Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks[1]

B. Nandy, N. Seddigh, P. Pieda, J. Ethridge
Nortel Networks, Ottawa, Canada
Email:{bnandy, nseddigh, ppieda, jethridg}@nortelnetworks.com

Version 7

ABSTRACT

Issues related to bandwidth assurance in Assured Forwarding based Differentiated Services (Diffserv) networks have been discussed in recent research papers [7] [8][11]. Some of the factors that can bias bandwidth assurance are Round Trip Time (RTT), UDP/TCP interaction and different target rates. The bias due to these factors needs to be mitigated before bandwidth assurance for a paying customer can be articulated in Service Level Agreements (SLAs). This paper proposes intelligent traffic conditioning approaches at the edge of the network to mitigate the effect of Round Trip Time, UDP/TCP interactions, and different target rates. The simulation results show a significant improvement in bandwidth assurance with intelligent traffic conditioning. The limitation of the proposed solutions is that they require communication between edge devices. In addition, these solutions are not applicable for a one-to-any network topology.

1. Introduction

The Differentiated Services (Diffserv) architecture [2] has recently become the preferred method to address QoS issues in IP networks. This packet marking based approach to IP-QoS is attractive due to its simplicity and scalability. An end-to-end differentiated service is obtained by the concatenation of per-domain services and Service Level Agreements (SLAs) between adjoining domains along the source-to-destination traffic path. Per domain services are realized by traffic conditioning at the edge and simple differentiated forwarding mechanisms at the core of the network. Two forwarding mechanisms recently standardized by the IETF are the Expedited Forwarding (EF) [6] and Assured Forwarding (AF) [5] Per Hop Behaviors (PHBs).

The basis of the AF PHB is differentiated dropping of packets during congestion at the router. The differentiated dropping is achieved via "RED-like" [1] Active Queue Management (AQM) techniques. The AF PHB RFC specifies four classes and three levels of drop precedence per class. AF is an extension of the RIO [3] scheme, which uses a single FIFO queue and two levels of drop precedence.

To build an end to end service with AF, subscribed traffic profiles for customers are maintained at the traffic conditioning nodes at the edge of the network. The aggregated traffic is monitored and packets are marked at the traffic conditioner. When the measured traffic exceeds the committed target rate, the packets are marked with higher drop precedence (DP1); otherwise, packets are marked with lower drop precedence (DP0). If the measured traffic exceeds the peak

[1] An abridged version of this paper is accepted for publication in High Performance Networking 2000 Conference, May 2000, Paris, France

target rate, the packets are marked with highest drop precedence (DP2). At the core of the network, at the time of congestion, the packets with DP1 marking have higher probability of being dropped than packets with DP0 marking. Similarly, packets with DP2 marking have higher probability of being dropped than packets with DP0 and DP1 marking. The different drop probabilities are achieved by maintaining three different sets of RED parameters – one for each of the drop precedence markings

Although the IETF Diffserv Working Group has finalized the basic building blocks for Diffserv, we argue that there are many open issues in understanding and evaluating the kinds of end-to-end services that could be created for an end user using the AF PHB. Various issues with bandwidth assurance in a Diffserv network have been reported in recent research papers [4][11]. A number of these issues need to be resolved before quantitative assurances of some form can be specified in SLA contracts.

Bandwidth assurance can be improved by intelligent treatment of aggregated flows at the core or at the edge of the network. Any approach to mitigate the impact of Round Trip Time (RTT), TCP/UDP interaction or target rate requires state tracking. Maintaining per-flow or per-policy state information at the core of the network will cause scalability concern. Another alternative is to address the bandwidth assurance issues at the edge of the network via intelligent traffic conditioning.

The key contribution of this paper is the proposal of intelligent traffic conditioners to mitigate the effects of various factors in biasing the achieved bandwidth. An RTT-Aware Marker based on the Time Sliding Window (TSW) [3] is developed to reduce the effects of RTT in determining the achieved bandwidth for TCP flows. Extensive study is performed to consider whether UDP/TCP fairness issues can be solved via intelligent mapping of TCP and UDP traffic to different drop precedence or AF classes. Finally, two Target Rate-Aware Markers are presented with the objective of distributing excess bandwidth in proportion to the target rates.

The rest of this paper is organized as follows. Related work is examined in the next Section. Section 3 describes the topology of the test network and various simulation parameters. Section 4 presents the solution to mitigate the impact of RTT. TCP/UDP interaction issues are addressed in Section 5. An algorithm for excess bandwidth distribution in proportion to target rates is discussed in Section 6. Section 7 provides an analysis, discussion and evaluation of the proposed solutions. Section 8 contains concluding remarks and points to areas of future work.

2. Related Work

Clark and Fang [3] reported the initial simulation study on a differentiated drop scheme. Their paper introduced RIO (RED with In/Out) and a remarking policer that utilized an average time sliding window (TSW) rate estimator and intelligent marker. The main contribution of that work was to show that source target rates could be assured in a simple capacity allocated network that relies on statistical multiplexing. Ibanez and Nichols [4], via simulation studies, showed that RTT, target rate and TCP/UDP interactions are key factors in the throughput of flows that obtain an Assured Service using a RIO-like scheme. Their main conclusion is that such an Assured Service "cannot offer a quantifiable service to TCP traffic". Seddigh, Nandy and Pieda [11] have confirmed with detailed experimental study that the above mentioned factors are critical for biasing distribution of excess bandwidth in an over-provisioned network. In addition, it has been shown [11] that the number of micro-flows in an aggregate and packet sizes play key roles in determining the bandwidth achieved in over-provisioned networks.

Recently, various researchers [7][12][14] have reported new approaches to mitigate the biasing effects of some of the factors outlined in [4] and [11]. Lin, Zheng and Hou[7] have proposed an

enhanced TSW profiler and two enhanced RIO queue management algorithms. Their simulation results show that the combination of enhanced algorithms improves the throughput and fairness requirements especially with different target rates, RTTs and co-existing UDP flows. However, the proposed solutions may not be scaleable due to the usage of state information at the core of the network.

Yeom and Reddy [12] have suggested an algorithm that improves fairness for the case where the individual flows in an aggregate have different RTTs. The proposed algorithm maintains per-flow information at the edge of the network. Kim [14] proposes a token allocation scheme to distribute tokens to individual flows originating from the same subscriber network. The paper claims that using this approach, fairness in TCP and UDP interaction and fairness between TCP connections with different RTTs can be achieved. The details of the algorithm are not clearly reported in the IETF draft[14].

