

# Adaptive Per-Flow Traffic Engineering Based on Probe Packet Measurements

Sven Krasser, Henry L. Owen  
School of Electrical and  
Computer Engineering  
Georgia Institute of Technology  
Atlanta, Georgia 30332–0250, USA  
{sven, owen}@ece.gatech.edu

Joachim Sokol, Hans-Peter Huth,  
Jochen Grimminger  
Siemens AG, CT IC2 Corporate Technology  
81730 Munich, Germany  
{joachim.sokol, hans-peter.huth,  
jochen.grimminger}@siemens.com

## Abstract

*In this research, we propose a new connection admission control and online traffic engineering framework. The framework is designed to fit small networks using differentiated services, e.g. radio access networks. Decisions are made at the edge routers of the network. Multiple disjoint label switched paths are pre-configured between each pair of edge routers (ERs). Between each pair of ERs, probe packets are sent on every path between those ERs and for every class of service. The characteristics of the transmission are measured at the ER at the end of the path, the egress ER. It sends the results back in feedback packets to the ingress ER at the beginning of the path. Additionally, low priority probe packets are sent at high rates to discover and reserve available bandwidth. The achieved throughput of those probes is also reported in feedback packets. Based on the results in these feedback packets, ERs render an admission decision for new connection requests and pick a path.*

## 1 Introduction

As Internet technologies replace other network architectures, the need for support of multiple distinct service classes in the Internet protocol (IP) arises. A scalable approach to this is to use the differentiated services (DiffServ) framework. DiffServ gives no strict quality of service (QoS) guarantees. Instead, packets are relayed according to different per-hop behaviors (PHBs) at each router. The DiffServ codepoint (DSCP), which is transmitted in the type of service header field of an IP packet (assuming version 4), determines the PHB of a packet. Several PHBs have already been defined: expedited forwarding (EF), a high priority service trying to achieve zero packet loss, minimal queuing delay, and minimal jitter; assured forwarding (AF), a group

of several PHBs giving a variety of different forwarding assurances by defining four classes (distinguished by the resources available per class, namely buffer space and bandwidth) with three different drop precedences; and best effort (BE), a low priority service equivalent to the service in DiffServ-unaware networks (like today's Internet).

## 2 Related Work

In [2], Elwalid *et al.* present a framework for multiprotocol label switching (MPLS) adaptive traffic engineering (MATE). MATE is designed to balance fluctuations of the network load by sending traffic flows to the same destination over different routes. These routes are implemented as label switched paths (LSPs). MATE gathers state information of these LSPs by using probe packets that are sent periodically from the ingress to the egress router of an LSP. These statistics and a path cost function are used to decide to which LSP a traffic flow is shifted.

Nelakuditi and Zhang propose and evaluate a framework both to select a set of candidate paths between a source and a destination node and to pick paths out of this set for new flows [7]. A widest disjoint path algorithm is used to calculate the set of candidate paths with the maximum width (the total amount of bandwidth that can be transmitted over this set of paths) based on global state information that is distributed in sparse intervals. A path for a new incoming flow is picked such that all paths in the set have equal blocking probabilities based on information locally available at the node. The results show that the proposed scheme outperforms best path routing based on a widest shortest path algorithm while minimizing the signaling overhead.

Barlow *et al.* present the local state fair share bandwidth (LSFSB) algorithm, a traffic engineering algorithm for radio access networks, in [1]. LSFSB assumes DiffServ, multiprotocol label switching (MPLS), and Hierarchical Mobile IPv6 support throughout the network. The design aims to be simple and fast, and it uses the available resources sparingly.

The algorithm relies on local state information only. Local in the scope of LSFSB refers to state information available on a single node. LSFSB does not require distribution of network state information. The radio access servers (RASs) on the edge of the network choose the LSP on which admitted traffic is forwarded.

In [6], we propose a connection admission control (CAC) and flow-based traffic engineering framework for small DiffServ domains based on path queue states. The proposed algorithm renders its decision based on path queue state (PQS) information gathered by edge routers (ERs). Each ER gathers information on the states of the queues on all paths to each peer ER it has. Then, the expected QoS properties for each path are computed. ERs render admission decisions based on this information and pick a suitable path for newly admitted traffic flows.

