Journal of Industrial Technology, Kangwon Nat'l Univ.,Korea, No. 20, 2000.
산업기술연구(강원대학교 산업기술연구소 논문집), 제20권, 2000.

# Improvement of fairness between assured service TCP users in a differentiated service network*

# 차별화 서비스 망에서 보장형 서비스의 TCP 사용자들간 공정성 개선 방안

이 지 형** 정 충 교***
Lee, Ji-Hyoung Jeong, Choong-Kyo

## Abstract

To support Quality of Service (QoS) in the existing Internet, Differentiated Service (Diff-Serv) has been proposed. But, the unfairness between TCP connections remains as a serious problem not only in the conventional best-effort service Internet but also in new Diff-Serv network.

In this paper, we propose the Balancing Marker Algorithm (BMA) improving the fairness between individual connections of aggregated sources in a Diff-Serv network. This algorithm is based on the 3-level priority marking method.

We compared the 2-level packet priority marker with the Balancing Marker proposed in this paper. And we showed that the BMA improved the fairness and the throughputs between the individual connections with different delays in an aggregated source.

Keywords : Internet, Differentiated Service, RIO

## 1. INTRODUCTION

Differentiated Service (Diff-Serv) has been proposed [1] as a service model that can be applied to the Internet backbone network. Diff-Serv processes QoS not by per-flow but by per-aggregation of flows. The complex functions are done in edge routers (ERs) and core routers transfer packets, referring only the Diff-Serv (DS) region of the IP header. Therefore, it has no problem of scalability.

But, the unfairness between TCP connections with different delays remains as a serious problem not only in the conventional best-effort Internet [2] but also in Diff-Serv network [3]. When an aggregated source makes an agreement on the service level with a service provider, the individual connections in an aggregated source experience different delays and congestion situations. In this case, connections with small round-trip delays may use more tokens in a bucket than those with large round-trip delays. Therefore, packets from TCP connections with large round-trip delays are likely to be classified as the OUT profile-packets discarded when

* 본 논문은 정보통신부에서 주관하는 정보통신 우수시범학교 지원사업에 의하여 수행되었음
** 강원대학교 전자공학과 석사과정
*** 강원대학교 전자공학과 조교수, 공학박사

congestions occur. So the TCP connections of aggregated sources won't utilize the total reserved bandwidth fairly.

In this paper, we proposed the Balancing Marker Algorithm (BMA) improving the fairness between TCP connections of aggregated sources.

In this algorithm, we check if sending rates of individual connections are reached to the Basic Assured Bandwidth (BAB) in the 1st bucket. And we re-check the fairness between the individual connections in the 2nd bucket. Re-marking of the packets in the 2nd bucket depends on the current average sending rates of the individual connections.

The rest of the paper is organized as follows. Section 2 explains the Token Bucket Algorithm. In Section 3, we propose the new marking algorithm, BMA. Section 4 presents the simulation with the proposed algorithm. Section 5 concludes the paper and points to future work.

## 2. DIFFERENTIATED SERVICE

### 2.1 Introduction

In contrast to the Integrated Service architecture, Diff-Serv aims to differentiate forwarding services by per-aggregation of flows, i.e. reservations are done for several related flows, e.g. for all flows between two subnetworks. These resource reservation procedures can be done not only statically but also dynamically according to a Service Level Agreement (SLA), which is done by mutual agreement.

According to the Differentiated Services concept, IP packets are marked with different priorities. This can be done within the users end-system or router or by the ISP. Every router reserves a certain amount of resources (especially bandwidth) for every service class.

And, to assign the different priorities to packets, we can use two schemes, namely, two and three drop precedence.

In two drop precedence, a packet of a flow is marked IN if the temporal sending rate at the arrival time of the packet is within the contract profile of the flow. Otherwise, the packet is marked OUT. The temporal sending rate of a flow is measured using TSM (Time Sliding Window) or a token bucket controller.

And, in three drop precedence, if the current sending rate is less than the reservation rate for green, the packet is marked as green. If the sending rate is greater than the reservation for green but less than the reservation for yellow, the packet is marked as yellow. Otherwise, the packet is marked as red.

Traffic conditioning including marking is done in a part so called traffic conditioner.