3. Simulation Detail

The studies in this paper were performed using the ns-2 simulator [15]. The simulator was enhanced to include networking elements with Diffserv edge and core device functionality as specified in [2].

The network topology used in the experiments can be seen in Figure 1. The setup consists of three network edge devices E1, E2, E3 and one core device C1. Each edge device is connected to an end host or traffic source. The TCP flows generated are all long lasting. Experiments with the RTT-Aware Traffic Conditioner are performed with the network topology shown in Figure 1. The topology of the network for the experiments with TCP/UDP interaction is an extension of Figure 1. Six edges are connected to six separate traffic sources. The Target Rate-Aware Traffic Conditioner also utilizes the same topology with six edges and sources. The bottleneck link is between core device C1 and edge device E3.

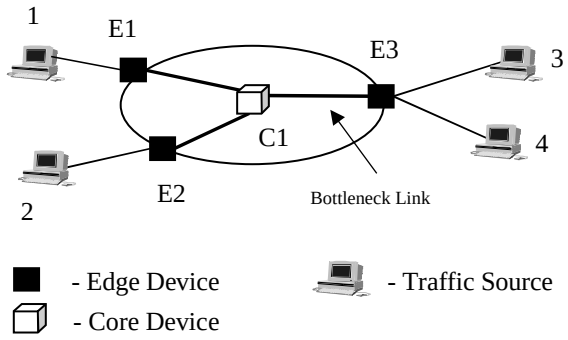

Figure 1: Simulation Testbed

The Edge devices in the testbed classify packets based on source and destination IP addresses. The traffic conditioning scheme is a remarking policer that utilizes the Time Sliding Window (TSW) tagger [3] scheme proposed to work in conjunction with RIO. This is referred to as the Standard Traffic Conditioner (TC).

The core device implements the AF PHB using the three-colour version of RIO [13]. Three sets of RED thresholds are maintained in the core device; one for each drop precedence. Three separate average buffer occupancy or queue length calculations are tracked: one for DP0 packets (q0), one for DP1 packets (q1) and one for DP2 (q2) packets. The probability of dropping DP0 packets depends only on the buffer occupancy q0. The probability of dropping DP1 packets depends on

the total buffer occupancy of q0 plus q1. The probability of dropping DP2 packets depends on the total buffer occupancy of q0 plus q1 plus q2. This scheme gives the appearance of three coupled virtual queues within a physical queue.

	Drop Precedence 0	Drop Precedence 1	Drop Precedence 2
$\mathbf{Min_{th}}$	40 pkts	25 pkts	10 pkts
$\mathbf{Max_{th}}$	55 pkts	40 pkts	25 pkts
$\mathbf{Max_p}$	0.02	0.05	0.1
$\mathbf{w_q}$	0.002	0.002	0.002

Table 1: RED Parameter Settings

Careful considerations need to be given to the setting of RED parameters for the three different Drop Precedences. The experiments in this paper unless otherwise specified utilize the RED parameter settings in Table 1. The min_{th} and max_{th} thresholds are selected so that no lower drop precedence packets are dropped till all higher drop precedence packets are being dropped.

4. Mitigating the Impact of Round Trip Time

Studies have shown that in Best Effort networks [9], the bandwidth achieved by TCP flows is a function of the Round Trip Time (RTT). This dependency is due to TCP's use of a self-clocked sliding window based mechanism. Recent studies [11] have shown that flows with different RTTs, despite having identical target rates, will get different shares of the bandwidth. For over-provisioned networks, the flows will mostly achieve their target rate irrespective of their RTTs[11][12]. This is because DP0 traffic is protected and DP1 traffic will be dropped before any DP0 packets are dropped. However, there will be an unfair sharing of the excess bandwidth in favor of those target aggregates with lower RTTs. In the under-provisioned case, neither of the aggregated flows will achieve its target. However, the flows with a high RTT will be further away from the target than the flows with a low RTT.

An initial simulation is performed to show the impact of RTT on bandwidth and to develop the basis for the RTT-Aware Traffic Conditioner. Two traffic aggregates are generated. Each aggregate has target rate of 2 Mbps. Each aggregate (between client 1 and 3; and between client 2 and 4) has six TCP flows. This profile results in a total allocated bandwidth of 4 Mbps, which is 40% of the bandwidth at the bottleneck link. The transmission delay between edges E1 and E3 (RTT_{13}) is kept at 20 ms while RTT_{24} (between client 2 and 4) is varied from 1 to 200 ms.

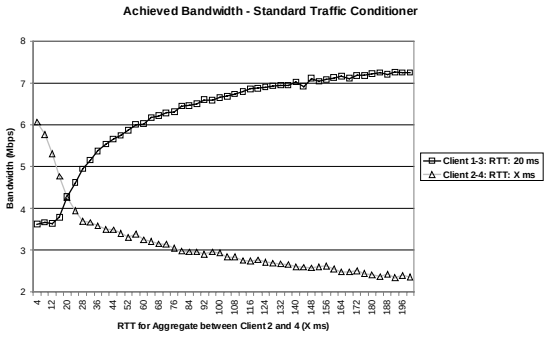

Figure 2: Achieved BW Using Standard TC

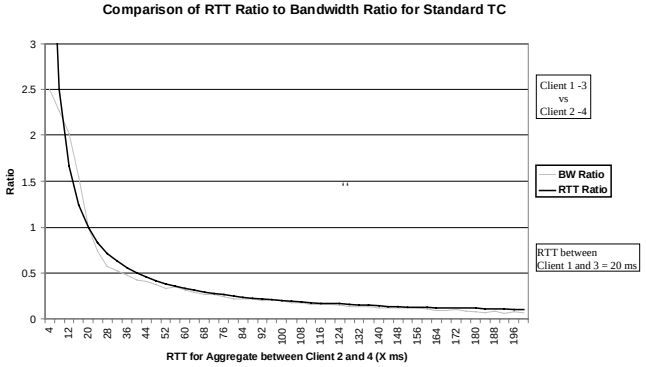

Figure 3: BW Ratio Vs. RTT Ratio

Figure 2 shows the total bandwidth achieved by each aggregate. Figure shows that as the RTT between client 2 and 4 is increased, the share of bandwidth of the aggregate decreases. The result reflects the steady state TCP behavior as reported by Mathis et al. [9]. Equation (1) shows that the BW is inversely proportional to RTT.

$$BW \propto \frac{MSS}{RTT * \sqrt{p}}$$, where MSS is the segment size and p is packet drop probability (1)

As the drop rate and MSS are same for both traffic aggregates, from Equation (1) the BW ratios can be represented as:

$$\frac{BW_{24}}{BW_{13}} = \frac{RTT_{13}}{RTT_{24}} \quad (2)$$

Figure 3 plots the ratio of RTTs and bandwidth from the simulation results. It shows that the ratio of the two RTTs is identical to the inverse ratio of the measured TCP aggregate bandwidth between clients 1-3 and 2-4, thus verifying Equation 2. Equation 1 forms the basis of the RTT-Aware traffic conditioner.

