Sustaining TCP Throughput Using Assured Forwarding and ECN in a Differentiated Services Network[1]

Ivan Kai-Kin Leung and Jogesh K. Muppala
Dept. of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Tel.: +852 2358 6978
Fax: +852 2358 1477
Email: muppala@cs.ust.hk

Abstract

This paper examines the interaction among the congestion control mechanisms built into the Transmission Control Protocol (TCP), the Assured Forwarding (AF) PHB of Differentiated Services (Diffserv) mechanism, and Explicit Congestion Notification (ECN) mechanism. We show that ECN improves the fairness in sharing excess bandwidth among different AF flows. We then propose two mechanisms: (a) Out marking scheme, and (b) throughput sustaining scheme to foster cooperation between ECN and AF mechanism to further improve the throughput of TCP sources. Our results show that the new schemes achieve significant improvement in the TCP throughput performance.

1. Introduction

The Transmission Control Protocol (TCP) is the most widely used transport protocol for reliable data transfer in the Internet. Any effort to improve the performance of TCP can have significant impact on the performance of the overall Internet [14]. TCP has several built-in congestion control mechanisms to respond to signals from the network about congestion. Several new mechanisms for congestion control in TCP have recently been proposed [6].

Explicit Congestion Notification (ECN) [5][15] is one such mechanism of interest to us in this paper. ECN is aimed at improving the performance of TCP for low-bandwidth delay-sensitive TCP traffic. This mechanism, if implemented, separates the policies of queueing and dropping packets from the policies for indicating congestion [15].

Another recent trend in networking is towards scalable support for quality of service (QoS) requirements of Internet applications. The Differentiated Services (Diffserv) approach [1] is proposed as a scalable mechanism for QoS support. The Assured Forwarding (AF) PHB provides a mechanism to support different forwarding assurances to different customers based on their profile.

The major aim of this paper is to examine the performance of TCP sources using the AF PHB to provide throughput assurances and the effect of using ECN in this environment to improve the throughput. The sources express their throughput requirements in terms of a target rate. The performance studies are conducted using the network simulator *NS-2* [16] on a simple network topology with a bottleneck link. We show that ECN improves the fairness in sharing excess bandwidth among different AF flows. We then propose two mechanisms: (a) Out marking scheme, and (b) Throughput sustaining scheme to foster cooperation between ECN and AF mechanism to further improve the throughput of TCP sources. Our results show that the new schemes show significant improvement in the TCP throughput performance in a Diffserv environment

This paper is organized as follows. First, we present some background on Diffserv and ECN and motivation for our study in Section 2. Then we present the network topology and performance metrics that are used in the experiments presented in this paper in Section 3. The interaction between Diffserv and ECN is discussed in Section 4. The new mechanisms to improve the cooperation between Diffserv and ECN are discussed in Section 5. Finally we present the conclusions and future research directions in Section 6.

[1] This work described in this paper has been supported by the Research Grants Council of Hong Kong, China
(Project no. HKUST6099/99E and Project no. DAG00/01.EG10)

2. Background and Related Work

The DiffServ architecture distinguishes edge routers from core routers. It only requires edge routers to maintain fine granularity states, e.g., per flow state. The core routers treat the traffic as an aggregate. DiffServ defines a limited set of Per-Hop Behaviors (PHBs) and implements them in core routers. The edge routers classify packets into flows, and apply traffic conditioning mechanisms on the packets – including *metering*, *tagging*, *shaping* and *policing*. The packets are tagged with some patterns in the DS field [13], which indicate to the core routers the PHBs that should be applied to the packets. The core routers do not need to classify packets but treat packets as an aggregate, and apply corresponding PHBs. The service provided to a particular packet stream is a combination of the traffic conditioning at edge routers and a series of PHBs in core routers. With different PHBs and different traffic conditioning mechanisms, DiffServ can provide differentiated services to traffic. By pushing the complexity to the edge and maintaining a simple core, DiffServ achieves scalability.