### 2.2 Traffic Conditioning

Traffic conditioning may include the following elements; classifier, meter, marker, shaper, dropper [4].

Packet classifiers select packets in a traffic stream based on the content of some portion of the packet header. There are two types of classifiers. The Behavior Aggregate (BA) classifier classifies packets based on the DS codepoint only. The Multi-Field (MF) classifier selects packets based on the value of a combination of one or more header fields, such as source address, destination address, DS field, protocol ID, source port and destination port numbers and etc.

Traffic meters measure the temporal properties of the stream of packets selected by a classifier against a traffic profile specified in a SLA.

Packet markers set the DS field of a packet to a particular codepoint, adding the marked packet to a particular DS behavior aggregate.

Shapers are used to shape traffic streams to a certain temporal profile and droppers simply discard packets.

Traffic conditioners consisting of these elements are usually located within DS ingress and egress boundary nodes, but may also be located in nodes within the interior of a DS domain, or within a non-DS-capable domain.

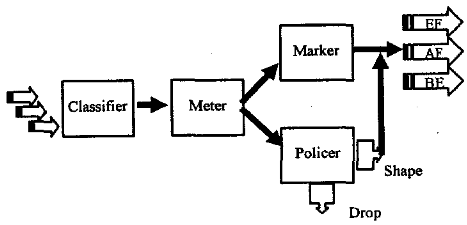



Fig. 1. Traffic Conditioner

## 3. PROPOSAL OF BALANCING MARKER ALGORITHM

In this paper, we intend to improve the problem of the unfairness occurred when individual connections in an aggregated source experience different delays.

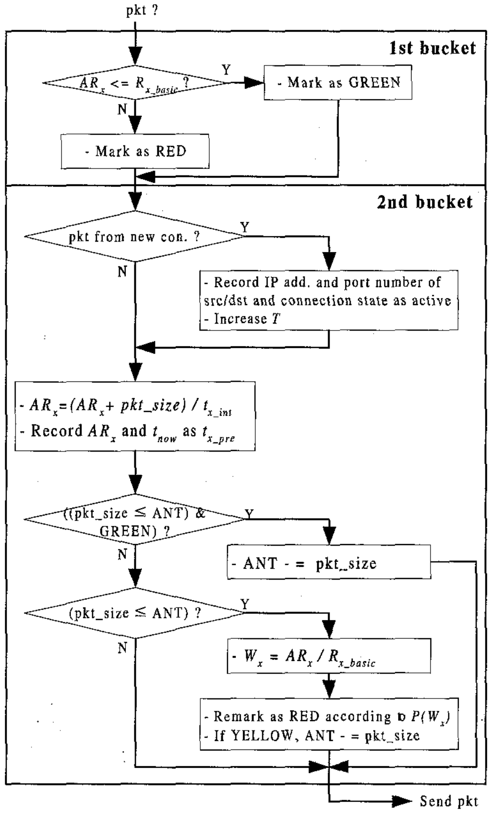
In case that an aggregated source contracts the total service profile, there is the unfairness between individual connections. All the connections sharing one bucket, it has a merit that we can increase their throughputs, but if there is no proper control function, it causes the serious unfairness because the connections with large round-trip delays can't be guaranteed the fair token usage ratio. That is, those with low RTTs consume most of the tokens in the common bucket and their packets are forwarded as IN packets. So they may achieve the high throughputs.

On the contrary, packets from the connections with high delays are likely to be forwarded marked as OUT packets. Therefore, in time of congestion, they may not accomplish even their Basic Assured Bandwidth (BAB, $R_{x\_basic}$), as networks drop OUT packets prior to IN packets.

To fix this problem, we propose the Balancing Marker Algorithm (BMA) that can improve the fairness between the individual connections by the double-checking mechanism. That is, instead of the common bucket method using in the existing AS all the packets from connections are checked just one time by a common token bucket algorithm, we check twice using the 1st and the 2nd token bucket algorithm. And, according to these, each packet is assigned one of the 3 level priorities.

In the BMA, the BMarker manages the 1st and the 2nd token buckets. Fig. 2 is the flow chart of the BMA.