RTT-Aware Traffic Conditioning

Various approaches are possible to address the impact of RTT on TCP throughput. One approach is to modify the TCP windowing mechanism at the end host and make it RTT aware. A second method is to use the knowledge of RTT to affect dropping at the congested core devices. A third alternative is to introduce a mechanism at the edge of the network to handle the impact of RTT on throughput. We have taken the third approach.

Equation 1 shows that if the packet drop rate can be adjusted in relation to RTT, the acquired bandwidth for the aggregate can be made less sensitive to RTT. This idea is the basis of the RTT-Aware traffic conditioning algorithm. The aggregates with high RTTs take longer to ramp up after a packet drop occurs. Thus, the achieved average bandwidths for high RTT aggregates are lower. Protecting a higher amount of traffic for long RTT aggregates can compensate for the loss in bandwidth. Our approach increases the amount of in-profile traffic for high RTT aggregates in a proportional manner.

If (*measuredRate* <= *TargetRate*)
/* i.e., IN-profile */
Map Packets to "dp0"
Else /* i.e., OUT-of-profile */
Map Packets to "dp0" with probability (1-**p)**
Map Packets to "dp1" with probability **p**

Where: **p** = **q** * **r**

$$\mathbf{q} = \frac{(MeasuredRate - TargetRate)}{MeasuredRate}$$

$$\mathbf{r} = \left(\frac{\min RTT}{aggregateRTT} \right)^2$$

Figure 4: RTT-Aware TC Algorithm

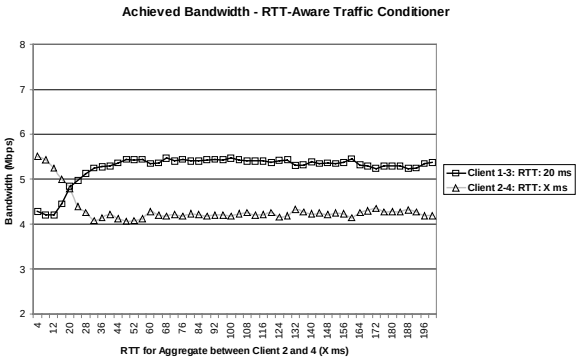

Figure 5: Achieved BW using RTT-Aware TC

Figure 4 outlines the RTT-Aware packet marking algorithm. The algorithm is an extension of the TSW marker[3]. A detailed derivation of the algorithm is presented in Appendix A. As long as the measured sending rate remains below the target rate packets are marked with DP0. Beyond the target rate, packets are marked DP1 with probability **p** and DP0 with probability (1-**p**). The probability **p** is calculated using knowledge of the traffic stream's measured RTT relative to the minimum RTT (minRTT) in the DS domain. For traffic streams with lower RTT, packets beyond the target rate will get marked to DP1 with higher probability. At the time of congestion, more packets with DP1 marking will be susceptible to dropping, thus adjusting the achieved bandwidth. Three assumptions for this scheme are: (a) all the flows in the aggregate have the same

RTT i.e., source and destination points are the same; (b) minRTT of the network is known to all the edge devices; (c) RTT for the aggregate flow is known at the edge of the network.

We repeat the same experiment for which the result was shown in Figure 2; except the RTT-Aware TC is used instead of standard TC. Figure 5 shows the results. Comparing Figure 5 with Figure 2, we observe that the impact of RTT has been significantly mitigated. The two aggregates achieve a similar share of the excess bandwidth.

One major assumption for the RTT-Aware TC is that the TCP flows are operating in congestion avoidance state. Equation (1) is not representative of bandwidth achieved if flows are in slow start. With a large number of flows in an aggregate and inappropriate setting of RED parameters, many flows can timeout and enter slow-start repeatedly. In such a case, it has been observed that the RTT-Aware TC is less effective in mitigating the impact of RTT in biasing bandwidth distribution. The next sub-section discusses the issues with large number of flows and studies the applicability of the proposed RTT-Aware marking algorithm.

Issues with Large Number of Active Flows

The total number of active flows in the core of the network and the buffer allocation plays an important role [10] in determining the TCP throughput for individual flows as well as flow aggregates. Large number of active TCP flows will cause the queue length to cross the RED max_{th} value and drop multiple packets causing timeout.

Simulation is performed using the standard TC with RED parameters as in Table 1. There are 200 flows per aggregate between clients 1 and 3, and 2 and 4 respectively. It is observed that with increased RTT between client 2-4, the difference in aggregate bandwidth between Clients 1-3, and Clients 2-4 are less than that shown in Figure 2. The bandwidth does not follow the relationship shown in Equation 1. This is due to the fact that at any given time more than 40% of the flows were incurring timeout. When, the RTT-Aware traffic conditioning was applied, the impact of RTT did not get resolved. Again, the proposed solution did not work since a large number of flows are not in steady state (i.e., not obeying Equation 1).