### 3 Overview

For this research, we use a similar network setup as in [6]. ERs at the rim of the network render admission decisions for incoming connection requests. Admitted traffic is forwarded from an ingress ER to an egress ER over one path out of a set of multiple paths connecting those ERs. Each ER gathers information on the states of all paths to each peer ER it has for every class of service by means of probe packets. Additionally, each ER uses low-priority packets to reserve and discover available bandwidth. Based on this information, the expected QoS properties for each path are computed. ERs render admission decisions based on this information and pick a suitable path for newly admitted traffic flows.

Figure 1 shows a simple example of the topology of the assumed radio access network. Mobile devices connect to base stations (both not shown in Figure 1) attached to radio access servers on the rim of the network. Those connect to edge routers (ERs) on the edge of the core network. There is one special ER called the edge gateway (EGW) in the core network that interfaces other RANs and the Internet. The ERs are ingress and egress points into the network. The capacity per link in the core network is 10 Mbps. In general, the network can also contain core routers (CRs), which only relay and do not inject traffic.

We consider four different services with distinct traffic flow characteristics in our simulator labeled by their PHB and their transport protocol: *ef-udp*, *af-udp*, *af-tcp*, and *be-tcp*. The characteristics used were chosen to demonstrate proof of concept.

The UDP traffic classes send traffic bidirectional with a constant bit rate of 80 kbps (*ef-udp*) and 100 kbps (*af-udp*), respectively. The TCP traffic classes send traffic unidirectional with a variable bit rate, but are policed to 100 kbps using a token bucket at the ingress ER.

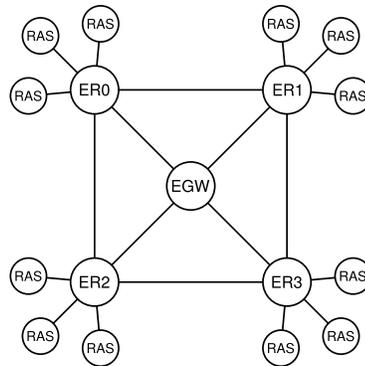


Figure 1. Topology of simulated test network.

When a connection fails to achieve its minimum QoS requirements, it is terminated. For *ef-udp*, no more than one loss per 2 second interval is tolerated. Both *af-udp* and *af-tcp* have to maintain a loss rate of no more than 2 packets during the last 2 seconds. Currently, there is no minimum QoS requirement for *be-tcp*.

Calls arrive at a rate  $\lambda$  at the network. There is a probability of  $p_{EGW}$  that the EGW is part of a connection and a probability of  $p_{ER0}$  that ER0 is part of a connection (see Figure 1).

Blocked connection requests are dropped. Hence, the connection request arrival process as seen by the CAC follows a memoryless Poisson distribution.

## 4 The Probe Packet-Based Traffic Engineering and Connection Admission Control Algorithm

As outlined previously, ERs send probe packets to each other to gather information on path properties. Three different metrics are considered: the probed path delay, the probed path loss, and the bandwidth reserved by low-priority probes.

### 4.1 Probing Packet Delay

The probes used to measure the delay are 64 byte-sized packets sent every 250 msec from an ingress to an egress ER. Every 1000 msec, a feedback packet is sent back by the egress ER. The delay measured comprises the propagation delay, the queuing delay, and the transmission delay. Note, that the transmission delay for probe packets and the transmission delay for normal data packets are generally not the same. Under the assumptions made for our network setup, an additional 100 bytes of packet size increase the transmission delay by 0.08 msec per link. We consider this as a negligible effect.

## 4.2 Probing Packet Loss

The loss is probed using the same probes as for probing path delays. Losses are detected by missing sequence numbers in the probe stream. All losses in a 3 second measurement window are considered. Because of the limited number of probes in the measurement window, low loss probabilities, which we desire to have in the network, cannot be estimated precisely enough. A better precision could be achieved in two ways. First, the rate probe packets are sent with can be increased to have more packets in the measurement window. However, this increases the intrusiveness since more actual bandwidth is consumed by probes. Second, the temporal interval of the measurement window can be increased. This worsens the dynamic properties of the estimate.