First, the individual connections may be allocated $R_{x\_basic}$ each according to the interior policies of the users network. . In this paper, we assume that $R_{x\_basic}$ is equal to the value dividing the total average sending rate that the aggregated source contracted with the network by the total number ($T$) of the connections maintaining the connection state (Surely, it is so simple thing that we let this value generalize to have a different value per connection).




| Variables | Mean |
|---|---|
| $x$ | Number of Connection |
| $R_{x\_basic}$ | Basic Assured Bandwidth of $x$ |
| $AR_x$ | Current Average Sending rate of $x$ |
| $t_{x\_int}$ | $t_{now}$ - $t_{x\_pre}$ |
| ANT | Available Number of Tokens in the 2nd Bucket |
| $W_x$ | Channel Possession Measure of $x$ |
| $P(W_x)$ | RED Re-marking Probability |

Fig. 2 Balancing Marker Algorithm

In case that any connection $x$ sent a packet, the BMarker checks if the connection $x$ observes $R_{x\_basic}$ in the 1st bucket. According as whether the connections obey $R_{x\_basic}$ or not, the BMarker assigns GREEN grade or RED grade.

In the 2nd bucket, the BMarker first makes some status records about the connections whenever a new packet arrives. This recording is made in the connection table maintained by the BMarker. This table contains, for each connection, the source address, port number, average packet arrival rate ($AR_x$), the time of last packet arrival ($t_{x\_pre}$), and connection state - that is, active or non-active.

Whenever a packet arrives at the 2nd bucket, the BMarker records relevant information to the connection table. If there is not an entry for this connection in the connection table, create a new entry in the table. After that, it decides the number of the active connections ($T$). Going through all the active connections in the connection table, it checks each $t_{x\_pre}$ against a threshold value. If a connection has not generated a packet for more than this threshold time, that connection is marked as non-active. In this way, counting the number of active entries, it can decide the number of active connections, $T$ at a certain time.

And, $AR_x$ is the rate estimate upon each packet arrival and $t_{x\_int}$ is the difference between $t_{\_now}$ and $t_{x\_pre}$. So, state variables $AR_x$ and $t_{x\_int}$ are updated each time a packet arrives.

If a packet from a connection was marked as GREEN in the 1st bucket and if there are tokens remaining in the 2nd bucket, the BMarker sends the packet with GREEN grade after reducing the number of tokens in the bucket by the packet size.

If a packet from a connection was marked RED and if there are no sufficient tokens for the packet in the 2nd bucket, the BMarker sends it without any remarking. If the incoming packet was GREEN, it is remarked as RED.

But, if there are tokens for this packet in the 2nd bucket and the incoming packet is RED, BMarker statistically remarks this packet as YELLOW according to a probability which is decided as follows: it checks whether this connection uses fair share of the total reserved bandwidth in the 2nd bucket or not. That is, according to the ratio of $AR_x$ and $R_{x\_basic}$, the BMarker calculates the utilization measure of each connection, $W_x$ as in Fig. 2. The higher current average sending rate is, the higher $W_x$ will be. The RED-to-YELLOW remarking probability is determined from this value. If $W_x$ is large, the probability will be small. Because the connections with low RTTs take less time than those with high RTTs to send packets, they will consume many tokens and maintain the high $AR_x$ and $W_x$. Therefore, RED packets from the connections with low RTTs will not be remarked as YELLOW, while those from the connections with large RTT will be more likely to be remarked as YELLOW, which will be protected against the congestion. So the throughputs of the connections with low RTTs will decrease and those of the connections with high RTTs will increase, which result in the increased fairness between connections.

## 4. SIMULATION

### 4.1 Network Topology and Parameters

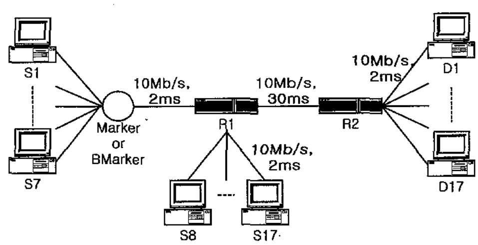



Fig. 3 Network Topology

| | |
|---|---|
| Transmission Rate and Delays Between from S1 to S7 and (B)Marker (Mb/s, (ms)) | 10, (100,80,60,40,20,10,2) |
| Simulation Time (secs) | 60 |
| TCP Source Type | Reno |
| Maximum Packet Size (bytes) | 1000 |
| Maximum Window Size (pkts) | 110 |
| Buffer Size of Router (pkts) | 100 |
| 1st Token Bucket Size (pkts) (Marker, BMarker) | (70↔35), (10↔5) |
| 2nd Token Bucket Size (pkts) (BMarker) | 70↔35 |
| Reservation Level (%) | 70↔35 |
| RED Parameter_GREEN | 40, 60, 0.02 |
| RED Parameter_YELLOW | 30, 50, 0.04 |
| RED Parameter_RED | 20, 40, 0.1 |