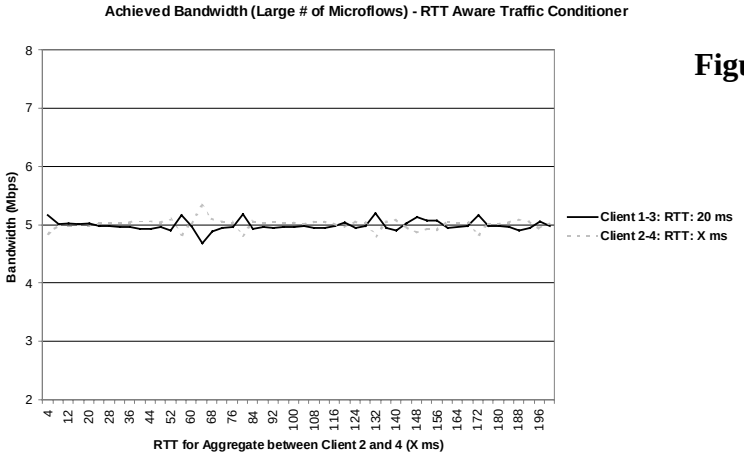

Figure 6: Achieved BW using RTT-Aware TC (200 microflows)

Two solutions can be considered for this problem. The first solution is to develop an RTT-Aware TC that takes into account the possibility of flows entering timeout. This is difficult since some of the flows will be in slow start and some in congestion avoidance state. A second solution is to use larger buffers. Proper engineering of RED parameters are key to this problem. The simulation is repeated with significantly larger buffers i.e., very high RED min_{th} and max_{th}. It is observed from Figure 6 that the RTT-Aware TC is capable of compensating the bandwidth differences for large number of flows.

5. Addressing Fairness issues with TCP/UDP Interactions

A paying Diffserv customer will inject both TCP and UDP traffic to the Diffserv network. The interaction between TCP and UDP may cause the unresponsive UDP traffic to impact the TCP traffic in an adverse manner. There clearly, is a need to ensure that responsive TCP flows are protected from non-responsive UDP flows, but at the same time to protect certain UDP flows which require the same fair treatment as TCP due to multimedia demands. Moreover, we argue it is the Diffserv customer who should decide the importance of the payload assuming the network is capable of handling both TCP and UDP traffic in a fair manner. We suggest that three fairness criteria for TCP and UDP traffic are:

1. In an over-provisioned network, both UDP and TCP target rates should be achieved.

2. In an over-provisioned network, UDP and TCP packets should have a reasonable share of the excess bandwidth. Neither TCP nor UDP should be denied access to the excess bandwidth.

3. In an under-provisioned network, TCP and UDP flows should experience degradation in proportion to their target bandwidth.

There are two possible approaches to solve the fairness issues: (a) Mapping TCP and UDP to different drop precedence of the same AF class, (b) Mapping TCP and UDP to different AF class queues.

Experiments are performed with two UDP sources with target rates of 1Mbps and sending rate of 6 Mbps each CBR. High sending rates of UDP flows are chosen so that the impact of UDP on TCP can be easily evaluated. Four TCP aggregates are generated with each aggregate consisting of 32 flows. The target rate of each TCP aggregate is varied from 0.5Mbps to 3Mbps. Thus the total target rates of UDP and TCP aggregates are varied from 4Mbps to 14Mbps so that bandwidth allocation at the core of the network changes from over-provisioned state (40% allocated capacity) to under-provisioned state (140% allocated capacity).

Mapping TCP and UDP to Different Drop Precedence

A drop precedence mapping scheme is one way to ensure fairness for both TCP and UDP. This study attempts to evaluate various options of mapping TCP and UDP to different drop precedence based on the matrix in Table 1. In all scenarios, TCP traffic within the target bandwidth ("IN-Profile") is assigned to DP0. In scenarios 1 and 6, UDP in-profile traffic is also assigned to DP0. UDP in-profile traffic assignment to DP1 (in scenarios 2, 3 and 4) and to DP2 (in scenario 5) are also considered. The experiments are performed with intelligent TC to perform appropriate mapping at the edge of the network.

Table 1: Possibilities for Mapping TCP and UDP to different drop precedences

Scenarios	1	2	3	4	5	6
TCP–IN Profile	DP0	DP0	DP0	DP0	DP0	DP0
TCP–OUT-of Profile	DP1	DP1	DP1	DP2	DP1	DP1
UDP–IN Profile	DP0	DP1	DP1	DP1	DP2	DP0
UDP–OUT-of Profile	DP1	DP1*	DP2	DP2	DP2 *	DP2

* No distinction is made between UDP–IN and UDP–OUT packets

In Scenario 1, both UDP and TCP in-profile packets are mapped to DP0 and out-of-profile packets are mapped to DP1. Both TCP and UDP flows achieve their target bandwidth in an over-provisioned network (Figure 7). The UDP flows get most of the share of the excess bandwidth. As the network approaches an under-provisioned state, the TCP flows suffer more degradation than the UDP flows. This is due to identical mapping of TCP and UDP out-of-profile traffic.

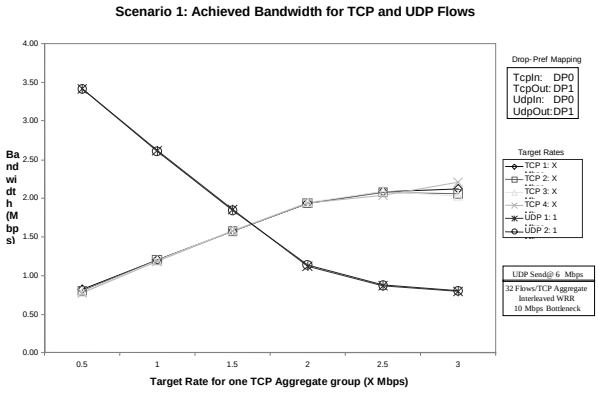

Figure 7: Scenario 1

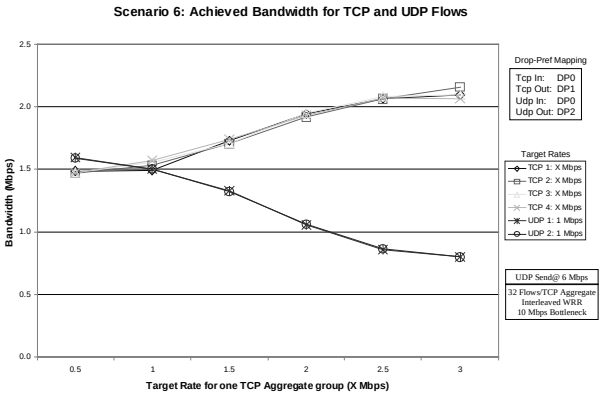

Figure 8: Scenario 6

It is observed that Scenarios 2 to 5 cannot assure UDP target bandwidth. This is due to the allocation of UDP in-profile traffic to DP1 or DP2. Thus, UDP in-profile traffic is dependent on DP0 TCP traffic. Total in-profile TCP traffic determines if the target rate of UDP can be achieved or not. In other words, the buffer occupancy of UDP in-profile traffic is dependent on the TCP in-profile traffic in DP0. Sharing of excess bandwidth is dependent on the assigned drop precedence of TCP and UDP. Similar argument is true for under-provisioned scenario.