We use the probed loss probability as a means to penalize paths with detected losses rather than as an exact estimate of the expected amount of lost packets.

## 4.3 Discovering and Reserving Available Bandwidth

Streams of low-priority probe packets are used to discover and reserve available bandwidth. We presented some related ideas previously in [5] and [4].

The new scheme proposed in this paper comprises two additional kinds of probes: discovery probes (DPs) and reservation probes (RPs). On each node, these probe packet types and data packets are serviced in a priority queuing fashion. Data packets have the highest priority, DPs the lowest priority. In case there is no data packet to be dequeued, the queue with the next lower priority, the RP queue, is considered. If there is no RP to dequeue, a DP is dequeued from the lowest priority queue. If this queue is also empty, the outgoing link stays idle until a packet becomes available.

Each pair of ERs sends DPs on every path with a fixed rate of  $R^{\text{DP}}$ . For example, ER $j$  sends DPs with a rate of  $R_{j,k}^{\text{DP}} = R^{\text{DP}}$  to ER $k$ . For the sake of brevity, we do not take different paths between ER $j$  and ER $k$  into account in this nomenclature. ER $k$  measures the rate these DPs are received with, denoted  $\tilde{M}_{j,k}^{\text{DP}}$  and reports it back to ER $j$  in a feedback packet. Depending on whether there is enough bandwidth on each link of a path to accommodate all DPs,  $\tilde{M}_{j,k}^{\text{DP}}$  is either close to  $R^{\text{DP}}$  or lower. If there is not enough bandwidth on a link for all DPs, DPs are dropped. The available bandwidth on that link is then shared among the DP flows. However, ER $j$  can assume that bandwidth amounting to at least  $\tilde{M}_{j,k}^{\text{DP}}$  is available.

Similar to DPs, ERs send out RPs. For RPs, the rate is not fixed but corresponds to the actual reservation. ER $j$  sends out RPs at a rate  $R_{j,k}^{\text{RP}}$ , and ER $k$  reports back its mea-

surement,  $\tilde{M}_{j,k}^{\text{RP}}$ . To make a reservation, ER $j$  increases  $R_{j,k}^{\text{RP}}$  by a value of  $Q^{\text{RP}} \leq \tilde{M}_{j,k}^{\text{DP}}$ . Since  $\tilde{M}_{j,k}^{\text{DP}}$  is only a part of the available bandwidth resulting from the competition of multiple DP flows, this scheme guarantees that even if all ERs simultaneously try to increase their reservation there is enough available bandwidth for all RP flows.

Claiming a reservation is straightforward. To increase the use of bandwidth for data flows by  $Q^{\text{data}}$ , the ER decreases its RP rate by  $Q^{\text{data}}$ . Once the data connection commences and the RP rate is decreased, this reservation is lost, i.e. after termination of the data connection, the ER does not get back its reserved bandwidth.

Algorithm 1 is used to adjust the current amount of reserved bandwidth, i.e. the rate of RPs,  $R^{\text{RP}}$ . This algorithm runs on every edge router for every LSP. It is invoked each time reserved bandwidth is used for sending data on a path. Additionally, it is invoked in regular intervals.

First, it is checked whether reservations are blocked. Such blocking can result from certain changes of  $R^{\text{RP}}$  by the algorithm. If reservations are not blocked, the algorithm proceeds. Otherwise, no changes are made and the algorithm exits.

Next, it is checked if the measured DP throughput  $\tilde{M}^{\text{DP}}$  falls below a minimum threshold  $M_{\text{min}}^{\text{DP}}$ . If that is the case, then it is deduced that too many reservations are made, and the reservation  $R^{\text{RP}}$  is decreased by a factor  $f_{\text{decr},1}$ . Furthermore, changes of the RP rate other than using reserved bandwidth are blocked for a duration of  $t_{\text{block}}$ .