Table 3. Parameters

To show the effectiveness of the proposed BMarker algorithm, we compared the 2-level packet priority marking method using the existing Marker (Method 1) with the 3-level packet priority marking method using the BMarker proposed in this paper (Method 2). Fig.

3 and Table 1 show the network topology and parameters. We used NS-1 simulator [5] and analyzed performance in aspect of the fairness.

Each source $S_x$ sends packets with different propagation delays to $D_x$. All the sources always have packets to send and always send a maximal-sized packet. A destination sends an ACK packet when it receives a data packet.

The individual connections in the aggregated source are from $S_1$ to $S_7$. The individual sources send packets with different propagation delays. Delays are as follows; $S_1$ 100ms, $S_2$ 80ms, $S_3$ 60ms, $S_4$ 40ms, $S_5$ 20ms, $S_6$ 10ms and $S_7$ 2ms. Such a large variation in the propagation delays in a user network is not realistic but we set these parameters in that way to emulate the effect of a large variation in the round-trip delays that usually leads to severe unfairness between connections. We fixed the transmission rate as 10Mb/s.

$S_8$ to $S_{17}$ are best-effort sources. We set 2ms as their propagation delays and 10Mb/s as their individual transmission rate. We chose the link between $R_1$ and $R_2$ as the bottleneck.

In each simulation, we examined the fairness and the throughputs in case of being two UDP sources. They act as ON-OFF sources. The period of ON-OFF is 0.04 (sec) and they send packets as the full speed of the link (10Mb/s) when ON.

We simulated 4 cases by mixing two parameters, the average token rate and the bucket size of the 2$^{nd}$ token bucket.

4 simulation conditions are as follows;
- case 1, average token rate 7 Mb/s, bucket size 70 pkts
- case 2, average token rate 3.5 Mb/s, bucket size 35 pkts
- case 3, average token rate 7 Mb/s, bucket size 35 pkts
- case 4, average token rate 3.5 Mb/s, bucket size 70 pkts

We will show the results of case 1 and case 2 in the body and the others in the appendix. The results were obtained by averaging 10 simulation runs.

And, we used the *Jain's metric* of fairness [6]. For $n$ flows, with flow $x$ receiving a fraction $tx$ on a given link, the fairness of the allocation is defined as;

$$Fairness\_index = \frac{(\sum_{x=1}^{n} t_x)^2}{n \times \sum_{x=1}^{n} (t_x^{\ 2})}$$

## 4.2 Simulation Results

* *Basic Assured Bandwidth (BAB)*

$$= \frac{Reserved\ BW\ of\ Aggregated\ Source}{Number\ of\ Individual\ Cons.}$$

* *Available Target Bandwidth (ATB)*

*- Individual Cons of Aggregated Source*

$$= BAB + \frac{Excess\,BW}{Number\,of\,(Individual\ Cons. + Best\ Effort\ Sources)}$$

*- Best Effort Sources*

$$= \frac{excessBW}{NumberOf\,(Individud\ Cons. + Best\ Effort\ Sources)}$$

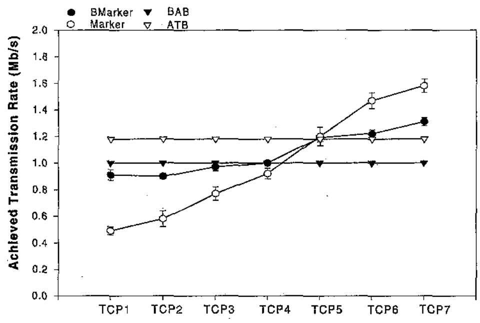
The simulation result for the first case (average token rate 7 Mb/s, bucket size 70 pkts, for the second token bucket) is shown in the Fig. 4. This figure shows the transmission rates achieved by each source using Method 1 and Method 2. Fig. 4 (a) is the transmission rates of the individual connections and Fig. 4 (b) is those of the best-effort sources.