In Scenario 6 both TCP and UDP in-profile packets are mapped to DP0. However, TCP out-of-profile packets are mapped to DP1, while UDP out-of-profile packets are mapped to DP2. The results are shown in Figure 8. In the over-provisioned case, both TCP and UDP achieve their target bandwidth. However, TCP obtains a greater share of the excess bandwidth than UDP. In an under-provisioned network, the TCP flows experience greater degradation from their target bandwidth than the UDP flows.

The results show that the target bandwidth for TCP and UDP flows can be achieved by protecting the in-profile traffic and mapping it to DP0. For an over-provisioned network, the manner in which the excess bandwidth is shared (i.e., fairness criteria 2) remains dependent on the drop precedence assignment of TCP and UDP out-of-profile packets. In an under-provisioned network (i.e., fairness criteria 3), isolation of TCP and UDP in-profile traffic is necessary. Mapping both UDP and TCP in-profile to the same drop precedence (i.e., Scenarios 1 and 6) results in unfairness to TCP, as it experiences degradation from its target bandwidth in comparison to UDP.

Mapping TCP and UDP to Different AF Class Queues

Another way to achieve fairness is to completely isolate the TCP and UDP traffic in two separate AF class queues at the core of the network. At the edge of the network, the intelligent TC marks the TCP and UDP packets to different AF classes. A weighted scheduling scheme is used at the core to enforce fairness among TCP and UDP flow aggregates. If the weights of the scheduling class queues are distributed in proportion to the TCP and UDP target rates, the fairness criteria can be satisfied. The weights for the queues can be selected using following method:

$$W_{TCP} = \left(\frac{\sum_{i=1}^{n} R_{TCP}^{i}}{\left(\sum_{i=1}^{n} R_{TCP}^{i} + \sum_{i=1}^{m} R_{UDP}^{i} \right)} \right) \qquad W_{UDP} = \left(\frac{\sum_{i=1}^{m} R_{UDP}^{i}}{\left(\sum_{i=1}^{n} R_{TCP}^{i} + \sum_{i=1}^{m} R_{UDP}^{i} \right)} \right)$$

where R_{TCP}^{i}: Target Rate for TCP aggregate i
R_{UDP}^{i}: Target Rate for UDP aggregate i
W_{UDP}: Weight for UDP Scheduling Queue
W_{TCP}: Weight for TCP Scheduling Queue

The above equations set the weight assuming that UDP aggregates are sending packets at rate equivalent to or greater than their target rates. TCP traffic and UDP traffic are mapped to different AF classes. IN-profile traffic is mapped to DP0 and OUT-of-profile traffic is mapped to DP1. A weighted round robin scheduler is used to schedule packets between two queues at the core of the network. The traffic mix is the same as that used for the results of Figure 7 and Figure 8.

The results are depicted in Figure 9. It is observed that all the three fairness criteria are satisfied. Both TCP and UDP achieve their target rates. In over-provisioned case, TCP and UDP obtain a reasonably fair share of the excess bandwidth. In the under-provisioned case, the aggregated bandwidth for TCP and UDP degrades proportionally.

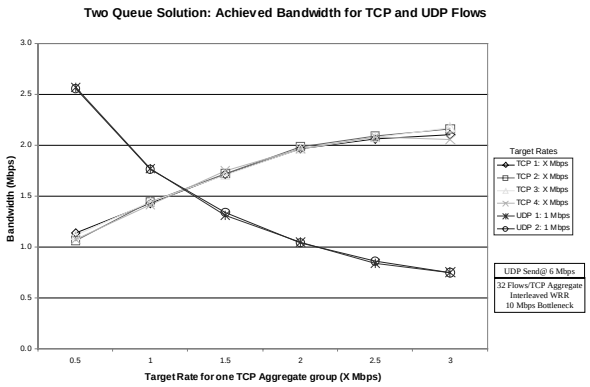

Figure 9: Scenario: Two Queue

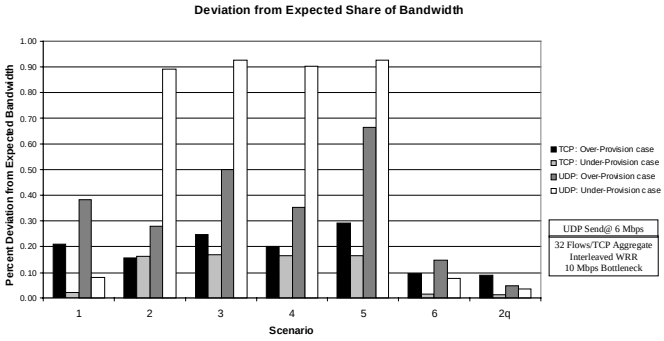

Figure 10: Deviation from Expected Bandwidth

Fairness Analysis

To further compare the results against the original fairness criteria, quantitative analysis is performed. The analysis compares the expected versus the actual bandwidth obtained for each of the seven scenarios. The percentage deviation from expected share of bandwidth is calculated. These values give two sets of averages, one for the four TCP aggregates and one for the two UDP aggregates. To facilitate detailed analysis, we distinguish between under-provisioned and over-provisioned cases. For the over-provisioned calculation, we use points for case of TCP aggregate target rate with values 0.5, 1, 1.5; for under-provisioned, we use 2, 2.5 and 3.

Equation (3) calculates the expected fair share of the bandwidth for UDP customers. The maximum sending rate of UDP is considered by taking the minimum between the UDP stream maximum sending rate and the fair share bandwidth.

$$\text{Expected UDP Agg BW} = Min\left\{\left(\frac{R^i_{UDP}}{\left(\sum_{i=1}^{n} R^i_{TCP} + \sum_{j=1}^{m} R^j_{UDP}\right)}\right) \times BW_{link}, S^i{}_{udp}\right\} \quad (3)$$

R^i_{udp} = Target rate (Mbps) for UDP customer j

R^j_{tcp} = Target rate (Mbps) for TCP customer i

BW_{link} = Link bandwidth (Mbps)

S^i_{udp} = Maximum sending rate (Mbps) for UDP customer i

Equation (5) determines the expected fair share of the bandwidth for TCP aggregates. Unused bandwidth (Equation 4) from the UDP aggregate(s) is divided between the TCP aggregates, proportional to their target rate. UDP aggregates have unused bandwidth when their sending rate is below the target rate.

