논문 06-31-4B-01 한국통신학회논문지 '06-4 Vol.31 No.4B

Performance Analysis of Random Early Dropping Effect at an Edge Router for TCP Fairness of DiffServ Assured Service

Kyeong Hur* ***Lifelong Member***

ABSTRACT

The differentiated services(DiffServ) architecture provides packet level service differentiation through the simple and predefined Per-Hop Behaviors(PHBs). The Assured Forwarding(AF) PHB proposed as the assured services uses the RED-in/out(RIO) approach to ensure the expected capacity specified by the service profile. However, the AF PHB fails to give good QoS and fairness to the TCP flows. This is because OUT(out- of-profile) packet droppings at the RIO buffer are unfair and sporadic during only network congestion while the TCP's congestion control algorithm works with a different round trip time(RTT). In this paper, we propose an Adaptive Regulating Drop(ARD) marker, as a novel dropping strategy at the ingressive edge router, to improve TCP fairness in assured services without a decrease in the link utilization. To drop packets pertinently, the ARD marker adaptively changes a Temporary Permitted Rate(TPR) for aggregate TCP flows. To reduce the excessive use of greedy TCP flows by notifying droppings of their IN packets constantly to them without a decrease in the link utilization, according to the TPR, the ARD marker performs random early fair remarking and dropping of their excessive IN packets at the aggregate flow level. Thus, the throughput of a TCP flow no more depends on only the sporadic and unfair OUT packet droppings at the RIO buffer in the core router. Then, the ARD marker regulates the packet transmission rate of each TCP flow to the contract rate by increasing TCP fairness, without a decrease in the link utilization.

Key Words : QoS, DiffServ, Assured Services, TCP, Packet dropping strategy, Fairness

Ⅰ. Introduction

The differentiated services(DiffServ) architecture has been proposed as a scalable way of providing quality of service(QoS) in the Internet[1, 2]. Currently, the Internet Engineering Task Force (IETF) has standardized two per-hop behaviors (PHBs) of Expedited Forwarding(EF) PHB and Assured Forwarding(AF) PHB[3, 4].

In assured service framework, the routers at the edge of the network monitor and mark packets of individual flows. Originally, assured service was proposed to use the RED-in/out(RIO)[5] approach to ensure the expected capacity specified by the service profile. The packets of a flow that obey the service profile are marked IN(in-profile) and the packets that are beyond the service profile are marked OUT(out-of-profile). In a DiffServ aware router, all the incoming packets are queued in the original transmission order. But, during network congestion, the router preferentially drops the OUT packets. However, the AF PHB fails to give good QoS and fairness to the TCP flows because of the TCP phase effect at the RIO buffer. It is caused by sporadic during only network congestion and unfair OUT packet droppings at the RIO buffer, and the TCP's congestion control algorithm working with a different round trip time(RTT), because window flow control protocols have a periodic cycle equal to the connection round trip

---

* 경인교육대학교 컴퓨터교육과(khur@ginue.ac.kr)
논문번호 : KICS2005-10-435, 접수일자 : 2005년 10월 27일, 최종논문접수일자 : 2006년 3월 27일

time[6, 7].

In the case that TCP sources share the bottleneck link bandwidth through the RIO buffer, each TCP flow will have an unfair packet transmission rate due to the TCP's congestion control algorithm working with a different RTT. A TCP flow which has a smaller RTT can transmit more packets compared to other flows which have a larger RTT. This means that TCP flows with different RTTs have inherent unfairness. However, at the RIO buffers in the core routers, OUT packets of TCP flows cannot be dropped in proportion to their packet transmission rates. Of course, we can first think that at the RIO buffer, the number of dropped OUT packets from each TCP flow is directly proportional to its packet transmission rate. However, it is not true because the TCP flows from the different edge routers are aggregated in the core routers so that their relative order in the packet transmission rate is not maintained in the core routers unlike in the edge routers. Furthermore, the OUT packets of each TCP flow are randomly dropped at the RIO buffer with a variable probability which is determined by the congestion state of the bottleneck link[5].

Due to this TCP transmission property, the congestion state(i.e., the average queue size) of the RIO buffer in the core router connected to the bottleneck link changes periodically, which is called the phase effect[6]. This periodical variation of the congestion state at the RIO buffer results in a situation where packets of some TCP flows having periodic cycles are more frequently dropped, due to the buffer overflow or the random drop according to the average queue size of RIO buffer. Therefore, the throughputs of those damaged TCP flows are much smaller than the throughputs of other TCP flows.

Previous works on DiffServ networks[5,8,9] have shown that the assured service provided depends on the interaction of the actions of the routers inside the network, the sender, the marker and the interaction among the different flows. Andin those previous works, with only the dropping mechanism of RIO buffers in the routers, marking strategies for improving QoS and fairness of TCP flows of assured services are proposed. That is, those marking strategies are devised under the dropping mechanism of the RIO buffer without an additional dropping mechanism. To improve QoS and fairness of TCP flows, packets of individual flows are adaptively marked respectively according to the marking strategy using the state information of the individual flow at the edge of the network, such as its current sending rate, its contract rate, its round trip time, and its packet drop-rate by the RIO buffer. Then, those proposed marking strategies at the ingressive edge router of the network need per-flow monitoring and signaling.

As previously described, there have been previous works on the marking strategy for improving QoS and fairness of TCP flows of assured services. But, there have been few attempts to improve QoS and fairness of TCP flows of assured services through an additional dropping strategy which needs only the per-flow marker that marks simply the packets of a TCP flow as IN or OUT packets according to only the contract rate. In this paper, we propose an Adaptive Regulating Drop (ARD) marker, as a novel dropping strategy at the ingressive edge router, to improve TCP fairness without a decrease in the link utilization.