After this, the expected bandwidth demand  $d$  is calculated. The implemented path bandwidth demand estimation works in a very straightforward manner. Every time a connection request from ER $j$  to ER $k$  is issued, ER $j$  adds the requested bandwidth and the time of the request to a list associated with the LSP that fits all requirements (besides the bandwidth needs) best. The estimated bandwidth demand per LSP is then calculated as 1.4 times the amount of bandwidth expected to be requested during the length of a blocking interval  $t_{\text{block}}$  based on the bandwidth needs in the LSP's list during the last 2 seconds. Bandwidth needs older than 2 seconds are erased from the list.

A refined algorithm can address issues like mobility estimation and known properties of the demand (e.g. an increased demand during certain hours of the day).

In case the calculated demand undershoots the minimum demand parameter  $d_{\text{min}}$ ,  $d$  is changed to  $d_{\text{min}}$ . This is to assure that even a non-utilized path can accept at least one new incoming connection request.

If the current reservation  $R^{\text{RP}}$  is lower than the demand  $d$ ,  $R^{\text{RP}}$  is increased to fit the demand. However,  $R^{\text{RP}}$  is never increased by more than the measured DP rate  $\tilde{M}^{\text{DP}}$ . Afterwards, changes of the reservation other than claiming reserved bandwidth are blocked for an interval of length  $t_{\text{block}}$  as described previously. This is to assure that the

**Table 1. Parameters of the reservation scheme**

Parameter	Default value	Description
$R^{\text{DP}}$	700 kbps	Fixed DP rate
$M_{\min}^{\text{DP}}$	1000 bps	Minimum allowed DP throughput rate before node decreases RP rate
$d_{\min}$	150 kbps	Minimal assumed demand
$f_{\text{decr},1}$	0.9	Factor to decrease RP rate if DP throughput is low
$f_{\text{decr},2}$	0.5	Maximal factor to decrease RP rate if RP rate exceeds demand
$t_{\text{block}}$	1.2 sec	Duration of blocking interval

measurement  $\tilde{M}^{\text{DP}}$  reflects the changes the next time it is considered.

The current reservation  $R^{\text{RP}}$  is decreased in case it is higher than the demand  $d$ . In general,  $R^{\text{RP}}$  is changed to fit the demand, but it is never decreased below  $f_{\text{decr},2} \cdot R^{\text{RP}}$  where  $f_{\text{decr},2}$  is the maximum decrease factor.

The parameters mentioned above, their default values, and a brief description of their functions are listed in Table 1.

The sole purpose of RPs is to use up bandwidth before the actual data traffic the reservation has been made for starts. In other words, RPs are used to indirectly signal that bandwidth is not available anymore by squeezing out DPs. Furthermore, for medium to high connection arrival rates, the RP rate set is used up for data traffic long before it can have an impact on the measured DP rate. Since in this scenario at the same time many connections end, this does not pose a problem to the scheme.

#### 4.4 Admission and Path Selection

First, all candidate paths are examined. If a path does not fulfill all criteria for the requested traffic class, it is pruned from the set of candidate paths. Then, the best path is picked from the remaining ones.

For EF, the probed delay must not exceed 26 msec and the considered path is not allowed to be penalized for recent probe losses. Moreover, 1.25 times of the requested bandwidth must be reserved. For AF, the considered path is not allowed to be penalized for probe losses, and the bandwidth requested must be reserved. For BE, only the requested bandwidth must be reserved.

For EF, the path with the lowest delay is picked; for AF,

**Algorithm 1:** Reservation adjustment algorithm.

---

```

if not blocked then
  if  $\tilde{M}^{\text{DP}} < M_{\min}^{\text{DP}}$  then
     $R^{\text{RP}} \leftarrow f_{\text{decr},1} \cdot R^{\text{RP}}$ 
    block for  $t_{\text{block}}$ 
  else
    calculate demand  $d$ 
     $d \leftarrow \max\{d, d_{\min}\}$ 
    if  $R^{\text{RP}} > d$  then
      if  $R^{\text{RP}} - d > f_{\text{decr},2} \cdot R^{\text{RP}}$  then
         $R^{\text{RP}} \leftarrow f_{\text{decr},2} \cdot R^{\text{RP}}$ 
      else
         $R^{\text{RP}} \leftarrow d$ 
      end
    else
       $R^{\text{RP}} \leftarrow \min\{R^{\text{RP}} + \tilde{M}^{\text{DP}}, d\}$ 
      block for  $t_{\text{block}}$ 
    end
  end
end