Per-Hop Behavior (PHB) defines the forwarding treatment accorded to a packet when it enters the DS domain. Two PHBs that have been proposed are Assured Forwarding (AF) [8] and Expedited Forwarding (EF) [9]. EF PHB provides low loss, low latency, low jitter and assured bandwidth. AF PHB allows a service provider to provide different levels of forwarding assurances according to the profiles of the customers. In AF service framework, the edge routers monitor and mark packets of flows (individual or aggregated). The packets of a flow that obey the service profile are marked IN (In-profile) and packets that are beyond the service profile are marked OUT (Out-of-profile). The network gives preference to IN packets while dropping OUT packets disproportionately at the time of congestion. The router does not distinguish between packets of individual flows and can use FIFO style scheduling mechanisms. This preferential drop mechanism is expected to provide better throughput for IN packets than OUT packets.

Several researchers have considered the performance of TCP sources over Diffserv networks [2][3][4][17][18]. Of particular interest to us is the work presented in [3][4][17][18]. While [3][4] focused on individual TCP flows, [17][18] focused on aggregate sources and specifically on techniques to achieve specific performance goals of individual flows within an aggregation while adhering to the service contracts. It has been shown through simulation [4][17][18] that flows with higher reservation may realize less throughput than their target rates, while flows with smaller reservation may easily achieve not only their target rates but exceed them. Our results presented later also agree with this observation.

In this paper, we consider the performance of TCP sources using AF PHB over Diffserv networks. We show that using ECN, each source not only achieves its target rate, but also gets a fair share of the remaining bandwidth. We consider three different ECN marking strategies, viz., the *mark-tail* (the packet just joining the queue is marked) [5], the *mark-front* (the packet at the front of the queue is marked) [12] and the *mark-random* (one packet from the queue is picked at random to be marked) [11] strategy. We show the performance improvement achieved by the different marking strategies. From the results we observe that a tighter cooperation between ECN and Diffserv may further improve the performance. A Diffserv aware ECN marking strategy is better at achieving a fair sharing of the excess bandwidth.

3. Experimental Setup

3.1. Network Topology

In this section, two network topologies that we use in our performance studies will be discussed. We examine the performance of the system containing TCP connections with different target transmission rates in the first topology. In the second topology, we examine a system containing TCP connections with different RTTs.

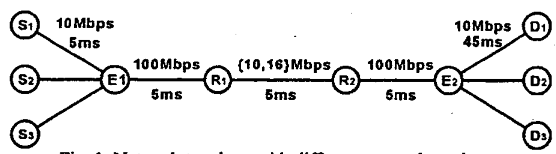


Fig. 1: Network topology with different target throughputs

Our first topology, shown in Fig. 1, is a dumbbell topology with a shared bottleneck link. Two values for the bandwidth of the shared bottleneck link viz. 10 Mbps and 16 Mbps are considered. The bandwidths of the other links are shown in Fig. 1. The three TCP sources S_1 to S_3 communicate with their corresponding destinations D_1 to D_3 respectively. E_1 and E_2 are edge routers in the Diffserv domain. Their major responsibility is to mark the packets from the sources as IN or OUT packets according to the transmission rates of the connections and the profiles of the sources. R_1 and R_2 are core routers. They are ECN capable routers with RED with in/out (RIO) mechanism [2]. When congestion occurs, the routers will drop or mark (ECN marking) the packet according to the DiffServ code point (IN or OUT) marked in the packets and the parameters of the RIO queue. The queue size is 500 packets. In our experiments, the threshold parameters for IN packets are [100,150,0.02,0.1] and that for OUT packets are [50,100,0.1,0.1]. The RTTs of the three connections are set to 130 ms.

TABLE 1
TARGET THROUGHPUTS OF THE CONNECTIONS

Connections	Target Throughput
S_1-D_1	1 Mbps
S_2-D_2	4 Mbps
S_3-D_3	2 Mbps

The workload in the network is generated by long-term FTP sources using either all non-ECN or all ECN capable TCP with different marking strategies. These FTP/TCP sources are characterized by the packets size that is set to 1000 bytes in the experiments. The duration of the experiments is 60 seconds. The target transmission rates (target throughputs) of the source-destination pairs are listed in Table 1.