The proposed ARD marker monitors IN and OUT packets of aggregate TCP flows, and at the aggregate flow level it performs remarking and dropping packets simultaneously using the state information of aggregate TCP flows, such as current IN packet rate, current OUT packet rate, and the capacity of the bottleneck link. This remarking and dropping of packets should affect all the TCP flows of the ingressive edge router proportionally to their current usage(i.e. fair dropping). The edge router directly connected to the source host of a TCP flow, i.e. the ingressive edge router, can perform this fair early dropping of packets before they enter into the core routers. It is because the ingressive edge router is the first place where the TCP flows are aggregated and there is a dropper for packet dropping inside of it. To prevent degradation of TCP throughput by packet dropping, this fair early dropping is performed only at the ingressive edge router. That is, the consecutive packet dropping at multiple domains is prohibited.

For the performance comparison, we temporarily defined the Adaptive Regulating Marker(ARM), which performs only the adaptive remarking according to the TPR as described above. Then, by comparing the results with the conventional RIO-based scheme and the ARM scheme, we show the effectiveness of the proposed ARD marker in TCP fairness improvement without a decrease in the link utilization. Also, we show that it improves TCP fairness through mitigating the TCP phase effect in the RIO buffer that brings about the unfairness among different TCP flows of assured services.

Ⅱ. ARD Marker

In our scheme, each TCP flow has the per-flow marker in the user's host that marks simply its packets as IN or OUT packets according to only the contract rate, i.e., the average transmission rate that is determined between the user and the ISP (Internet Service Provider). The proposed ARD marker is implemented in the ingressive edge router directly connected to the source host of a TCP flow as shown in Fig. 1.

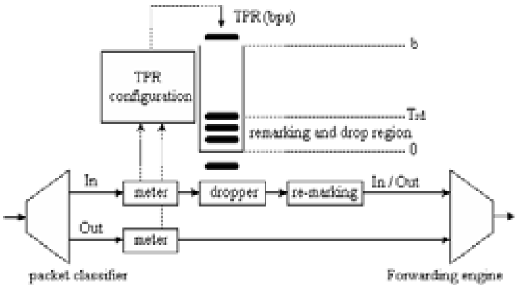

Fig. 1. The proposed ARD marker.

2.1 Fair regulative drop mechanism of the ARD marker for TCP flows

At the aggregate flow level, the ARD marker performs fair remarking and dropping packets simultaneously. And the operations of it work adaptively according to the current state information of aggregate TCP flows, such as the current IN packet rate, the current OUT packet rate, and the capacity of the bottleneck link by monitoring IN and OUT packets of aggregate TCP flows. The ingressive edge router can perform fair early dropping of packets before they enter into the core routers because it is the first place where the TCP flows are aggregated. To prevent degradation of TCP throughput by packet dropping, this fair early dropping is performed only at the ingressive edge router. That is, the consecutive packet dropping at multiple domains is prohibited.

To drop packets pertinently, the ARD marker adaptively changes a Temporary Permitted Rate (TPR) for aggregate TCP flows. The TPR is smaller than or equal to the capacity of the bottleneck link and larger than the current input IN packet rate of aggregate TCP flows. Furthermore, it is set inversely proportional to the measured input OUT packet rate indicating the current degree of excessive use of aggregate greedy TCP flows beyond each flow's contract rate. To reduce the excessive use of greedy TCP flows by notifying droppings of their IN packets constantly to them without a decrease in the link utilization, according to the TPR, the ARD marker performs random early fair remarking of their excessive IN packets to OUT packets at the aggregate flow level. Through the adaptive fair remarking according to the TPR, the ARD marker can regulate the packet transmission rate of a greedy TCP flow to the contract rate by reducing its excessive use. Furthermore, the reduction in the excessive use of greedy TCP flows increases the packet transmission rate of a damaged TCP flow to the contract rate. Therefore, the TCP fairness in assured services is improved. This adaptiveremarking of the ARD marker according to the current network traffic, using the TPR, can avoid excessive remarking of packets to OUT, which decrease the link utilization by excessive OUT packet droppings at the RIO buffer.

To implement this fair regulative remarking, the ARD marker shown in Fig. 1 uses a leaky bucket where the token filling rate is set to the TPR for the aggregate TCP flows. In the leaky bucket, there is a threshold for the remarking and drop T_{rd}. If the number of tokens in the leaky bucket is less than the T_{rd}, an arriving IN packet is randomly remarked to OUT. That is, the excessive IN packets of greedy TCP flows beyond the TPR are randomly remarked to OUT packets. If the arriving rate of IN packets exceeds TPR, the token consumption rate exceeds the token filling rate. Then, the token level in the bucket falls into the remarking and drop region, under the T_{rd}. In the remarking and drop region, each arriving IN packet is randomly remarked to OUT with a probability of P_{rem}, where P_{rem} is a function of the token count in the leaky bucket(TK_{vm}) as shown in Eq.(1). In Eq.(1), MAX_{rem} is the maximum remarking rate. When the leaky bucket runs out of tokens, all arriving excessive IN packets are remarked to OUT packets. Packets remarked to OUT packets do not consume tokens while outgoing IN packets, which are not remarked and not dropped, consume tokens. This remarking rule is intended to be more pertinentfor detecting and dropping IN packets of greedy TCP flows. It is because, as the instantaneous aggregate IN packet rate becomes larger beyond the TPR, the more tokens are consumed and the possibility that IN packets of greedy TCP flows arrive in the remarking region also becomes larger. Thus, we set the P_{rem} inversely proportional to the number of remaining tokens TK_{vm} as shown in Eq.(1). Therefore, we can reduce the erroneous remarkings where IN packets of damaged TCP flows are remarked.

$$P_{rem} = (T_{rd} - TK_{num}) \cdot MAX_{rem} / T_{rd} \quad (1)$$

In addition, in the above remarking of packets, the ARD marker starts early the random remarking of packets to OUT with P_{rem} before the leaky bucket runs out of tokens. The leaky bucket is a deterministic flow control network element that can be used as a traffic marker. Like the drop-tail queue, a simple leaky bucket remarks all IN packets that arrive when there are no tokens available. As argued in [6], much of the Internet traffic is highly periodic, either because of periodic sources (e.g., real time audio or video) or because window flow control protocols have a periodic cycle equal to the connection round trip time(e.g., a network-bandwidth limited TCP bulk data transfer). This phase effect could bring unfairness in the remaking among different TCP flows.