```

---

the path with the lowest loss penalty value is picked; and for BE, again the path with the lowest delay is picked. If there is no path remaining in the set of candidate paths, the connection request is rejected.

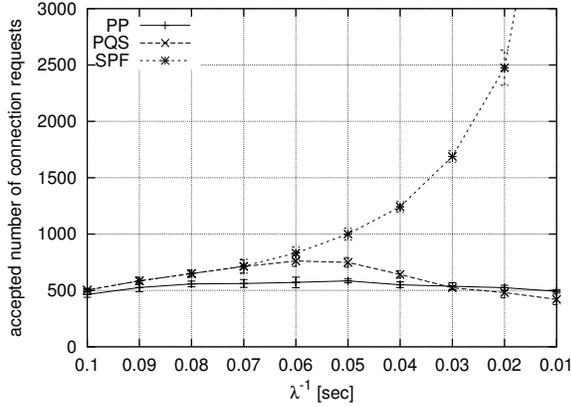
## 5 Results

The following results are gathered by simulating the outlined network system using an augmented version of ns-2 2.1b9a for 150 seconds after a 60 second initialization and warm-up period.

We compare three different algorithms: the probe packet-based algorithm (PP) as proposed by this paper, the path queue state-based algorithm (PQS) as described in [6], and shortest path first routing (SPF). The latter does not use any CAC and admits all connection requests to the network. However, if a call cannot maintain the minimum QoS requirements, it is terminated. Therefore, all calls maintain the minimum QoS as long as they are active. We use this property as a baseline to evaluate our algorithm.

### 5.1 Connection Admission Control

Figure 2 shows the number of admitted EF connection requests for each of the three algorithms. Note the reversed  $x$ -axis with high connection request inter-arrival times corresponding to a low load on the left side. Because the AF results look very similar, we do not show them in this paper. Since SPF uses no CAC, the number of accepted connection requests in this scheme reflects the total number of requests. The number of requests accepted by PP and PQS stays moderately constant even though the number is increased for



**Figure 2. Number of accepted EF connection requests for  $p_{ER0} = 0.3$  and  $p_{EGW} = 0.5$ .**

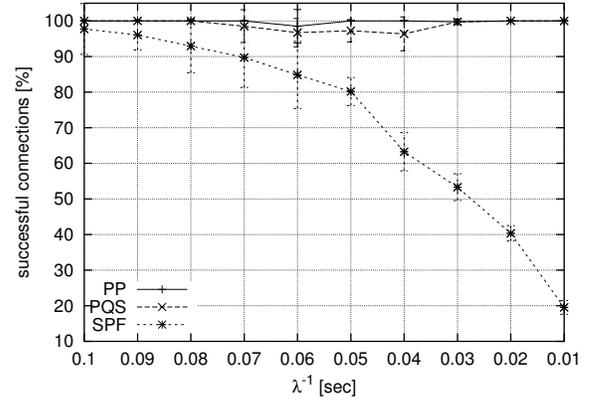
medium inter-arrival times around 0.06 seconds. PQS is able to follow the actual demand very closely as long as the network is not loaded (high inter-arrival times). PP is more conservative in this interval and blocks some requests, but its results are more constant over the whole scale.

In Figure 3, we present the percentage of successful EF connections. The CAC-less SPF approach is not able to guarantee a high percentage of successful calls while both PP and PQS are able to achieve this goal. However, a few connections especially in the PQS scheme had to be terminated due to a too low QoS.