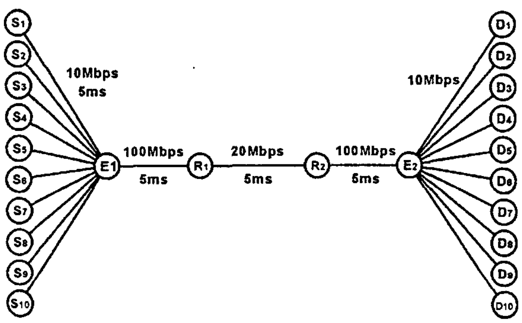


Fig. 2: Network topology with different RTTs

TABLE 2
RTTs OF THE CONNECTIONS

Connections	RTTs of Set #1 (ms)	RTTs of Set #2 (ms)
S_1-D_1	94	58
S_2-D_2	102	74
S_3-D_3	110	90
S_4-D_4	118	106
S_5-D_5	126	122
S_6-D_6	134	138
S_7-D_7	142	154
S_8-D_8	150	170
S_9-D_9	158	186
S_{10}-D_{10}	166	202

The second topology, which is similar to the first topology, is shown in Fig. 2. The number of connections in this topology is 10 (S_1-D_1 to S_{10}-D_{10}). The settings and functions of routers E_1, E_2, R_1 and R_2 are the same as before. The target throughputs of the connections are set to 1 Mbps. In the experiments, two sets of RTTs listed in Table 2 are used for generating two sets of results to evaluate the effect under different conditions. These sets of RTTs are also used in [17].

3.2. Performance Metrics

The performance metrics used in the experiments are the throughput, excess bandwidth claimed and the *Fairness Index on Excess Bandwidth (FIEB).*

Since the bandwidth (BW) of the bottleneck link is greater than the sum of the target throughputs of the connections, the connections are expected to claim more bandwidth than their target throughputs. The difference between the actual throughput and the target throughput of a connection is called *Excess BW* of that connection.

To evaluate how the connections share the excess BW of the system, the *Fairness Index on Excess BW (FIEB)* is used. Suppose the actual throughputs and the target throughputs of the connections are denoted as P_i and R_i where i $(1 \le i \le m)$ is the connection number of S_i-D_i. Modifying the fairness index defined by Jain [10], we define the *FIEB* as:

$$FIEB = \frac{(\sum_{i=1}^{m}(P_i - R_i))^2}{m\sum_{i=1}^{m}(P_i - R_i)^2}$$

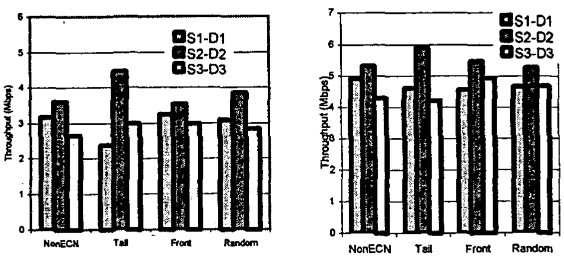
Similar to Jain's fairness index, *FIEB* is also a number between 0 and 1, i.e. $0 \le FIEB \le 1$.

4. Interaction Between ECN And Diffserv

In this section, we examine the interaction between ECN and Diffserv in a Diffserv network. Diffserv inherits the problems of global synchronization and bias against connection with long RTT of TCP connections [17][18]. ECN can improve the aggregate throughput, fairness and stability for TCP flows in a simple best-effort network [5][11]. It is reasonable to expect improvement in TCP performance in a DiffServ environment by adding ECN mechanism. We examine the performance for different ECN marking strategies.

4.1. Results

We first examine the performance of TCP connections with different target throughputs sharing a bottleneck link as in Topology 1. Two sets of results are obtained for bottleneck link bandwidths of 10 Mbps and 16 Mbps.