Total Unused UDP BW is given by:

$$U_{UDP} = Max\left\{\left[\sum_{k=1}^{m}\left(\frac{R^k_{UDP}}{\left(\sum_{i=1}^{n} R^i_{TCP} + \sum_{j=1}^{m} R^j_{UDP}\right)}\right) \times BW_{link} - \sum_{k=1}^{m} S^k_{udp}\right], 0\right\} \quad (4)$$

$$\text{Expected TCP Customer BW} = \left(\frac{R^i_{TCP}}{\left(\sum_{i=1}^{n} R^i_{TCP} + \sum_{i=1}^{m} R^i_{UDP}\right)}\right) \times \left(BW_{link} + U_{udp}\right) \quad (5)$$

$$\text{Deviation from Expected BW} = \left|\frac{ExpectedBandwidth - MeasuredBandwidth}{ExpectedBandwidth}\right| \quad (6)$$

The graph in Figure 10 illustrates the results of the quantitative analysis. From the graph we can see that the test with two class queues had the least deviation from expected BW. Deviation in Scenario 6 is also comparable to the test with two class queues. Scenario 1 has high deviation for excess BW (for both TCP and UDP). UDP performs poorly for under provisioned cases in Scenario 2-5.

6. Excess BW Distribution for Aggregates with Different Target Rates

In a Diffserv network, different customers will contract different target rates. Recent research has shown that in an over-provisioned network, with standard TC, there is an almost even distribution of excess bandwidth irrespective of the target rate[11]. This may not be an acceptable solution, as the high paying customer with a higher target rate will expect a higher share of the excess bandwidth. Further discussion on the merit of equal versus proportional distribution of excess bandwidth can be found in section 7. This work assumes proportional distribution is desirable.

This section describes and evaluates two intelligent traffic conditioners developed to address the issue of proportional distribution of excess bandwidth. The first solution uses DP0 and DP1 and

is referred to as Target Aware TC with two drop precedence (TATC-2DP). The TATC-2DP approach is similar to the RTT-Aware TC. The excess out-of-profile traffic is allocated back to in-profile in proportion to the target rates. This will lead to higher assured bandwidth for aggregates with high target rate. The algorithm in Figure 11 outlines the TATC-2D marking scheme for the traffic conditioner.

The second solution uses all three drop precedence and is called TATC-3DP. In this scheme, the excess bandwidth is divided between DP1 and DP2 in proportion to the target rate. The algorithm for the TATC-3D is captured in Figure 12.

If (*measuredRate* <= *TargetRate*)
/* i.e., IN-profile */
Map Packets to "dp0"
Else /* i.e., OUT-of-profile */
Map Packets to "dp0" with probability (1-**p);**
Map Packets to "dp1" with probability **p;**

Where: **p = q * r :**

$$\mathbf{q} = \frac{(MeasuredRate - T\arg etRate)}{MeasuredRate}$$

$$\mathbf{r} = \left(\frac{\min TrgetRate}{aggregateTrgetRates} \right)^2$$

Figure 11: The TATC-2D Algorithm

If (*measuredRate* <= *TargetRate*)
/* i.e., IN-profile */
Map Packets to "dp0"
Else /* i.e., OUT-of-profile */
Map Packets to "dp0" with probability (1-**p);**
If (packet is not marked "dp0")
Map Packets to "dp1" with probability **1-q;**
Map Packets to "dp2" with probability **q;**

Where: **p** and **q:**

$$\mathbf{p} = \frac{(MeasuredRate - TargetRate)}{MeasuredRate}$$

$$\mathbf{q} = \left(\frac{\min TargetRate}{AggregateTargetRate} \right)$$

Figure 12: The TATC-3D Algorithm

We perform the same set of experiments using both the TATC-2DP and the TATC-3DP. The first experiment is performed with two sets of aggregates from clients 1 to 3 and 2 to 4 respectively. Each aggregate consists of six TCP flows. One aggregate has a target rate of 1Mbps and the other aggregate has a target rate that is varied between 0.5 to 11.5 Mbps; thus creating a capacity allocation from 15% to 120% at the bottleneck link.

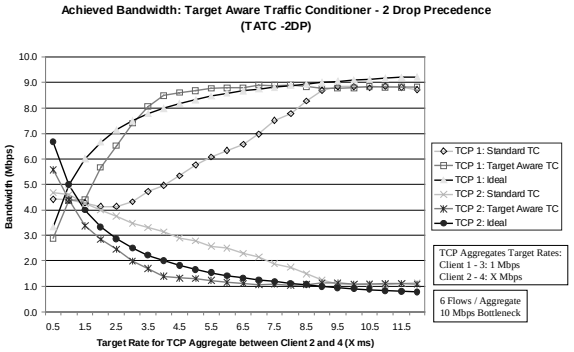

Figure 13: Achieved Bandwidth using TATC-2DP

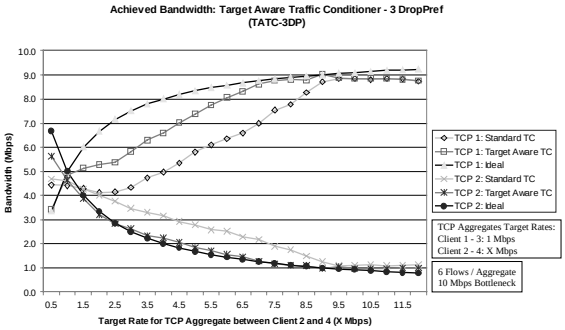

Figure 14: Achieved Bandwidth using TATC-3DP

Figure 13 shows the results of the experiment with standard TC and Target-Aware TC when the TATC-2DP algorithm is used. The expected bandwidth is also plotted. It is observed that there is a gap in the expected and achieved bandwidth when standard TC is used. The excess bandwidth is not proportionally distributed, as would be desired by a customer. Instead, we see an almost even distribution of the excess bandwidth between two sets of competing flows. When the Tar-

get-Aware TC is used, the achieved bandwidth is closer to the expected bandwidth for both the flow aggregates.

