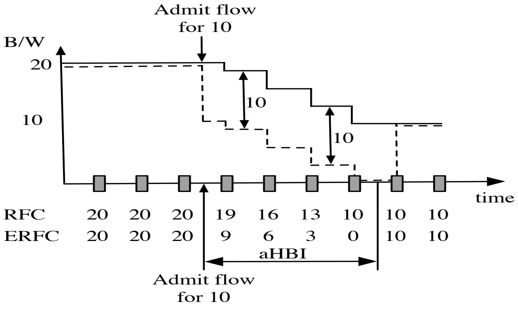


Figure 7: $ERFC$ example - constant RF_i

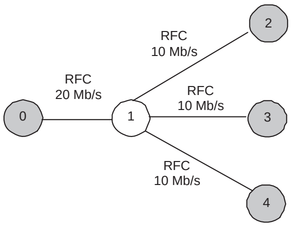


Figure 8: Path allocation dilemma

An important point to note is that such constant cost for a recent flow (RF_i) will result in under-utilization of the network. As the flow begins to exert an effect on the network and hence effect the RFC, the $ERFC$ will continue to decrease, despite the fact that the RF_i factor is already accounting for the full impact of the flow.

However, it is also important to note that an early slacking of RF_i before the flow exerts its full influence can have negative consequences. If the effect of RF_i and the impact of the flow measured by the heartbeats does not fully account for its total allocation, the ingress router will see a false picture of its available fair capacity and will over-allocate the link. Thus, although a constant RF_i does result in a temporary under-utilization (maximum duration of αHBI), this effect is more desirable than over-allocation. Furthermore, while decaying functions for RF_i are not precluded in ERM, the decaying function will be dependent upon the type of flow/aggregate which is an open topic for future work.

4.1 Inter-path dependency

A secondary dilemma for $ERFC$ arises from inter-path dependencies for a given ingress router. For instance, consider the topology in Figure 8 and the paths from node 0 to the other three edge routers (nodes 2, 3, and 4). For the three egress paths from node 0, the RFC values are calculated as 10, 10, and 10 Mb/s respectively. Most notably, since the link from 0 to 1 is not a bottleneck, it is ignored.

Thus, if inter-path dependencies are ignored, a total of 30 Mb/s (10 Mb/s per path) could be allocated on the link from 0 to 1. Even if the link from 0 to 1 was the bottleneck, it is still possible to allocate up to $N \times RFC$ bandwidth where N is the number of egress paths from the ingress sharing the link.

To solve this issue, several approaches are available with a variety of computational trade-offs. To start, the simplest approach is to further scale the RFC based on the number of egress paths competing for a link. In Figure 8, the RFC on the link from 0 to 1 would drop to $\frac{20}{3} = 6.67$ Mb/s. A variation on this approach is to calculate an inter-path dependency list on each topology and to have a newly admitted flow's RF_i affect all of the paths who would share links for that flow. This is done on path-wise basis, not a link-wise basis to improve the computational overhead. The net result of these two approaches is a simple but less efficient algorithm with a cost of $O(N)$ per flow admission where N is the number of edge routers and hence also the number of egress paths.

Alternatively, one could also force a new allocation to update the $ERFC$ value for each link that it affects. In turn, each path sharing those links would have its path-wise $ERFC$ updated as well. In such a case, RF_i is applied on a link-wise basis rather than a path-wise basis. Although this algorithm will grant a higher degree of precision and hence efficiency, its cost of $O(MN)$ or $O(N^2)$, where N is the number of edge nodes and M is the maximum path size, may be computationally infeasible.

4.2 Advanced Concepts

In this subsection, we summarize several of the advanced concepts (due to space constraints) that include:

- *Reducing ERM overhead:* While the use of RPF from PIM-SSM offers the most significant reduction in overhead (no need for reflecting path information), a variety of other techniques can be employed to reduce the overhead of ERM at a cost of a large βHBI. One approach to overhead reduction is to partition the egress routers into multiple trees that are iterated in a round robin fashion. A similar approach is to employ a probabilistic approach whereby core nodes evaluate their chance to include information using a fixed probability. Moreover, if traffic patterns are fairly well distributed among ingress nodes, (βHBI) can be further slackened to improve scalability. In such situations, it may be possible to combine ERM heartbeats with link state advertisements (LSAs) depending on the underlying LSA distribution mechanism.
- *Security:* A critical aspect of ERM is the notion that heartbeat messages and their respective contents can be considered secure. While heavier mechanisms such as digital signatures can be employed to verify the validity of the gathered data, such an approach can be extremely costly with little benefit. The reason for this lack of benefit is the presence of a security violation in a core or edge router represents significantly more serious problems beyond a malicious heartbeat. Furthermore, the distinct divisions of edge versus core router behavior provides the capacity to do physical interface-based filtering to prevent rogue heartbeat messages from external sources from being processed in the first place.
- *Bursty traffic:* Although not discussed explicitly, the presence of extremely bursty and coordinated flows could create difficulties for ERM. In such a scenario, a flow would reserve a set of resources and never consume the resources except in large, sporadic bursts. To ERM, the idle periods would appear that such capacity is again fair capacity that can be allocated. Although such a scenario is possible in theory, the occurrence of such a scenario is unlikely. Since ERM is assumed to be operating on core networks with high speeds, there is a fairly large probability that a large number of flows will be using the network. Given a large number of flows, it is reasonable to assume that the flows will not be coordinated so as to all burst simultaneously [8]. Rather, it is more likely that such flow bursts would be misaligned so as to multiplex together in a reasonable manner. To that end, the governing mechanisms of ERM can be tuned to operate in a safer mode in the event of large amounts of coordinated bursty traffic. Furthermore, ERM can send 'dummy' traffic so as to prevent idle but reserved traffic from otherwise being multiplexed. Thus, while potentially a problem with a low probability if left unchecked, we believe sufficient mechanisms can be applied to address this issue.
- *Non-uniformly distributed traffic:* While it may appear at first glance that an ingress router cannot acquire complete use of a link, ERM does possess facilities to achieve such a goal. In the event of unbalanced topologies or traffic matrices, an ingress node will be forced to compete over multiple allocation cycles to fully realize its unbalanced needs. Once the ingress node competes for and consumes the bandwidth over the appropriate links, the ingress node will possess that bandwidth until it slackens its usage over those links. It is important to note that the conservative nature of ERM will have a profound dampening effect when imbalances change frequently and dynamically.

  In addition, since ERM is targeted at the DiffServ environment in the core of the network where there will be large numbers of flows rather than only a few superflows, the additional time to achieve such an allocation is even less of an issue. Finally, although not discussed in this paper due to space requirements for complete elaboration, it is possible to adjust the weighting of the RFC given to a specific ingress router (rather than the default of weight of 1 by virtue of the heartbeat uniqueness) through additional signaling mechanisms.

5. SIMULATION STUDIES

For our simulation studies, the simulations were conducted using the ns-2 simulator and the GenMCast extension module. The goal of our studies was to examine the predictability of ERM and the controllability of the various parameters proposed for network intelligence gathering and admission control.

In our simulation studies, we compared four separate approaches, the DiffServ Bandwidth Broker (BB), ERM, GRIP [7], and EAC (End-to-End Admission Control) [5]. GRIP represents Probing Based Admission Control (PBAC) scheme with core router involvement. In our simulation studies, GRIP always tries to consume all of link bandwidth by setting its parameters according to the type of the traffic before a simulation begins. In addition, when a router in GRIP denies a new flow into the network by dropping its probe packet, the router will send a deny notification back to the ingress router of this flow. EAC is used to represent the Endpoint Admission Control scheme.

For our network topology, we used the backbone of Internet2 with two to four edge nodes connected to each core router and a link bandwidth of 100 Mb/s. The remaining information concerning the simulation setup is discussed below:

- Traffic was generated to work with a sizable number of flows and to ensure complete utilization of the links in the network. A typical core link will have hundreds of active flows present.
- The settings for ERM were an HBI of 0.1 s, β of 1, and α of 10. In this networking topology, a 1 second (αHBI) was sufficient for a flow to realize its impact on the network.
- The network was monitored from t=100 seconds.
- A token bucket policing mechanism was used on all flows such that any packets out of the profile were immediately dropped rather than remarked.
- Each flow will try to enter the network at most three times.
- The short/long term flows followed an exponential distribution with an average length of 5 and 50 seconds for UDP flows and an average size of 250K bytes and 2.5M bytes for TCP flows.
- For bursty traffic, both ON and OFF periods were exponentially distributed with mean values equal of 500 ms.
- The bandwidth QoS requirements of flows followed an exponentially distribution with a mean of 50KB/sec.
- A monitoring TCP flow of 50KB/sec was used to examine if the network provided stability for admitted flows.