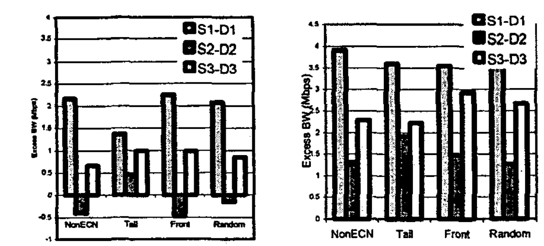


(a) BW = 10 Mbps (b) BW = 16 Mbps

Fig. 3: Throughputs of connections with different target throughputs

The throughputs of the connections are shown in Fig. 3. In this figure, four groups of bars show the throughputs for *non-ECN*, *mark-tail*, *mark-front* and *mark-random* ECN marking strategies.

We observe that under both scenarios (10 Mbps and 16 Mbps bottleneck link), *mark-tail* shows better performance in maintaining the throughputs closer to the target throughputs. For *non-ECN*, *mark-front* and *mark-random* strategies, the connection with lowest target throughput (1 Mbps) is able to claim more bandwidth than others. In both graphs, the throughputs of the three connections for *mark-front* and *mark-random* are nearly the same. This is because these strategies try to balance the sharing of bandwidth equally for each connection without considering the target throughput of the connections. This is expected since the fairness evaluation presented in [11] shows that *mark-front* and *mark-random* perform the best in terms of fairness among the connections.

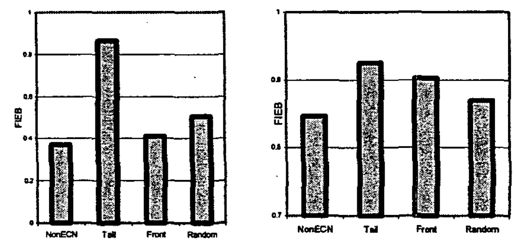


(a) BW = 10 Mbps (b) BW = 16 Mbps

Fig. 4: Excess BW of connections with different target throughputs

Fig. 4 presents the excess BW values for the connections. Negative values indicate that the actual throughput is lower than the target throughput requirement. The graphs show that only *mark-tail* is able to satisfy the target throughput requirement. The results also indicate that the connection with the largest target throughput requirement (S_2-D_2) only receives the smallest share of the excess bandwidth. On the other hand, the connection with the smallest target throughput (S_1-D_1) receives the largest share of the excess bandwidth. Similar results were reported earlier in [3][4][17][18].

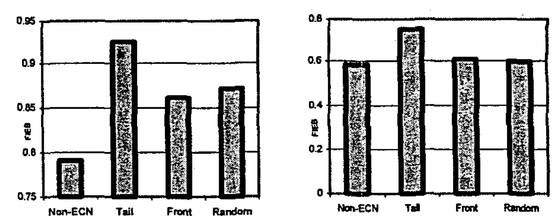
Fig. 5 shows the *FIEB* for different marking strategies. The three ECN marking strategies (*mark-tail*, *mark-front* and *mark-tail*) show better performance than *non-ECN*. It shows that the ECN mechanism improves the fairness in sharing of excess bandwidth in DiffServ domain. *Mark-tail* strategy shows the best performance, while *mark-front* and *mark-random* do not yield significant improvement.



(a) BW = 10 Mbps (b) BW = 16 Mbps

Fig. 5: FIEB of connections with different target throughputs

The next set of results, shown in Fig. 6, gives the performance of TCP connections with different RTT with the topology shown in Fig. 2. From the results in Fig. 6(a) we see that the three ECN marking strategies improve the fairness significantly. Among the marking strategies, *mark-tail* performs the best (16% improvement compared with *non-ECN*) while *mark-front* and *mark-random* perform nearly the same with about 9% improvement.



(a) RTT range: 96-166 ms (b) RTT range: 58-202 ms

Fig. 6: FIEB of connections with different RTTs

In Fig. 6(b), with larger RTT range among the connections, the performance of *mark-tail* is still the best among all the strategies. The *non-ECN*, *mark-front* and *mark-random* perform nearly the same although the fairness indexes show that *mark-front* and *mark-random* are slightly better.