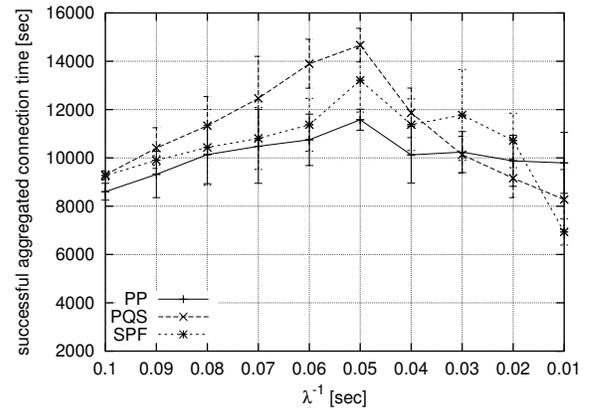
The aggregated time of successful connections is depicted for EF in Figure 4 and for AF in Figure 5. This value is the total time of all connections as long as they are in their QoS specification. For EF, a slight peak can again be noticed for medium inter-arrival times. PQS carries the most connections in this interval, and PP has over nearly the whole simulated scale the shortest aggregated time. For AF, PQS performs best for medium inter-arrival times as can be deduced by the higher aggregated time. PP performs better for low inter-arrival times (high load). However, the SPF results show that there is still room for additional connections that is not used.

## 5.2 Quality of Service

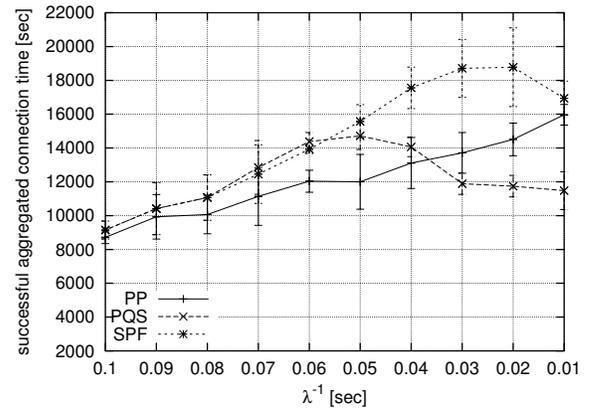
Figure 6, Figure 7, and Figure 8 show cumulative distribution functions (CDFs) of the packet delay for a connection request inter-arrival time of 5 msec, an ERO probability of 0.3, and different EGW probabilities for PP, PQS, and SPF, respectively. Both PP and PQS can guarantee high probabilities for low delays while SPF fails as expected in this respect. PP packet delays are approximately bound by the 26 msec admission criteria as can be seen in the sub-graphs in Figure 6. The graphs for  $p_{EGW} = 0.6$  and



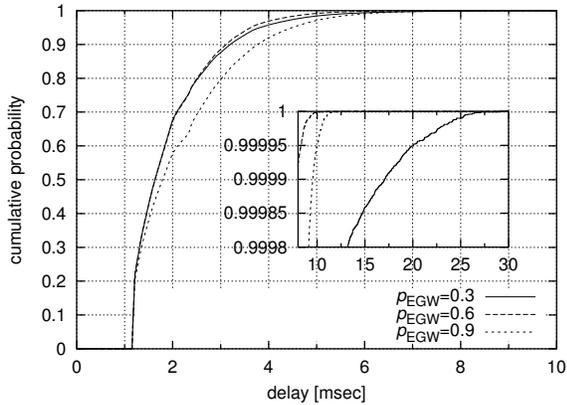
**Figure 3. Percentage of successful EF connections for  $p_{ER0} = 0.3$  and  $p_{EGW} = 0.5$ .**



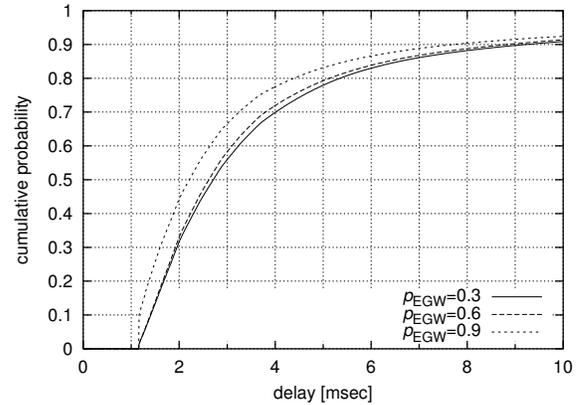
**Figure 4. Aggregated time of successful EF connections for  $p_{ER0} = 0.3$  and  $p_{EGW} = 0.5$ .**