In Fig. 4 (a), we can see the large difference of the transmission rates between the individual connections. $TCP_1$ with the highest delay hardly achieved the only half of BAB, 1Mb/s but $TCP_7$ with the smallest delay reached the much higher rate than BAB and used 1.58Mb/s over ATB. Theoretically, the congestion window if a TCP connection increases at the rate inversely proportional to the square of the round-trip delay [7]. In this simulation, the fairness index of Method 1 was 0.86.

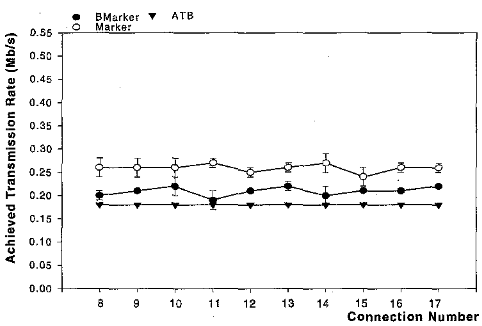
We can see the improvement in the fairness in the Method 2.

In case of packets of $TCP_7$ with lowest delay, it might take for them little time to reach the bucket. They might consume many tokens relative to the connections with high delays. But, as $TCP_7$ had already consumed many tokens relatively, its $AR_x$ and $W_x$ will be so high that the packets from $TCP_7$ will be assigned a low RED-to-YELLOW re-marking probability.

On the contrary, in case of RED packets from $TCP_1$, as its $AR_x$ was low relatively, they would




(a) Individual Connections




(b) Best -Effort Sources

Fig. 4. Achieved Transmission Rate of Case 1

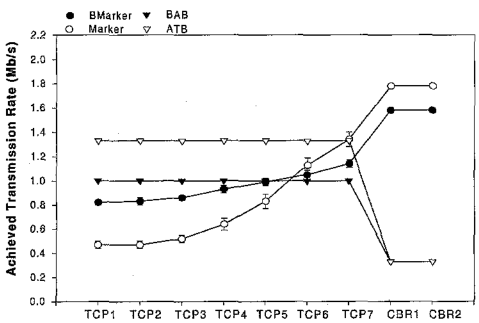



Fig. 5. Individuals and UDP Sources - Case 1

be assigned the high YELLOW re-marking probability according to low $W_X$. So many RED packets of $TCP_7$ will be re-marked as YELLOW and finally this resulted in the improvement of throughput of this connection.

In the figure, we can recognize that the individual connections utilized the reserved bandwidth so fairly and that most of the individual connections achieved BAB.

Also, the their standard deviations were low and it means the performance of the BMarker is so stable. The fairness index of Method 2 was 0.98 and it is much higher than that of Method 1.

In both methods, the connections used the reserved bandwidth 7Mb/s fully. The total throughput of Method 1 was 7.01Mb/s and that of Method 2 was 7.50Mb/s.

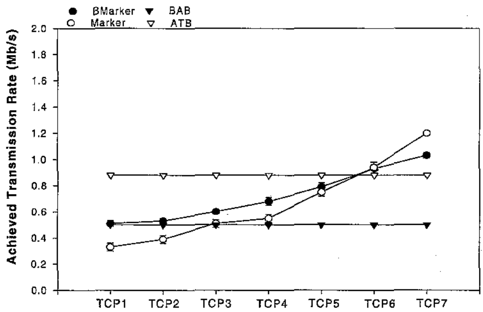
Fig. 4 (b) shows the sending rates achieved by the best-effort sources. When we used the Method 2, their rates were almost same as their ATB. But, the sending rates in Method 1 were higher than those in case of using Method 2. It means that the BMarker performed its task better than the basic Marker. The total sending rates of the best-effort sources were 2.59Mb/s in Method 1 and 2.09Mb/s in Method 2.