The result shows that *mark-front* and *mark-random* strategies are unable to mark the *right target* in this situation; so they behave like a *non-ECN* system. Because of the randomness of the packet marking of the *mark-*

random strategy, the packets from the connection with higher throughput (usually with higher target rate) will usually be marked. That connection will be forced to reduce its transmission rate no matter whether the marked packet is an IN or an OUT packet. This mechanism favors the connections with lower target throughputs. It explains why the *mark-random* and *mark-front*, being fair strategies, are unable to give good performance.

5. Integration Of ECN And Diffserv

The results presented in the previous section where the ECN mechanism is adding into DiffServ network show that not all marking strategies give improvement in the performance of TCP flows. Although *mark-tail* strategy can provide better performance than the *non-ECN* environment, *mark-front* and *mark-random* perform nearly the same as the *non-ECN*. However, *mark-front* and *mark-random* are expected to perform better than *mark-tail*.

The main reason for the problem is that ECN and DiffServ work independently. DiffServ uses the edge routers for marking purpose (IN and OUT packets) and core router for maintaining the target transmission rate of different connections. On the other hand, ECN mechanism works only at the core router to control the network congestion. ECN marking mechanism does not take into account the DiffServ code point (IN or OUT) marked at the edge router. In this case, the connections with different profiles are treated the same from the ECN point of view. This situation conflicts with the concept of DiffServ whose aim is to provide different treatment for different connections according to their profiles. As a result, simply adding ECN mechanism into DiffServ network may not always yield better performance. To enhance the cooperation between DiffServ network and ECN mechanism, intelligent ECN marking schemes should be applied. In the next section, two such schemes are proposed.

5.1. Improving Cooperation between Diffserv and ECN

In this section, two new schemes, *OUT Marking Scheme* and *Throughput Sustaining Scheme*, are proposed to enhance the cooperation between ECN and DiffServ. The detailed descriptions of the schemes are presented below.

OUT Marking Scheme

In order to make sure that ECN can work well with DiffServ, the DiffServ code points (IN or OUT) of the packets should be considered when ECN marking takes place. In a RIO queue, there is only one physical queue but this queue can be conceptually divided into IN queue and OUT queue containing the IN packets and OUT packets respectively. The edge router in the DiffServ domain marks the packet as OUT packet only when the transmission rate of the connection exceeds its target transmission rate. Therefore, OUT packet is an indication of already satisfying the target throughput requirement and implying that the connection is trying to claim more bandwidth than its target rate. In this case, dropping/marking OUT packets will not affect the target throughput of the connections.

Therefore we propose the *OUT Marking Scheme*. The main idea of the scheme is to do ECN marking by considering the DiffServ code point of the packet. When the RIO queue decides to drop/mark a packet, OUT packets are preferentially dropped/marked over IN packets. To perform the ECN marking, a packet will be selected from the OUT queue based on the ECN marking strategies (*mark-tail*, *mark-front* and *mark-random*) suggested earlier. If the OUT queue is empty, then IN packets will be selected. A summary of selecting the packet using different marking strategies is shown in Table 5.

TABLE 5
SUMMARY OF OUT MARKING SCHEME

	Non-ECN	Mark-tail	Mark-front	*Mark-random*
OUT queue is not empty	Packet at the tail of OUT queue is dropped	Packet at the tail of OUT queue is marked	Packet at the head of OUT queue is marked	Packet is selected at random to mark from the OUT queue
OUT queue is empty	Packet at the tail of IN queue is dropped	Packet at the tail of IN queue is marked	Packet at the head of IN queue is marked	Packet is selected at random to mark from the IN queue

Throughput Sustaining Scheme

In the DiffServ architecture, bandwidth allocation is based on service profiles. The underlying premise is that each entity is assured of its target throughput specified in its service profile when congestion is experienced and can exceed such profiles when there is no congestion. In the DiffServ domain, when a TCP connection loses a packet, the TCP connection will reduce its sending rate. With the *additive increase/multiplicative decrease* mechanism of TCP [1], the reduction of the sending rate always causes the throughput to fall below the target throughput as the TCP congestion control mechanism reduces the sending rate at least by half of the current one. This causes the average operating throughput of TCP connection unable to meet the service profile.