Similar results are shown in Figure 14 for the case when the TATC-3DP algorithm is used. In another experiment, six different flow aggregates with different target rates are pushed through bottleneck links of 45 Mbps and 22 Mbps. The total allocated target rate constitutes 40 % and 80% of the bottleneck link capacity respectively. The experiment is performed with standard TC and TATC-3DP. Figure 15 shows the achieved bandwidth for all aggregates in the case of the 45 Mbps bottleneck link. Figure 16 shows the achieved bandwidth for all the aggregates in case of 22 Mbps bottleneck link. It is seen that improvement in bandwidth allocation is significant for heavily over-provisioned network. The experiment was repeated for the TATC-2DP and the results closely resemble those in Figure 15 and 16.

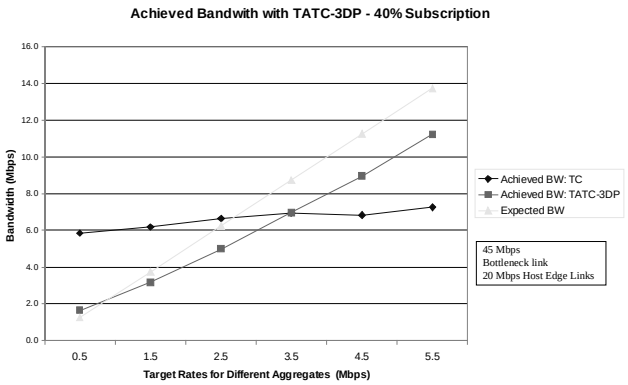

Figure 15: TATC-2DP – 40% Capacity Allocation

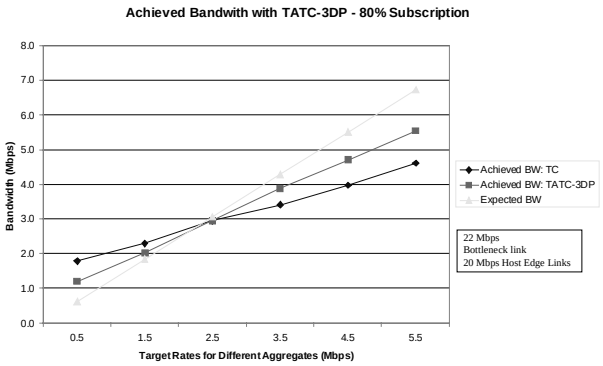

Figure 16: TATC-3DP – 80% Capacity Allocation

Table 2 reflects the extent to which the different Target-Aware TCs are able to achieve the expected bandwidth based on proportional distribution of the excess. The table shows the average deviation from expected value achieved by each traffic conditioner in each of the three experiments performed in this section.

For all three experiments, it can be concluded that the standard TC has a higher percentage deviation from expected results than either of the TATC algorithms. The performance of TATC-2DP and TATC-3DP are comparable.

Table 2: Percentage Average Deviation from Expected Results.

TC	Figures	Figures 15 & 16	
	13 & 14	80 % Provisioned	40 % Provisioned
Standard	0.32	0.50	0.91
TATC-2DP	0.11	0.23	0.19
TATC-3DP	0.08	0.25	0.21

7. Discussion

The previous sections have presented various methods for ensuring a fairer distribution of bandwidth for flows in an AF-based Diffserv network. In this section, we evaluate the applicability of the proposed solutions and identify the limitations.

RTT-Aware TC

The RTT-Aware TC has the following requirements. Firstly, it is applicable for traffic streams where all flows in the aggregate have the same RTT. Secondly, it requires the edge devices to determine the RTT of aggregates passing through it. One possible way to do this is to consider a

single flow as representative of the aggregate. The edge device can perform RTT measurement of the aggregate traffic at the edge of the network. This will require per-flow state monitoring of data packets and observing the return of corresponding ACKs in the reverse direction. Such a scheme assumes that the delay from the edge to the host is minimal.

The third requirement is to determine the minimum RTT for aggregates in the network. Two approaches are possible. If queueing delay at core devices is minimal then for pre-configured point-to-point connections, the RTT can be estimated based on the transmission delay of intermediate links. If, however, queueing delay is a major component in the RTT, then the RTT needs to be dynamically measured. Thus, to determine the minRTT, the edge nodes need to exchange RTT information co-operatively.

The analysis of the previous section showed that with a large number of active flows, the engineering of RED parameters is also important to bandwidth assurance for Assured Forwarding based services. However, a setting of large *maxth* is impractical since it will cause a large and variable end-to-end packet delay. Alternatively, the number of active flows at the core of the network can be limited. This will require admission control at the edge of the network.

TCP/UDP Interaction

The TCP/UDP studies (Figure 10) show that using drop precedence mapping, a certain level of fairness can be achieved. Mapping TCP and UDP in-profile traffic to DP0 helps to achieve the target bandwidth. However, mapping of TCP and UDP out-of-profile traffic to different drop precedence is necessary to handle the bandwidth distribution at over-provisioned and under-provisioned states. Scenario 6 satisfies the required mapping and it is reflected in low percentage deviation in fairness index (Figure 10). As shown in Figure 10, use of two queues to isolate TCP and UDP traffic provides the optimum solution. However, the approach has a possible drawback due to the necessity of knowing the fraction of TCP and UDP target rates at the core of the network. This requirement can be handled by the use of bandwidth broker - to communicate target rates - or by pre-allocating weights for each queue based on an estimate of UDP and TCP traffic.

Target-Aware TC

It is debatable whether the excess bandwidth in an over-provisioned network should be divided among aggregates in proportion to the subscribed target rates or should be divided equally. This is a business decision that shouldn't be influenced by technical limitations. Should providers wish to offer a proportional distribution of the excess, they should have the building blocks to do so at their disposal. The TATC-2DP and TATC-3DP are two examples of such building blocks.

Although the performance results of the TATC-2DP and the TATC-3DP are comparable, there are practical issues to consider when evaluating a Target-Aware TC. The TATC-2DP will increase the amount of in-profile traffic in the network. This makes traffic engineering more difficult because the total in-profile traffic cannot be estimated from the subscribed total target-rates. On the contrary, the TATC-3DP has in-profile traffic that is consistent with the target rates since excess traffic is partitioned between DP1 and DP2.

Both the TATC schemes require knowledge of the minimum Target Rate in the network. This is not as difficult to obtain as the minimum RTT in the network. Target Rates are typically static and don't change as often as RTT. Thus, the minimum Target Rate can be periodically determined via the existing policy management framework and communicated to the edge devices using a COPS-like protocol.