Fig. 5 shows the sending rates of the individual connections and the UDP sources. Because the UDP sources dont respond to the traffic control strategies of networks, they occupy all the excess bandwidth of the networks. The sending rate of a UDP source was 1.78Mb/s in Method 1 and 1.58Mb/s in Method 2. That is, two UDP sources got used up all the excess bandwidth.

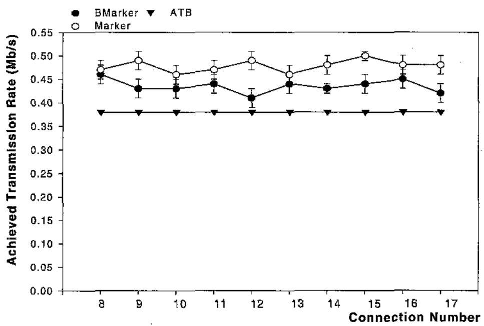
In Fig. 5, we can see that the sending rates of the individual connections are more invariable in Method 2 than in Method 1. The throughputs and the fairness of Method 1 were 5.40Mb/s, 0.85 and those of Method 2 were 6.62Mb/s, 0.99. In case 2, we set the average token rate 3.5Mb/s and the $2^{nd}$ bucket size 35 packets. In Fig. 6, as you see, the fairness in Method 2 was higher than that of Method 1. All the individual connections achieved BAB and they used the total reserved bandwidth fairly. The fairness indexes of Method 1 and Method 2 were 0.84 and 0.94. And as the total throughput was each 4.67Mb/s and 5.07Mb/s, that of Method 2 was higher. The best-effort sources achieved the sending rates similar to ATB, as we set the parameters properly.

Fig. 7 shows the sending rates of the individual connections and the UDP sources. All the connections from $TCP_1$ to $TCP_7$ achieved their BABs and also used the tokens in the $2^{nd}$ bucket fairly. The total throughputs achieved by the individual connections were 5.37Mb/s in Method 1 and 5.58 in Method 2.

Until now, we simulated Method 1 and



(a) Individual Connections



(b) Best -Effort Sources

Fig. 6 Achieved Transmission Rate of Case 2

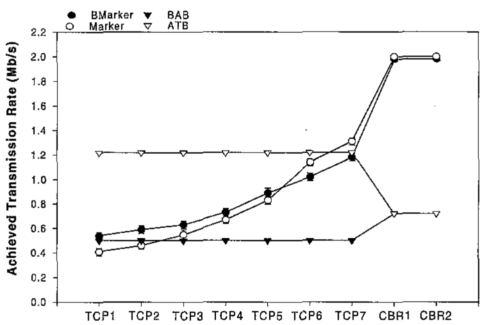


Fig. 7. Individuals and UDP Sources - Case 2

Method 2 changing the parameters, the reservation level and the 2nd bucket size. Table 2 shows the fairness indexes and the throughputs as the results of all the simulations. The values within parentheses are the throughputs. We present the results of the case 3 and case 4 are presented in the appendix.

| | NO UDP | | UDP | |
|---|---|---|---|---|
| | Marker | BMarker | Marker | BMarker |
| Case1 (7M, 70) | 0.86(7.01) | 0.98(7.50) | 0.85(5.40) | 0.99(6.62) |
| Case2 (3.5M, 70) | 0.83(4.67) | 0.95(5.28) | 0.85(5.34) | 0.95(5.61) |
| Case3 (7M, 35) | 0.87(6.97) | 0.97(7.38) | 0.87(6.52) | 0.98(6.52) |
| Case4 (3.5M, 35) | 0.84(4.67) | 0.94(5.07) | 0.85(5.37) | 0.93(5.58) |

Table 2. Fairness Indexes and Throughputs

## 5. CONCLUSIONS

In this paper, we proposed the Balancing Marker Algorithm to improve the unfairness that occurs when the individual connections in an aggregated sources experience different delays. The effectiveness of this algorithm was shown through simulations. We showed that the BMA distributed the tokens fairly to each connection and that the BMA increased the fairness and the throughput compared with the 2-level packet priority marking method.