In order to provide better guarantee of satisfying the target throughput, the *Throughput Sustaining Scheme* is proposed. A similar scheme was suggested in [3]. The purpose of the scheme is to sustain the target throughput of the TCP connections by setting the lower bound of the sending rate to the target throughput stated in its service profile. At the TCP source, it estimates the instantaneous sending rate by using the *estimated RTT* and *congestion window* size of the TCP connection as follows:

$$SendingRate = \frac{CongestionWindow}{RTT}$$

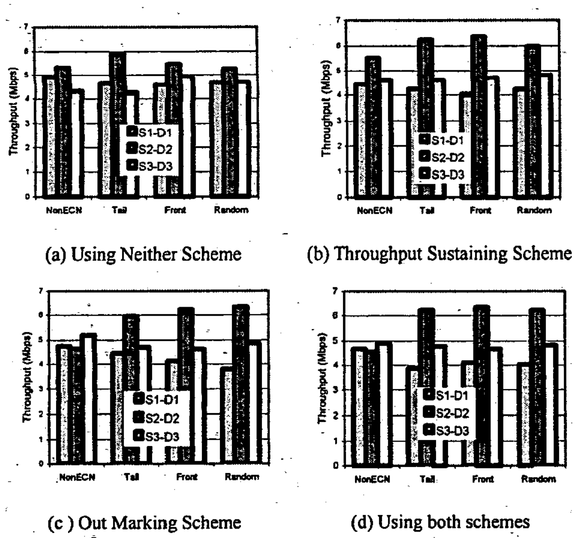
The *Throughput Sustaining Scheme* can be summarized as:

If (Adjusted Sending Rate < Target Throughput)
  Current Sending Rate = Target Throughput
Else
  Current Sending Rate = Adjusted Sending Rate

Here the Adjusted Sending Rate is the result after adjusting the congestion window. The target throughput provides a lower bound on how low the congestion window can be adjusted.

5.2. Results

We now examine the performance of the network by adding one or both of the above schemes. In the first experiment we use the topology in Fig. 1 with 16 Mbps bottleneck link. There are 3 TCP connections with the target rates of 1 Mbps, 4 Mbps and 2 Mbps sharing the bandwidth of the bottleneck link.

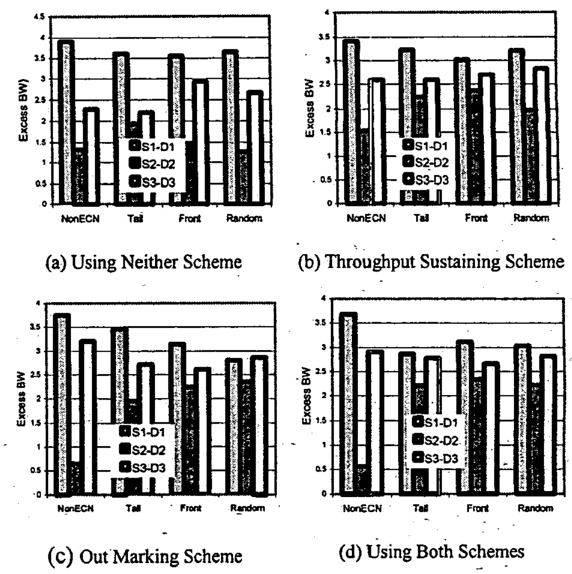


(a) Using Neither Scheme (b) Throughput Sustaining Scheme

(c) Out Marking Scheme (d) Using both schemes

Fig. 7: Throughputs of connections with different target rates and new schemes

Fig. 7 shows the throughputs of connections with different target rates. Since the bottleneck link bandwidth is 16 Mbps and the target throughputs of the connections are 1 Mbps, 4 Mbps and 2 Mbps, the excess bandwidth of the system is 9Mbps (16Mbps-7Mbps). In perfect fair situation, each connection should be able to get an excess bandwidth of 3Mbps such that their expected throughputs should be 4 Mbps, 7 Mbps and 5 Mbps respectively.