Introducing randomness in the packet selection process of the flow control mechanism could solve this problem. An example is the random early detection(RED) gateway that reduces the unfairness of the drop-tail queue. We applied a similar concept to the leaky bucket by introducing randomness and early decisions on the packet remarking process. Through the early and random remarking decisions on packets, the ARD marker remarks IN packets of each flow approximately in proportion to its current IN packet transmission rate. For the performance comparison, we temporarily defined the Adaptive Regulating Marker(ARM), which performs only this adaptive remarking according to the TPR as described above.

But, however regulative this adaptive remarking works, those remarked packets to OUT packets of greedy TCP flows are still dropped sporadically and unfairly at the RIO buffer in the core router. Therefore, the ARD marker introduces dropping of packets in the remarking process to improve the TCP fairness. In the ARD marker as shown in Fig. 1, an aggregate dropper is combined to drop some excessive IN packets fairly and constantly according to the TPR, instead of remarking them to OUT packets. Thus, the throughput of a TCP flow no more depends on only the sporadic and unfair OUT packet droppings at the RIO buffer in the core router. That is, for those remarked packets to OUT packets of greedy TCP flows, the sporadic and unfair packet droppings at the RIO buffer decrease, while the constant and fair packet droppings at the ARD marker increase. Then, the ARD marker increases TCP fairness of assured services more than the ARM scheme. This restrictive packet dropping is introduced to improve

TCP fairness without reduction of throughput.

At the aggregate flow level, the dropper in the ARD marker drops excessive incoming IN packets randomly with a constant probability P_{drop}, only when the token level of the leaky bucket stays in the remarking and drop region, under the T_{rd}. That is, when the aggregate IN packet rate exceeds the TPR, some of the excessive incoming IN packets from a greedy TCP flow are dropped without a token consumption, instead of being remarked as OUT, in proportion to its current IN packet transmission rate, before they enter into the core routers. This is because the relative order in the IN packet transmission rate is maintained in the ingressive edge routers.

Note that since some of the excessive IN packets, which are likely to be remarked and dropped in the core routers, are dropped instead of being remarked to OUT, this fair early dropping gives little impact on the throughputs of TCP flows. To prevent degradation of TCP throughput, this fair early dropping is performed only at the ingressive edge router and only in the remarking situations. Intrinsically, to achieve the fairness among TCP flows, the TCP host markers should know the state of each TCP flow. However, by doing this fair early regulative packet dropping adaptively to the TPR, the ARD marker can achieve the TCP fairness without a decrease in the link utilization, although it operates at the aggregate flow level without per-flow information. The simulation results in Section 3 support this argument.

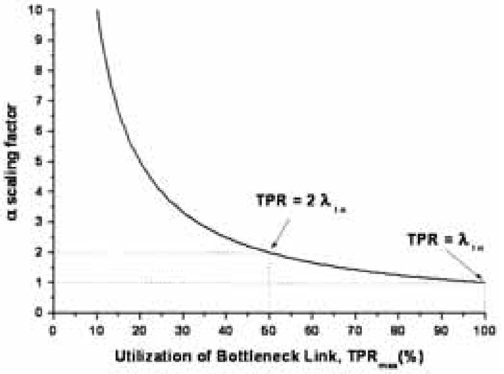

Fig. 2(a) the relation between α and link utilization

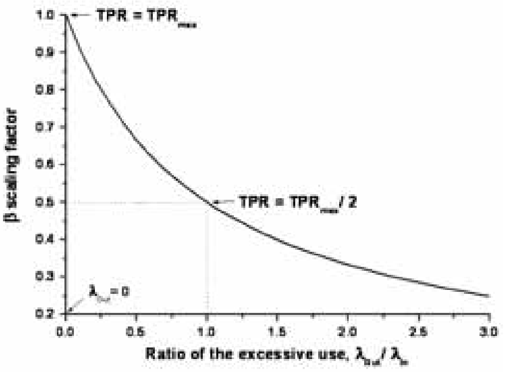

Fig. 2(b) the relation between β and $\lambda_{Out} / \lambda_{In}$

Fig. 2. TPR configuration from the value of α or β scaling factor

2.2 Adaptive configuration method of the TPR

A configuration method of TPR, adaptively according to changes of network traffic, makes the operations of the ARD marker work adaptively. So, it prevents a decrease in the link utilization and sporadic control of the ARD marker over the IN packets of greedy TCP flows. Otherwise, if the TPR is not changed although the state of network traffic has varied, the ARD marker cannot prevent a decrease in the link utilization. It occurs because of excessive packet remarking and dropping at the ARD marker, when the input IN packet rate of aggregate TCP flows becomes much larger than the TPR. Furthermore, when it becomes much smaller than the TPR because of the above excessive packet remarking and dropping at the ARD marker, the token level in the leaky bucket cannot enter the remarking and drop region. And then the operations of ARD marker cannot be performed constantly over the IN packets of greedy TCP flows to reduce their excessive use. Consequently, in this case, TCP fairness cannot be achieved. The TPR is calculated as follows;

$$TPR = TPR_{\max} \cdot \left(\frac{\lambda_{In}}{\lambda_{In} + \lambda_{Out}} \right) \qquad (2)$$

where $TPR_{\max}$ the maximum TPR corresponding to the bandwidth of the bottleneck link, λ_{IN} denotes the aggregate input IN packet rate, i.e., the aggregate rate of the incoming IN packets of TCP flows, λ_{OUT} denotes the aggregate input OUT packet rate. For measuring the incoming rate of each colored packets at the aggregate flow level, two input traffic meters are used after packet classification, as shown in Fig. 1. Each input traffic meter measures the aggregate rate of incoming each colored packet for a time interval $n\tau$ and does it every time interval. Using these values, every $n\tau$ time interval the TPR configuration block in the ARD marker newly calculates the TPR for the next $n\tau$ time interval. The TPR is used as the token filling rate of the leaky bucket. The difference between the average aggregate IN packet rate of λ_{IN} and the TPR determines the operation region of the leaky bucket. If the instantaneous aggregate IN packet rate is larger than the TPR, the leaky bucket stays in the remarking and drop region and some of the IN packets from greedy TCP flows are remarked and dropped. Therefore, the chance of an IN packet drop increases as the TPR becomes smaller.