For the simulations, we considered the three traffic scenarios listed below:

- *Traffic scenario I:* The first scenario is a simple CBR/UDP scenario to reflect the original considerations of GRIP [7] and EAC [5]. Flows were composed of 50% short term traffic and 50% long term traffic.
- *Traffic scenario II:* The second scenario introduces bursty flows. As presented earlier, bursty flows have the potential to create difficulties for ERM: to ERM, the idle periods will appear that such capacity is again fair capacity that can be allocated. Flows were composed of 50% CBR traffic and 50% VBR traffic, 50% short term traffic and 50% long term traffic.
- *Traffic scenario III:* Since the majority of traffic is dominated by TCP, the admission control scheme should also address TCP flows. In this scenario, flows were composed of 80% TCP traffic, 10% UDP-CBR traffic, and 10% UDP-VBR traffic. The composition of short term flows and long term flows is 80% and 20% to better reflect web traffic.

To ensure the comparison is fair, we use *real* link utilization as one of metrics. We define *real* link utilization is the ratio of actual data traffic to total link capacity. The approaches were evaluated along the following performance metrics:

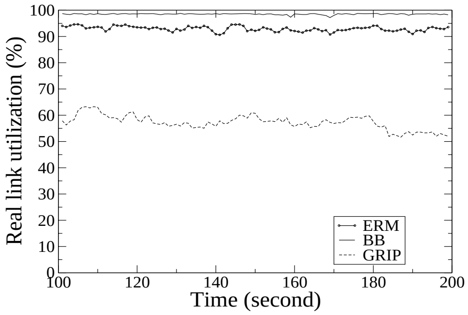


Figure 9: *Real* link utilization - Traffic scenario I

Table 1: Average response time and percentage of admitted flows in Traffic scenarios I and II

Sc.	Admission Control	Response Time (ms)	Percentage of Admitted Flows
I	BB	10.90	54.3
I	ERM	0	51.6
I	GRIP	3.12	29.4
II	BB	10.0	54.7
II	ERM	0	58.4
II	GRIP	3.38	32.5

- *Control Effectiveness:* The controllability is assessed by monitoring a single TCP flow that stays active over the course of the entire simulation. An over-allocation will result in the flow backing below its requested bandwidth.
- *Link Utilization:* A single link in the core of the network is monitored for its utilization. The end goal is to push the real link utilization as close to 100% as possible without decaying into an unstable state for the network.

5.1 Scenario I: CBR Traffic

Figure 9 shows the link utilization achieved by ERM, BB, and GRIP when traffic scenario I (100% CBR traffic) was used. Since the CBR traffic consumes its claimed QoS bandwidth, the Bandwidth Broker scheme achieves the highest *real* link utilization because this centralized scheme controls the network's condition precisely. Because heartbeats of ERM have a lagging view of the network and ERM heartbeats also consume a certain amount of bandwidth, ERM achieves a *real* link utilization slightly below the BB in this traffic scenario. Note, the overhead of admission control in the ERM scheme, is only dependent on the networking topology (including the number of edge nodes and the length of edge-to-edge paths). Thus, for the same networking topology, as network bandwidth increases, the percentage of overhead decreases.

In contrast, GRIP always over-estimates the link utilization to avoid the over-allocation of resources. While the router in GRIP estimates there is no free link bandwidth and stops admitting new flows, the actual free link bandwidth may be much higher, especially when traffic involves

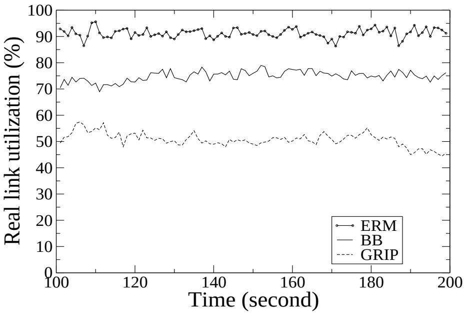


Figure 10: *Real* link utilization - Traffic scenario II

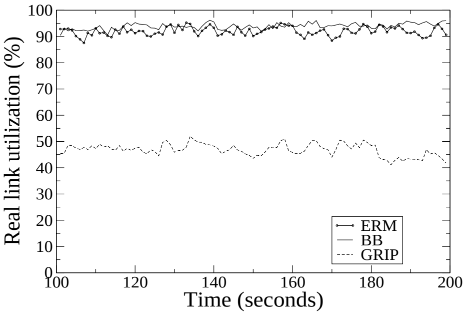


Figure 11: *Real* link utilization - Traffic scenario III

many short term flows. The link utilization of EAC was not shown in Figure 9 because the monitoring TCP flow could not receive stable service. In EAC, there is little way to control the link utilization since the link bandwidth will be consumed by both the actual traffic and the probe traffic. Thus, the monitoring TCP flow has to backoff when the network is heavy, which prevents it from receiving its claimed bandwidth. Hence, we do not include the simulation results of EAC in the figures.