## REFERENCES

[1] S. Blake et al., "An Architecture for Differentiated Services", *RFC2475, IETF Diff-Serv Working Group*, December 1998

[2] T.V. Lakshman and Upamanyu Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss", *IEEE/ACM Transactions on Networking*, June 1997

[3] Ikjun Yeom and A.L. Narasimha Reddy, "Impact of marking strategy on aggregated flows in a differentiated services network", *Proceedings of IWQoS*, May 1999

[4] S. Blake et al., "An Architecture for Differentiated Services", *RFC2475, IETF Diff-Serv Working Group*, December 1998

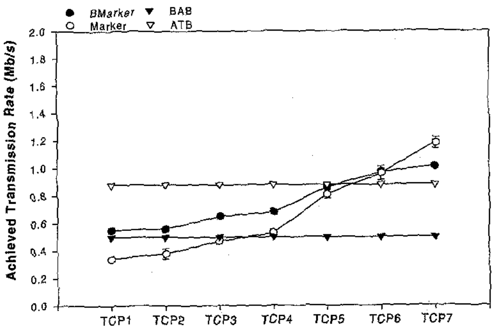
[5] Network Simulator, *University of California at Berkeley, CA*, 1997

[6] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", *Computer Networks and ISDN Systems*, 1989

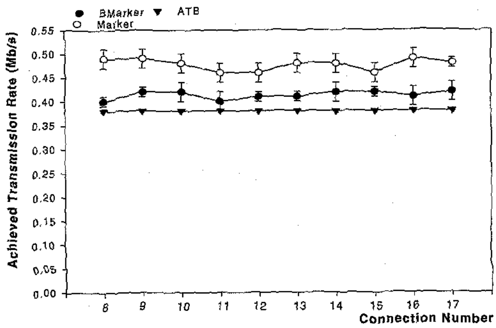
[7] W. Richard Stevens, "TCP/IP Illustrated", Volume I, 1994

## APPENDIX

### - Case 3




(a) Individual Connections




(b) Best -Effort Sources

Fig. 8. Achieved Transmission Rate of Case 3

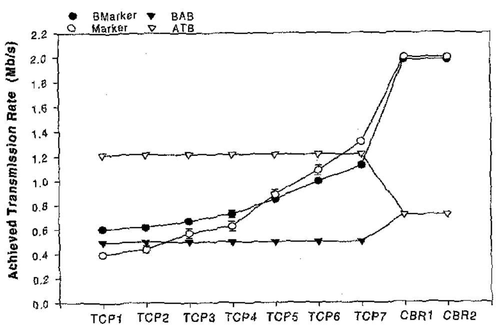
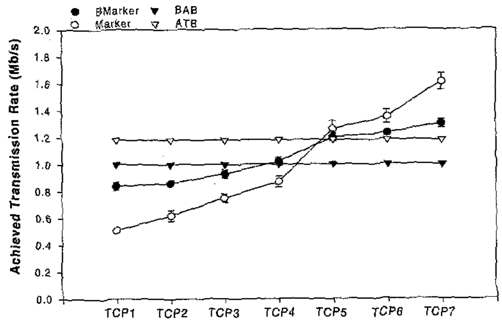
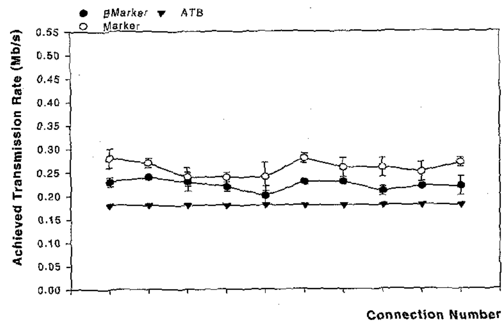



Fig. 9. Individuals and UDP Sources - Case 3

### - Case 4




(a) Individual Connections




(b) Best -Effort Sources

Fig. 10. Achieved Transmission Rate of Case 4

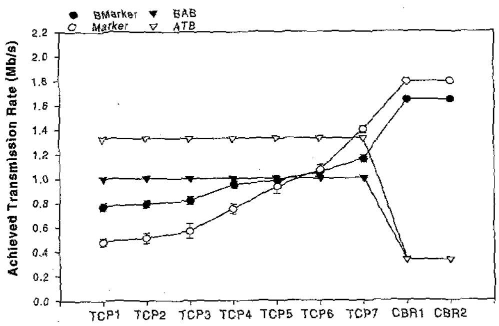



Fig. 11. Individuals and UDP Sources - Case 4