$$TPR = TPR_{\max} \cdot \beta = TPR_{\max} \cdot \left(\frac{1}{1 + (\lambda_{Out} / \lambda_{In})} \right) ,$$

$$\beta = \left(\frac{1}{1 + (Excessive\ use\ ratio)} \right), \quad \beta \le 1 \qquad (3)$$

$$TPR = \left(\frac{TPR_{\max}}{\lambda_{In} + \lambda_{Out}} \right) \cdot \lambda_{In} = \alpha \cdot \lambda_{In} ,$$

$$\alpha \cong \left(\frac{TPR_{\max}}{Link\ utilization} \right), \quad \alpha \ge 1 \qquad (4)$$

Equation (2) can be written as shown in Eqs.(3) and (4). The β in Eq.(3) denotes a scaling factor for the $TPR_{\max}$ in configuring the TPR, and the α in Eq.(4) denotes a scaling factor for the λ_{IN}. In Eq.(3), we named the ratio of $\lambda_{OUT}/\lambda_{IN}$ as the excessive use ratio of greedy TCP flows, to which the β factor is inversely proportional. And the value of $(\lambda_{IN}+\lambda_{OUT})$ is regarded as the utilization of the bottleneck link in Eq.(4). The α factor is inversely proportional to the value of $(\lambda_{IN}+\lambda_{OUT})$. That is, if the link utilization becomes lower, the TPR is set more larger than the aggregate IN packet rate λ_{IN}. Thus, a smaller amount of IN packets of greedy TCP flows is remarked and dropped in the remarking and drop region. It shows that the TPR is configured to be larger to increase the link utilization according to the current degree of link utilization. Figures 2(a) and 2(b) show this TPR configuration from the value of α or β scaling factor.

As shown in Eq.(3), the calculated TPR is smaller than or equal to the capacity of the bottleneck link $TPR_{\max}$. In this paper, we assumed that the traffic rate defined in the Service Level Agreement(SLA) at the ingressive edge router is always set equal to the $TPR_{\max}$; the bandwidth of bottleneck link in the domain. Then, the sum of the contract rates of TCP flows(the aggregate contract rate) at the ingressive edge router is smaller than or equal to the bottleneck link bandwidth. Therefore, this new configuration method of the TPR, by which the TPR is set not to be larger than the $TPR_{\max}$ equal to SLA, does not violate the SLA.

Note that the TPR is set proportional to the measured input IN packet rate λ_{IN} as shown in Eq.(4). Therefore, the ARD marker can remark and drop IN packets of greedy TCP flows constantly according to the current aggregate IN packet rate. In addition, as shown in Eq.(3), the TPR is set to become smaller as the measured input OUT packet rate λ_{OUT} becomes larger, where the λ_{OUT} indicates the current degree of excessive use of aggregate greedy TCP flows beyond each flow's contract rate. As the TPR becomes smaller, the ARD marker enters into the remarking and drop region more frequently. Therefore, the chance of an IN packet drop for greedy TCP flows is inversely proportional to the value of the TPR and proportional to the current degree of excessive use of greedy TCP flows. Such a remarking and dropping rule is proper for reducing the excessive use of greedy TCP flows to achieve TCP fairness.

Furthermore, this operation is intuitively true. It is because OUT packets occur only when there is an excessive capacity in the network so that greedy TCP sources generate packets more than each flow's contract rate and the number of OUT packets is directly proportional to the excessive capacity of the network. So, the ARD marker adaptively prevents the greedy TCP flows from using the excessive capacity of the network, likely to be used by them excessively, through the IN packet drop for greedy TCP flows according to the TPR. Consequently, now, the excess capacity of the network is allocated to the damaged TCP flows. In assured services, it is desirable that the aggregate IN packets of TCP flows fully utilize the bottleneck link when the aggregate contract rate of those TCP flows is equal to the bottleneck link bandwidth $TPR_{\max}$. In this case, the λ_{OUT} becomes zero, so that the TPR is calculated to be $TPR_{\max}$ equal to the bottleneck link bandwidth, and thus the ARD marker never enters into the remarking and drop region since the aggregate input IN packet rate λ_{IN} cannot exceed the aggregate contract rate.

On the other hand, the TPR should be larger than the λ_{IN} measured during the last $n\tau$ interval. If the TPR is calculated to be less than the previously measured λ_{IN}, the ARD marker can enter into the remarking and drop region while λ_{IN} is being decreased. This is not a desirable situation, since the decrease of λ_{IN} means the result of the IN packet drop by the operations at the remarking and drop region in the last $n\tau$ interval. So, further IN packet drop in the remarking and drop region can result in the decrease of the TCP throughput. To avoid this undesirable situation, as shown in Eqs.(2) and (4), the TPR can be computed by multiplying $TPR_{\max}$ by the ratio of λ_{IN} to the sum of λ_{IN} and λ_{OUT}. Therefore, the computed TPR is always larger than λ_{IN} as shown in Eq.(4), because the sum of λ_{IN} and λ_{OUT} is smaller than $TPR_{\max}$ corresponding to the bandwidth of the bottleneck link. This is because TCP flows cannot fully use the bandwidth of the bottleneck link due to their congestion control algorithm. As previously explained, by using this new configuration method of TPR, the ARD marker remarks and drops IN packets of greedy TCP flows properly and constantly whenever the network traffic changes. As results of these operations described in the previous Subsections, the ARD marker can regulate the packet transmission rate of each TCP flow to the contract rate without a decrease in the link utilization.