Table 1 shows the average admission control response time and the percentage of admitted flows in traffic scenario I and traffic scenario II. Here, we only consider the transmission delay of the control packets (BB packets, probes and feedbacks). For simplicity, we assume the computation time of doing admission control in each node is equal to zero. For ERM, the edge router simply checks its own version of the available resources and makes the admission control decision immediately. Thus, there is no response delay for ERM. BB and GRIP had a proportional response time which was on the order of link delay, since both needed to send request packets to other nodes (the bandwidth broker in BB and the routers on the path in GRIP). Because the BB had the highest *real* link utilization in Figure 9, the number of admitted flows is the largest, as shown in Table 1. Even though the response time of EAC was not shown in Table 1, EAC had the longest response time. The reason lies in the fact that the probing period of each flow must be long enough to gather its path utilization precisely. The probing period of EAC is usually on the order of seconds as noted in [10].

5.2 Scenario II: CBR and VBR Traffic

Figure 10 shows the *real* link utilization achieved when traffic scenario II (50% CBR traffic and 50% VBR traffic) was used. In this scenario, ERM performed better than other admission control schemes, demonstrating ERM can handle the situation where non-synchronized bursty flows exist. When a flow is admitted by the Bandwidth Broker scheme, the BB always allocates this flow a certain amount of bandwidth according to the flow's QoS requirement, regardless of whether or not the flow really consumes it. Thus, when a flow only consumes part of its claimed bandwidth, the other part of the claimed bandwidth is wasted. GRIP has a similar issue: dummy packets are used to pretend each flow consumes all of its claimed bandwidth, thus allowing the router to compute how many flows exist on a link. In ERM, edge nodes do the admission control according to the information provided by heartbeats, which store the actual link utilization regardless how much bandwidth has been claimed by flows. When flows only consume part of their claimed bandwidth, ERM achieves high *real* link utilization by multiplexing [8].

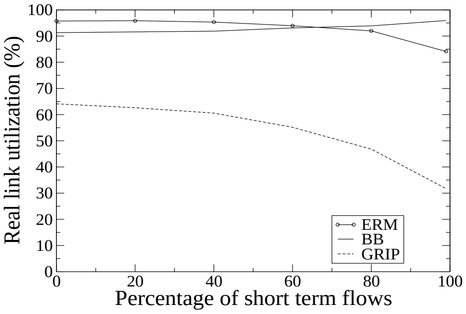


Figure 12: *Real* link utilization when the percentage of short term flows varied

5.3 Scenario III: CBR, VBR, and TCP Traffic

Figure 11 shows the *real* link utilization achieved when traffic scenario III (80% TCP, 10% CBR and 10% VBR) was used. Note that ERM still achieves similar *real* link utilization to the BB-oriented approach. In this scenario, GRIP performed worse because compared to the TCP source flows, the TCP ACK flows only claimed a small amount of bandwidth in their QoS requirements, which made the traffic more heterogeneous (GRIP works best for the homogeneous traffic).

In order to further explore the effects of short vs. long term flows, Figure 12 shows the *real* link utilization achieved as the ratio of the percentage of short term flows was varied. When the ratio of TCP and UDP traffic was set to a certain value (here, it was 4:1), the *real* link utilization in the BB scheme was stable. The ERM and GRIP schemes were both sensitive to the percentage of short term flows: the more the short term flows, the less the *real* link utilization. To avoid best-effort like behavior, distributed admission control schemes must over-estimate link utilization. Short term flows exacerbated the over-estimation to a larger degree (eg. α in ERM).

In order to further explore the effects of TCP vs. UDP flows, Figure 13 shows the *real* link utilization achieved when the percentage of TCP flows was varied. With the change of the number of TCP flows in the network, the performance of ERM was stable, since ERM used the actual link utilization to do the admission control. When the percentage of TCP flows increased, which meant the percentage of UDP-VBR flows decreased, the BB scheme achieved higher *real* link utilization because more claimed QoS bandwidth was

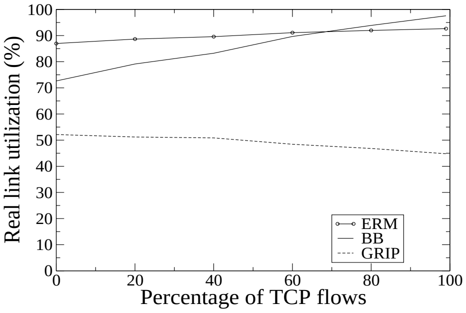


Figure 13: *Real* link utilization when the percentage of TCP traffic varied

actually consumed. GRIP performed worse when the ratio of TCP flows increased because more heterogeneous traffic caused less *real* link utilization.