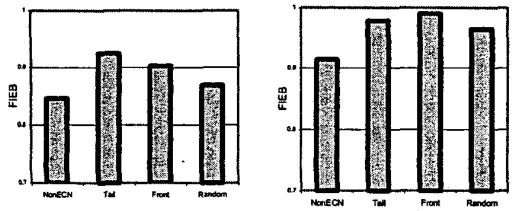
(a) Using Neither Scheme (b) Throughput Sustaining Scheme

(c) Out Marking Scheme (d) Using Both Schemes

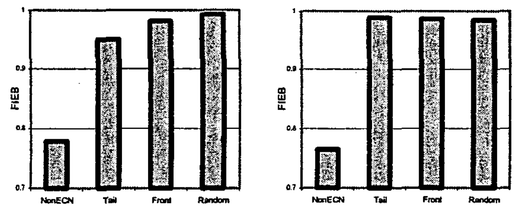
Fig. 8: Excess BW of connections with different target throughputs and new schemes

From Fig. 7 we see that the *Throughput Sustaining Scheme* and *OUT marking scheme* provide better control on individual throughputs than the system without either schemes. Although the results of these two schemes are not perfectly fair, the schemes show their intention to distribute the excess bandwidth equally to the connections. Both schemes show similar results no matter which ECN marking strategy is used. Fig. 7(d) shows the results using both schemes. The combination of both schemes yields similar result as that using only one of the schemes.

Fig. 8 shows the excess bandwidth using different ECN marking strategies. In the perfectly fair situation, all three bars in a group should be at the same value of 3 Mbps. The new schemes show marked improvement for both *mark-front* and *mark-random* ECN marking strategies.



(a) Using Neither Schemes (b) Throughput Sustaining Scheme



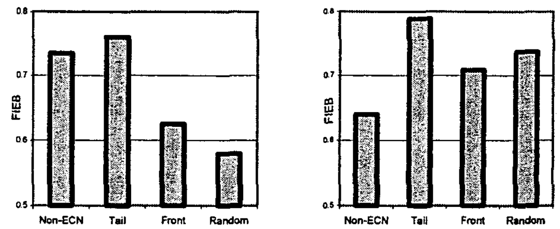
(c) Out Marking Scheme (d) Using Both Schemes

Fig. 9: FIEB of connections with different target throughputs and new schemes

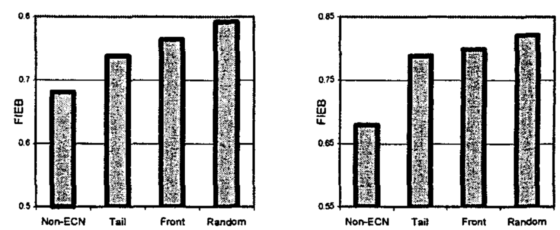
Fig. 9 shows the *FIEB* performance of the schemes. Without applying any of the schemes, the ECN mechanism can only provide the *FIEB* around 0.85 to 0.92. The three marking strategies show similar results in all the cases. The OUT Marking Scheme can achieve the *FIEB* of at least 0.94 while the Throughput Sustaining Scheme can achieve the *FIEB* of at least 0.95. By using both schemes at the same time, we can achieve a *FIEB* value of 0.97. The schemes seem to solve the cooperation problem between ECN and DiffServ and work very well with the *mark-random* marking strategy.

The results for the "58-202ms RTTs" range is presented in Fig. 10. There are 10 TCP connections with the target throughputs 1 Mbps and different RTTs (see table 4) sharing the bandwidth (20Mbps) of the bottleneck link.

Our last set of results are obtained by applying the proposed schemes to the topology in Fig. 2. The *FIEB* results of the connections with different RTTs using the combinations of proposed schemes are shown in Fig. 10. Without applying any schemes (shown in figure Fig. 10 (a)), *mark-tail* strategy performs the best while *mark-random* strategy performs the worst. The problem may be due to the fact that the fairness power of *mark-random* does not consider the DiffServ requirement on providing different services to different connections. However, by applying the new schemes on the *mark-random* strategy, it yields the best result (near 0.8) when OUT Marking Scheme is used (shown in Fig. 10 (c)) and shows a great improvement when *mark-random* strategy works with Throughput Sustaining Scheme (shown in Fig. 10 (b)).