The introduced mechanism of the ARD marker has some analogy with the RED gateway. The RED gateway performs a random early dropping of packets from TCP flows when the average queue size is in the $[\min_{th}, \max_{th}]$ range, before all the arriving packets are dropped due to the increased average queue size larger than $\max_{th}$. By this control, the average queue size for TCP flows is controlled to vary almost between the $[\min_{th}, \max_{th}]$ range. The ARD marker performs the random early remarking and dropping of IN packets from TCP flows according to a configured TPR during a $n\tau$ interval, when the token level is in the $[0, T_{rd}]$ range, before all the arriving IN packets are remarked to OUT due to no available tokens with the P_{drop}. Then, from the above result caused by RED gateway control for TCP flows, we can guess the variations of the token level during a $n\tau$ interval. That is, the token level will stay in the remarking and drop region of the $[0, T_{rd}]$ range. This token level variation means that, after the controls of the ARD marker for reducing the excessive use of TCP flows, the aggregateIN packet rate of TCP flows will be stabilized and increased to the configured TPR, larger than the λ_{IN} measured during the last $n\tau$ interval. That is, this result shows the evidence of mitigation of the TCP phase effect, which increases TCP fairness. The simulation results in the following Section support this argument.

Ⅲ. Performance Study

In this Section, we analyze the performance and the effectiveness of the proposed ARD marker. We compare TCP fairness and aggregate throughput with the conventional RIO-based scheme and the ARM scheme through experiments using the ns2 simulator. TCP Reno is used for all the simulation results in this paper. Figure 3 depicts the simulation topologies used to study the performance of the ARD marker. Firstly, we have used 4 TCP flows in the simulations to show the TCP fairness of our ARD marker. Secondly, 20 TCP flows have been used to show that the ARD marker scales well with more flows. In the first case shown in Fig. 3(a), 4 TCP hosts H1, H2, H3, H4 each has a source marker implemented inside, which marks simply its packets as IN or OUT packets according to only the contract rate. We assume that each host contracts 0.25 Mbps for assured service. Thus, initially each host could have up to 0.25 Mbps packets marked as IN. The remaining packets are marked as OUT.

Assured service is implemented in the core routers CR1 and CR2 through the RIO scheme[5]. Both IN and OUT packets are buffered in the same queue having 100 packet queue length. We use two sets of RED parameters for IN and OUT packets. The RED parameters for IN packets are: 45 packets, 60 packets, and 0.02 for $\min_{IN}$, $\max_{IN}$ and $P\max_{IN}$, respectively, where $\min_{IN}$ and $\max_{IN}$ represent the lower and upper bounds for the average queue size for IN packets, and $P\max_{IN}$ is the maximum drop probability for an IN packet when the average queue size is in the $[\min_{IN}, \max_{IN}]$ range. The $\min_{OUT}$, $\max_{OUT}$ and $P\max_{OUT}$ are the corresponding parameters for the OUT packets. They are set to be 20 packets, 40 packets, and 0.05 for $\min_{OUT}$, $\max_{OUT}$ and $P\max_{OUT}$, respectively. Four TCP flows tcp1, tcp2, tcp3, tcp4 originate from hosts H1, H2, H3, and H4, respectively and all terminate at CR2. The RTT of each TCP flow is 26, 28, 30, 32 ms, respectively. Throughput of each TCP flow is measured at the CR2 core router. The ingressive edge router ER has the proposed ARD marker implemented inside. In Fig. 3(a), at the ARD marker, $TPR_{\max}$ equal to SLA is set to be 1.2 Mbps which corresponds to the bottleneck link bandwidth between CR1 and CR2, and the aggregate contract rate of the TCP hosts is 1.0 Mbps smaller than the bottleneck link bandwidth.

In the second case using 20 TCP flows, each TCP host has a0.25 Mbps contract rate and it has the simple source marker implemented inside. And, as shown in Fig. 3(b), the bottleneck link bandwidth between CR1 and CR2 is changed to 6 Mbps. Then, in this case, $TPR_{\max}$ is set to be 6 Mbps and the aggregate contract rate of the TCP hosts is 5.0 Mbps smaller than the $TPR_{\max}$. In addition, according to the change of bottleneck link bandwidth in this case, the RIO queue has 400 packet queue length, and it has 180 packets and 240 packets for $\min_{IN}$ and $\max_{IN}$ and 80 packets and 160 packets for $\min_{OUT}$ and $\max_{OUT}$. Twenty TCP flows tcp1, tcp2, , tcp19, and tcp20 originate from hosts H1, , and H20, respectively. The tcp1 flow has the smallest 26.4 ms RTT, while the tcp20 flow has the largest 34 ms RTT. The RTT of each flow increases by 0.4 ms such

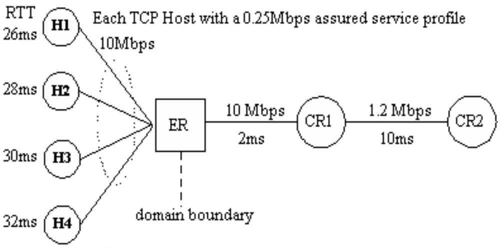

Fig. 3(a) Simulation topology using 4 TCP flows

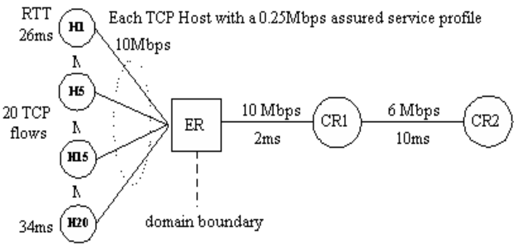

Fig. 3(b) Simulation topology using 20 TCP flows

Fig. 3. Simulation topologies used to study performance of the ARD marker.