Due to space constrains, the effects of different ERM parameters were not included in this paper. Intuitively, an increase in β causes the network to react more slowly and hence decreases utilization slightly. A larger α coupled with many short-term flows also decreases utilization since explicit resource release messages are not used.

6. SUMMARY

In summary, we presented in this paper a novel approach for DiffServ admission control that incorporated the use of coordinated heartbeats coupled with the concept of a distributed fair capacity measurement. We proposed intuitive methods for capturing the inherent lagging factor associated with heartbeats and scaling factors for further governing the overall system change rate. The ERM protocol captured the implicit trade-offs associated with heartbeat scalability and addressed methods to improve both heartbeat and admission control scalability and efficiency.

To evaluate the effectiveness of ERM, we compared ERM with the traditional DiffServ Bandwidth Broker scheme, GRIP, and EAC under a wide variety of network traffic conditions. The end result from the simulations showed that ERM could offer both controllability and efficiency, all in a scalable manner. Thus, we believe ERM offers a promising new approach that delivers a fast, scalable, and precise admission control for Differentiated Services.

With regards to future work, there are several issues that merit additional study. These issues include the expansion of fairness models, derivations of formal theoretical control models for ERM, and considerations for inter-domain admission control.

7. REFERENCES

[1] W. Almesberger, T. Ferrari, and J. L. Boulder. Scalable resource reservation for the Internet. In *Proc. of IEEE IWQoS'98*, Napa, CA, USA, 1998.

[2] A. Basu, A. Lin, and S. Ramanathan. Routing using potentials: a dynamic traffic-aware routing algorithm. In *SIGCOMM*, pages 37–48, 2003.

[3] S. Bhattacharyya. An overview of source-specific multicast (SSM). *RFC 3569*, July 2003.

[4] G. Bianchi and N. Blefari-Melazzi. Admission Control over Assured Forwarding PHBs: A Way to Provide Service Accuracy in a DiffServ Framework. In *Proc. of GLOBECOM*, San Antonio, Texas, Nov. 2001.

[5] G. Bianchi, F. Borgonovo, A. Capone, L. Fratta, and C.Petrioli. Endpoint admission control with delay variation measurements for QoS in IP networks. *ACM Computer Communication Review*, pages 61–69, April 2002.

[6] G. Bianchi, A. Capone, and C. Petrioli. Throughput analysis of end-to-end measurement-based admission control in IP. In *Proc. of IEEE INFOCOM 2000*, Israel, Mar. 2000.

[7] N. Blefari-Melazzi and M. Femminella. A comparison of the utilization efficiency between a stateful and a stateless admission control in IP networks in a heterogeneous traffic case. *Telecommunication Systems, Kluwer Academic Publishers*, pages 231–258, March/April 2004.

[8] T. Bonald and J. Roberts. Congestion at flow level and the impact of user behaviour. *Computer Networks*, pages 521–536, July 2003.

[9] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) Version 1, Functional Specification. *IETF RFC 2205*, Sept. 1997.

[10] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint admission control: Architectural issues and performance. In *Proc. of ACM SIGCOMM*, September 2000.

[11] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Transactions on Information Theory*, 45(7):2462–2480, Nov. 1999.

[12] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *Proc. of IEEE INFOCOM*, Apr. 2001.

[13] V. Elek, G. Karlsson, and R. Ronngren. Admission control based on end-to-end measurements. In *IEEE INFOCOM 2000*, Israel, 2000.

[14] K. Lai and M. Baker. Measuring link bandwidth using a deterministic model of packet delay. In *Proc. of ACM SIGCOMM'00*, 2000.

[15] J. Moy. OSPF Version 2. *IETF RFC 1583*, June 1994.

[16] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services field (DS Field) in the IPv4 and IPv6 headers. *IETF RFC 2474*, Dec. 1998.

[17] R.Braden, D. Clark, and S. Shenkar. Integrated Services in the Internet architecture: An overview. *IETF RFC 1633*, June 1994.

[18] F. Reichmeyer, K. Chan, D. Durham, R. Yavatkar, S. Gai, K. McCloghrie, and S. Herzog. COPS usage for Differentiated Services. *IETF Internet Draft*, Nov. 1998. Work in Progress.

[19] A. Striegel and G. Manimaran. Edge-based fault-detection in DiffServ networks. In *Proc. of IEEE Dependable Systems and Networks (DSN)*, Washington, D.C., June 2002.

[20] D. Zhu, M. Gritter, and D. R. Cheriton. Feedback based routing. *Computer Communication Review*, 33(1):71–76, 2003.