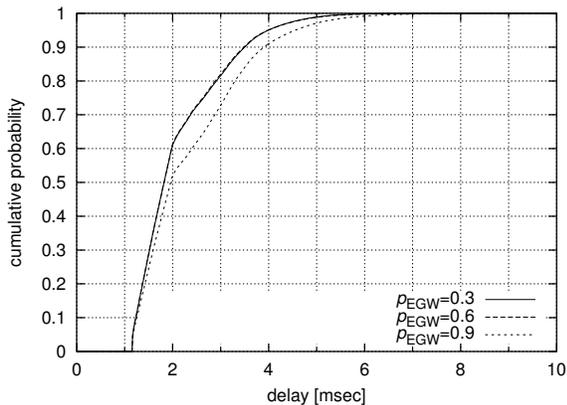
**Figure 5. Aggregated time of successful AF connections for  $p_{ER0} = 0.3$  and  $p_{EGW} = 0.5$ .**



**Figure 6. Cumulative distribution functions of EF packet delay for  $\lambda^{-1} = 5$  msec and  $p_{ER0} = 0.3$  using the PP algorithm.**



**Figure 8. Cumulative distribution functions of EF packet delay for  $\lambda^{-1} = 5$  msec and  $p_{ER0} = 0.3$  using the SPF algorithm.**



**Figure 7. Cumulative distribution functions of EF packet delay for  $\lambda^{-1} = 5$  msec and  $p_{ER0} = 0.3$  using the PQS algorithm.**

$p_{EGW} = 0.9$  show even better properties.

## 6 Conclusion

We proposed a new traffic engineering and connection admission control scheme based on measurements of probe packet transmissions. The scheme is able to give QoS guarantees to the connections it is carrying. We compared the scheme to the PQS algorithm outlined in [6] and to a shortest path routing approach. Both the proposed algorithm and PQS perform equally well. However, there is still room for additional traffic in the network that both algorithms are not able to utilize. A more detailed analysis regarding performance and overhead can be found in [3].

Future work includes more refined admission and path

selection functions. Especially the latter has to penalize longer paths more so that no resources are wasted.

## References

- [1] D. Barlow, H. Owen, V. Vassiliou, J. Grimminger, H.-P. Huth, and J. Sokol. Router-based traffic engineering in MPLS/DiffServ/HMIP radio access networks. In *Proc. IASTED International Conference on Wireless and Optical Communications*, pages 360–365, 2002.
- [2] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *Proc. IEEE INFOCOM 2001*, pages 1300–1309, 2001.
- [3] S. Krasser. *Adaptive Measurement-Based Traffic Engineering in Packet Switched Radio Access Networks*. PhD thesis, Georgia Institute of Technology, June 2004.
- [4] S. Krasser, H. Owen, J. Grimminger, H.-P. Huth, and J. Sokol. Distributed bandwidth reservation by probing for available bandwidth. In *Proc. IEEE International Conference on Networks 2003*, pages 443–448, 2003.
- [5] S. Krasser, H. Owen, J. Grimminger, H.-P. Huth, and J. Sokol. Probing available bandwidth in radio access networks. In *Proc. IEEE Global Communications Conference 2003*, volume 6, pages 3437–3441, 2003.
- [6] S. Krasser, H. Owen, J. Grimminger, H.-P. Huth, and J. Sokol. Online traffic engineering and connection admission control based on path queue states. In *Proc. IEEE SoutheastCon 2004*, pages 255–260, 2004.
- [7] S. Nelakuditi and Z.-L. Zhang. A localized adaptive proportioning approach to QoS routing. *IEEE Communications Magazine*, 40(6):66–71, June 2002.