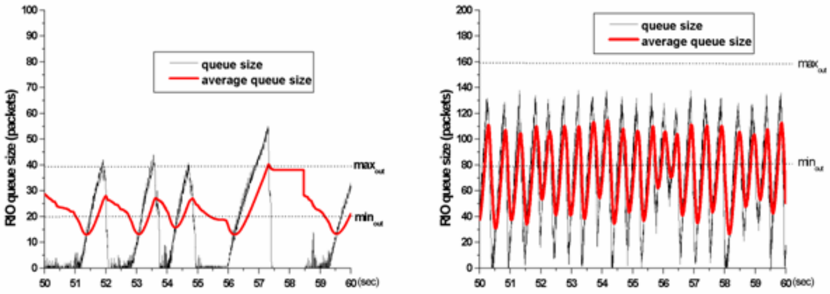

Fig. 4(a) RIO queue size at 4 flows

Fig. 4(b) RIO queue size at 20 flows

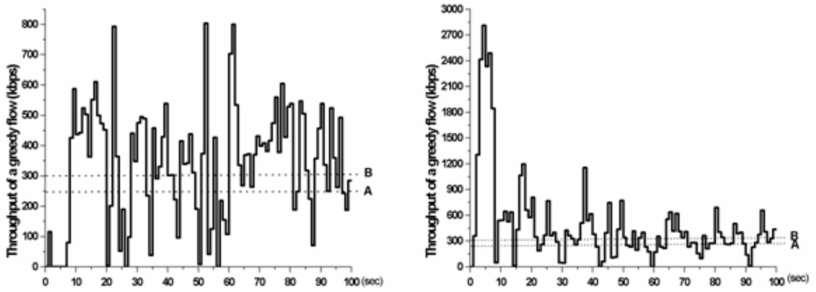

Fig. 4(c) RIO at 4 flows
A : Contract rate, B : Fair share for bottleneck link

Fig. 4(d) RIO at 20 flows
A : Contract rate, B : Fare share for bottleneck link

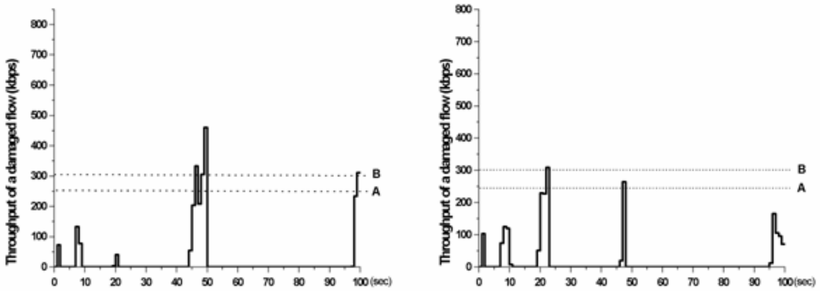

Fig. 4(e) RIO at 4 flows
A : Contract rate, B : Fare share for bottleneck link

Fig. 4(f) RIO at 20 flows
A : Contract rate, B : Fare share for bottleneck link

Fig. 4. TCP fairness performances of the RIO-based scheme

as 26.4 ms, 26.8 ms, 27.2 ms, , 33.6 ms, and 34 ms. In all our simulations, we set the size of the leaky bucket b to 60 packets, T_{rd} is set to 15 packets, and MAX_{rem} probability is set to 0.5. The time interval $n\tau$ during which the ARD marker measures the traffic rates is set to 100 seconds, where τ is set to 0.1 second and n is set to 1000. The drop probability P_{drop} in the combined dropper is constantly set to 0.02.

Initially, the token filling rate of the leaky bucket is set to be $TPR_{\max}$ in ARD and ARM markers. And by the proposed TPR configuration method, ARD and ARM markers measure the input IN and OUT packet rates during $n\tau$ at ER. In the case using 4 TCP flows shown in Fig. 3(a), the λ_{IN} measured is 494 kbps and the λ_{OUT} measured is 546 kbps. Note that the sum of λ_{IN} and λ_{OUT} of aggregate TCP flows at ER is much smaller than the bandwidth of the bottleneck link $TPR_{\max}$ due to the TCP's congestion control algorithm. According to the Eq.(2), the TPR for the next $n\tau$ time interval is calculated as 570 kbps which is larger than the λ_{IN} measured during the last $n\tau$ interval. On the other hand, in the case using 20 TCP flows shown in Fig. 3(b), the λ_{IN} and λ_{OUT} measured are 1658 kbps and 4102