Other Issues

For over-provisioned networks, the target rates of aggregated flows are mostly achievable. As we have seen, the degree to which the excess bandwidth is fairly distributed depends on various factors. Diffserv SLAs will likely contain some form of quantitative guarantee on performance

parameters such as bandwidth. It has been shown that various factors play an important role in determining the bandwidth obtained by TCP and UDP flows. As such, it is important to develop scalable approaches to ensure fair distribution of bandwidth that can be guaranteed in accordance with SLA performance parameters. Though these parameters cannot be exact our work has shown that technical solutions are feasible to provide some guarantee with certain constraints.

It is important to understand the scope (i.e., topological extent) of services for which some form of quantitative assurance can be given. Various traffic conditioning schemes may be feasible for all traffic between an ingress point and an egress point or a set of egress points. Further study is needed to address the scalability issues as the number of egress points increase. Developing traffic conditioners for a one-to-anywhere topology requires further work.

8. Conclusions

The contributions of this paper are the following: (a) An intelligent traffic conditioner to mitigate the impact of RTT on the achieved bandwidth for traffic aggregates with equal target rates; (b) Possible approaches to address the fairness issues between TCP and UDP traffic aggregates; (c) Two intelligent traffic conditioners to distribute the excess bandwidth in over-provisioned networks in proportion to the target rates.

The limitation of the above approaches are: (a) All the solutions assume one-to-one and one-to-few network topology (not one-to-any); (b) RTT-Aware and Target-Aware intelligent TCs are tied to TSW tagging algorithm. However, this can be extended to other tagging approaches as well; (c) Edge nodes have to communicate among themselves to obtain certain state information.

Appendix A

The throughput of TCP flows can be represented by: $R_A = \frac{C * MSS}{RTT * \sqrt{p}}$ (1)

where C is a constant, R_A is the measured throughput, MSS is the packet size, RTT is the round-trip time and p is the packet drop probability for that flow. The simplest objective of the RTT-Aware Traffic Conditioner is to ensure that two or more flows with the same target rate and packet size will obtain an equal share of the excess bandwidth regardless of their round-trip times.

Consider two flows with achieved rates R_A^1 and R_A^2 the objective is to obtain:

$R_A^1 = R_A^2$ (2)

If the packet sizes for the two flows are the same then the equation reduces to:

$RTT_1\sqrt{p_1} = RTT_2\sqrt{p_2}$ (3)

If the two flows have different round trip times, then:

$$\frac{p_2}{p_1} = \left(\frac{RTT_1}{RTT_2}\right)^2 \quad (4)$$

Therefore, if we desire the same achieved rate for the two flows, the ratio of their packet drop probabilities should have an inverse squared relationship to the round-trip times.

The TSW marker [3] operates as follows: if the measured rate of a flow is beyond its target rate, it marks packets out with the probability p_{out}:

$$p_{out} = q = \frac{R_A - R_T}{R_A} \quad (5)$$

We make the following assumption:

$$\frac{p_{out}^1}{p_{out}^2} \propto \frac{p_1}{p_2} \quad (6)$$

That is we assume that the ratio of out-of-profile packet marking is directly proportional to the ratio of packet drop probabilities at the core. This is true because packets from both flows end up being counted towards the same average queue calculations in the core

Therefore, for two flows at the edge,we modify the out-of-profile marking scheme as follows:

$$p_{out}^1 = q \quad \text{and} \quad p_{out}^2 = \left(\frac{RTT_1}{RTT_2}\right)^2 q \quad (7)$$

If there was a third flow, it would be marked out-of-profile with probability: $p_{out}^3 = \left(\frac{RTT_1}{RTT_3}\right)^2 q$

(8)

Based on the above, the marking scheme can be extended and generalized to be applicable to *n* numbers of flows passing through the edge.

The generalized marking scheme for out-of-profile packets would be: $p_{out}^i = \left(\frac{RTT_{\min}}{RTT_i}\right)^2 q \quad (9)$

where RTT_i is the round-trip time for that particular flow and RTT_{min} is the minimum round-trip time of all the flows in the network.

This derivation has been developed for the case where each TCP flow has its own target rate. However the derivation can be extended to be applicable for TCP aggregates as long as all the flows in the aggregate have the same RTT.

9. References

[1] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance ", *IEEE/ACM Transactions on Networking*, V.1 N.4, August 1993, p. 397-413.

[2] Blake, S. Et al, "An Architecture for Differentiated Services", RFC 2475, December 1998

[3] Clark D. and Fang W., "Explicit Allocation of Best Effort Packet Delivery Service", *IEEE/ACM Transactions on Networking*, V.6 N. 4, August, 1998

[4] Ibanez J, Nichols K., "Preliminary Simulation Evaluation of an Assured Service", Internet Draft, draft-ibanez-diffserv-assured-eval-00.txt>, August 1998

[5] Heinanen J., Baker F., Weiss W., and Wroclawski J., "Assured Forwarding PHB Group", RFC 2597, June 1999.

[6] Jacobson V, Nichols K, Poduri K, "An Expedited Forwarding PHB", RFC 2598, June 1999.

[7] Lin W, Zheng R and Hou J, "How to Make Assured Services More Assured", *In Proceedings of ICNP*, Toronto, Canada, October 1999.

[8] Yeom, I and Reddy N, "Realizing throughput guarantees in a differentiated services network", *In Proceedings of ICMCS*, Florence, Italy, June 1999.

[9] Mathis M, Semske J, Mahdavi J, Ott J, "The macroscopic behaviour of the TCP congestion avoidance algorithm.", *Computer Communication Review*, 27(3), July 1997

[10] Morris, R., "TCP Behavior with Many Flows", *In Proceedings of IEEE International Conference on Network Protocols*, October 1997, Atlanta, Georgia

[11] Seddigh, N., Nandy, B., Pieda, P, “Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network”, *In Proceedings of Globecom’99*, Rio De Janeiro, December 1999.

[12] Yeom, I and Reddy N, "Impact of marking strategy on aggregated flows in a diff-serv network," *In Proceedings of IWQoS’99*, London

[13] Elloumi O, De Cnodder S and Pauwels K, “Usefulness of three drop precedences in Assured Forwarding Service”, <draft-elloumi-diffserv-threevstwo-00.txt>, Internet Draft, July 1999.

[14] Kim H, “A Fair Marker”, <draft-kim-fairmarker-diffserv-00.txt>, Internet draft, April 1999

[15] Network simulator (ns-2), University of California at Berkeley, CA, 1997. Available via http://www-nrg.ee.lbl.gov/ns/.