(a) Using Neither Schemes (b) Throughput Sustaining Scheme



(c) Out Marking Scheme (d) Using Both Schemes

Fig. 10: FIEB of connections with different RTTs with new schemes

From the graphs, it is observed that the different combinations of the proposed schemes do not affect the performance of *mark-tail* strategy. *Mark-random* works very well with OUT Marking Scheme and Throughput Sustaining Scheme. It is observed that only the *mark-random* strategy working with both proposed schemes can achieve a *FIEB* value over 0.8.

6. Conclusions

In this paper we examined the interaction among the congestion control mechanisms built into the Transmission Control Protocol (TCP), the Assured Forwarding (AF) PHB of Differentiated Services (Diffserv) mechanism, and Explicit Congestion Notification (ECN) mechanism. We showed that ECN improves the fairness in sharing excess bandwidth among different AF flows. We found some problems with the combination of ECN and Diffserv, if they are unaware of the application of the other. We then proposed two mechanisms: (a) Out marking scheme, and (b) throughput sustaining scheme to improve cooperation between ECN and AF mechanism to further improve the throughput of TCP sources. Our results show that the new schemes show significant improvement in the TCP throughput performance in a Diffserv environment.

7. References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated

Services," *RFC2475,* Network Working Group, December, 1998.

[2] D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," *ACM Transactions on Networking*, vol. 6, no. 4, pp. 362-373, August 1998.

[3] W. Feng, D. D. Kandlur, D. Saha and K. G. Shin, "Understanding and Improving TCP Performance over Networks with Minimum Rate Guarantees," *IEEE/ACM Trans. on Networking*, vol. 7, no. 2, pp. 173-187, Apr. 1999.

[4] W. Feng, D. D. Kandlur, D. Saha and K. G. Shin, "Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated Services Internet," *IEEE/ACM Trans. on Networking*, vol. 7, no. 5, pp. 685-697, Oct. 1999.

[5] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communications Review*, vol. 24, no. 10, pp. 8-23, October 1994.

[6] S. Floyd, "A Report on Recent Developments in TCP Congestion Control," *IEEE Comm. Mag.*, vol. 39, no. 4, pp. 84-90, Apr. 2001.

[7] T. Hamann and J.Walrand, "A New Fair Window Algorithm for ECN Capable TCP (New-ECN)," *Infocom 2000*, Mar. 2000, pp. 1528-1536.

[8] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assurred forwarding PHB group," *Internet RFC 2597*, June 1999.

[9] V. Jacobson, K. Nichols and K. Poduri, "An Expedited Forwarding PHB," *Internet RFC 2598*, June 1998.

[10] R. Jain, *The Art of Computer Systems Performance Analysis*, Wiley, 1991.

[11] I. K. K. Leung, and J. K. Muppala, "Effect of Different Marking Strategies on Explicit Congestion Notification (ECN) Performance," *Proc. IEEE International Conference on Communications (ICC 2001)*, Helsinki, Finland, 11-15 Jun. 2001.

[12] C. Liu and R. Jain, "Delivering Faster Congestion Feedback with the Mark-Front Strategy," *International Conference on Communication Technologies (ICCT 2000),* Beijing, China, August 2000.

[13] 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*, IETF December 1998

[14] J. Padhye and S. Floyd, "On Inferring TCP Behavior," *Proc. SIGCOMM 2001.*

[15] K.K. Ramakrishnan and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP," *IETF RFC 2481*, January 1999.

[16] UCB/LBNL/VINT Network Simulator – NS (version 2), http://www-mash.cs.berkeley.edu/ns/.

[17] I. Yeom and A. L. N. Reddy, "Marking for QoS Improvement," *Computer Communications*, vol.24, no.1, pp.35-50, Jan. 2001.

[18] I. Yeom and A. L. N. Reddy, "Modeling TCP Behavior in a Differentiated Services Network," *IEEE/ACM Trans. on Networking*, vol. 9, no. 1, pp. 31-46, Feb. 2001.