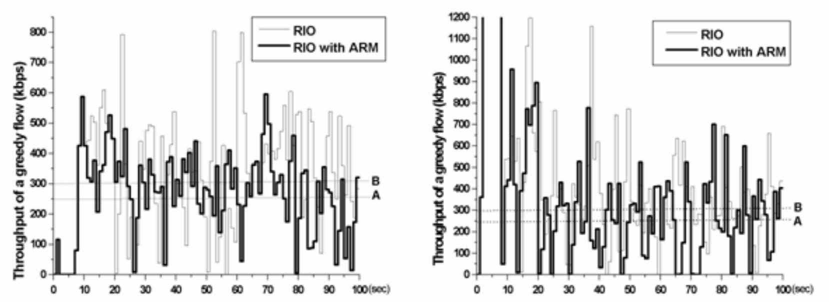

Fig. 5(a) ARM at 4 flows
A : Contract rate. B : Fair share for bottleneck link

Fig. 5(b) ARM at 20 flows
A : Contract rate. B : Fare share for bottleneck link

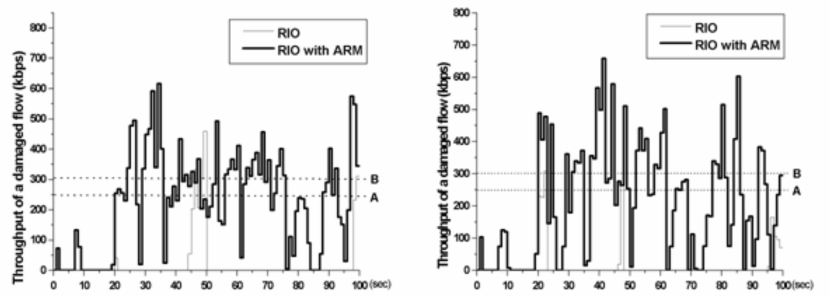

Fig. 5(c) ARM at 4 flows
A : Contract rate, B : Fare share for bottleneck link

Fig. 5(d) ARM at 20 flows
A : Contract rate. B : Fare share for bottleneck link

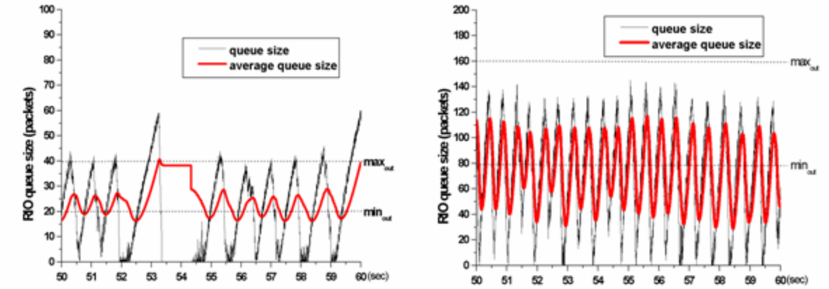

Fig. 5(e) RIO queue size with ARM at 4 flows

Fig. 5(f) RIO queue size with ARM at 20 flows

Fig. 5. TCP fairness performances of the ARM scheme

kbps, respectively. Then, the TPR for the next $n\tau$ time interval is calculated as 1727 kbps larger than the λ_{lN}.

In this paper, all the simulations are executed for 200 sec equal to two $n\tau$ intervals. During the first 100 sec time interval, ARD and ARM markers measure the input traffic rates to determine the TPR by using the proposed configuration method. They update the TPR every $n\tau$ time interval. To show the performances of both markers, we compare the throughputs of both markers and the conventional RIO-based scheme during the second 100 sec time interval. Note that the difference between the ARD and ARM markers is the constant fair early dropping at the remarking and drop region, as previously explained. In the conventional RIO-based scheme, there is no marker for aggregate TCP flows implemented in the ER and there are only the simple per-flow markers implemented in each TCP host and the RIO buffers implemented in DiffServ aware routers. For the simulation topologies shown in Figs. 3(a) and 3(b), ideally, each TCP flow should have 250 kbps IN throughput as the contract rate, and it should get 300 kbps total throughput as the fair share for the bottleneck link bandwidth. Figure 4 shows the performances of the RIO-based scheme at the above simulation topologies. And Figs. 5

and 6 show the performances of the ARM scheme and ARD markers, respectively.

For each simulation topology, Figs. 4(a) and 4(b) show the TCP phase effect[6,7] in the RIO buffer of the CR1 connected to the bottleneck link, where the average queue size changes around the $\min_{OUT}$ periodically so that OUT packet droppings are sporadic, respectively. The RIO buffer does not distinguish between packets of individual flows so that unfair OUT packet dropping, only proportional to the average queue size, is performed. So, this periodical variation of the average queue size results in a situation where OUT packets of some TCP flows having periodic cycles are more frequently dropped whenever the average queue size becomes larger than the $\min_{OUT}$ periodically[5]. That is, due to the phase effect, the throughputs of TCP flows can be highly biased. Therefore, a greedy TCP flow transmits more packets beyond the contract rate, while such damaged TCP flow cannot transmit packets at the contract rate. Figures 4(c) and 4(d) show the throughput variations of a greedy TCP flow at each simulation topology respectively while Figs. 4(e) and 4(f) show the throughput variations of a damaged TCP flow respectively, when using only the RIO-based scheme.

Figures 5(a) and 5(b) show the throughput variations of the greedy TCP flow at each simulation topology respectively when using the ARM scheme. When compared to the results using only the RIO-based scheme shown in Figs. 4(c) and 4(d), we can see that thethroughput per second of the greedy TCP flow is clipped to decrease. So, its average throughput becomes a lower value around the fair share for bottleneck link bandwidth 300 kbps by using the ARM scheme. Note that the ARM scheme performs only the random early fair remarking of excessive IN packets from greedy TCP flows to OUT, according to the TPR, to reduce the excessive use of them by notifying droppings of their IN packets constantly. The reduction in the excessive use of the greedy TCP flow increases the packet transmission rate of the damaged TCP flow because now it feels that there is more capacity in the network, as shown in Figs. 5(c) and 5(d). However, those remarked packets to OUT of greedy TCP flows are still dropped sporadically and unfairly at the RIO buffer. Therefore, the periodic cycles in packet transmission rates of greedy TCP flows and those of damaged TCP flows cannot be destroyed. Consequently, there still exists the phase effect in the RIO buffer of the CR1 at each simulation topology, in Figs. 5(e) and 5(f).

Figure 6 shows performances of the proposed ARD marker, which has introduced the fair early dropping of IN packets of greedy TCP flows in the remarking process of the ARM scheme. By the fair early dropping effect at the ingressive edge router, the throughput per second of the greedy TCP flow at each simulation topology shown in Figs. 6(a) and 6(b) decreases faster than the case using the ARM scheme. This is because the throughput of a TCP flow no more depends on only the sporadic and unfair OUT packet droppings at the RIO buffer in the core router when using the ARD marker. Furthermore, the throughput per second of the greedy TCP flow is suppressed more than the case using the ARM scheme because of the increased constant and fair packet droppings for the remarked packets to OUT of greedy TCP flows at the ARD marker. Consequently, as shown in Figs. 6(c) and 6(d), the throughput per second of the damaged TCP flow also increases faster than the case using the ARM scheme so that its average throughput becomes higher than the case. This improved TCP fairness is also seen in Figs. 6(e) and 6(f) for each simulation topology. By destroying the periodic cycles in packet transmission rates of greedy TCP flows and those of damaged TCP flows through the fair early dropping, as shown in those figures, the TCP phase effect in the RIO buffer is mitigated much when using the ARD marker. These results show that the combined dropper in the ARD marker improves TCP fairness while it works at the aggregate flow level since the only difference between the ARM scheme and ARD marker is that the ARD marker drops IN packets

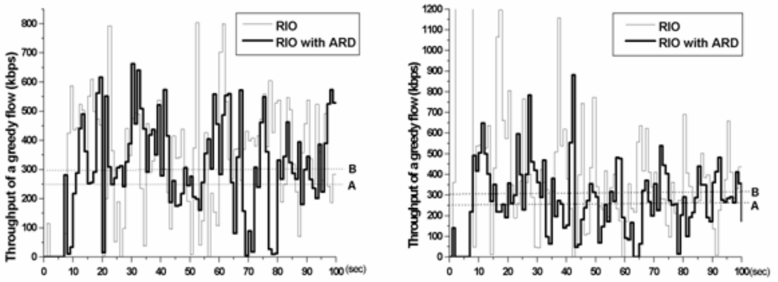

Fig. 6(a) ARD at 4 flows
A : Contract rate, B : Fair share for bottleneck link

Fig. 6(b) ARD at 20 flows
A : Contract rate, B : Fare share for bottleneck link

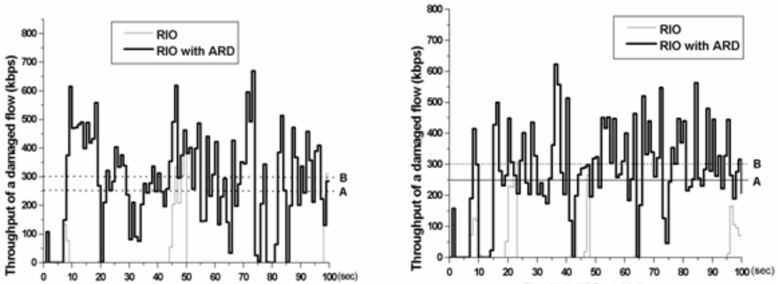

Fig. 6(c) ARD at 4 flows
A : Contract rate. B : Fare share for bottleneck link

Fig. 6(d) ARD at 20 flows
A : Contract rate, B : Fare share for bottleneck link

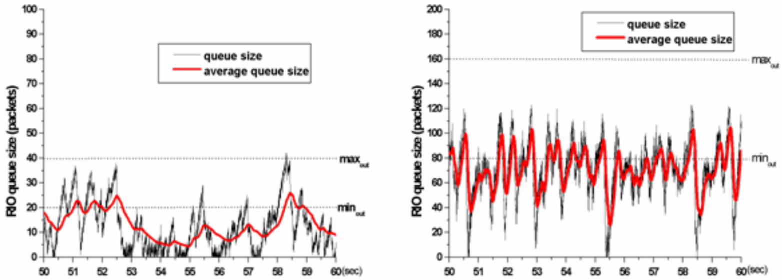

Fig. 6(e) RIO queue size with ARD at 4 flows

Fig. 6(f) RIO queue size with ARD at 20 flows

Fig. 6. TCP fairness performances of the ARD scheme

randomly in the remarking region using the combined dropper. That is, by introducing packet dropping at the ingressive edge router, we can eliminate the phase effect of TCP that occurs in the RIO buffer in the core routers.

To support this argument, we compare the variations of token level in the leaky bucket during the $n\tau$ interval in Fig. 7. The number of tokens in the leaky bucket is measured every τ time. Figure 7(a) shows the observed token level at TPR for the RIO-based scheme. Like the result in Figs. 4(a) and 4(b) for each simulation topology, the TCP phase effect is also shown in this figure where the token consumption rate of aggregate TCP flows highly fluctuates. Furthermore, as described in Fig. 5, the token consumption rate still fluctuates although the ARM scheme works in Fig. 7(b). But, when using the ARD marker as shown in Fig. 7(c), the token level stays in the remarking and drop region of the $[0, T_{rd}]$range. That is, the aggregate IN packet rate of TCP flows is stabilized and increased to the configured TPR, which is larger than the λ_{IN} measured during the last $n\tau$ interval. This result shows that the ARD marker, which performs the random early remarking and dropping of packets according to

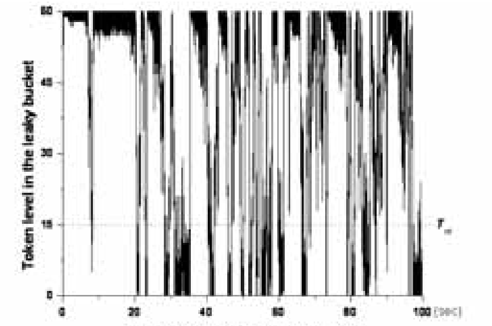

Fig. 7(a) Observed token level at TPR

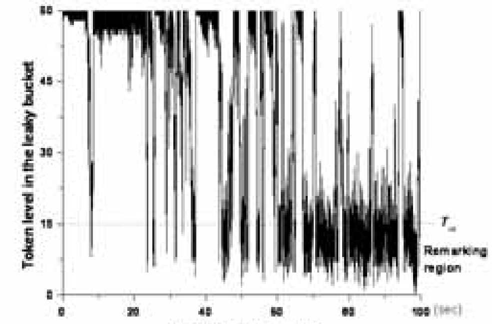

Fig. 7(b) ARM with TPR

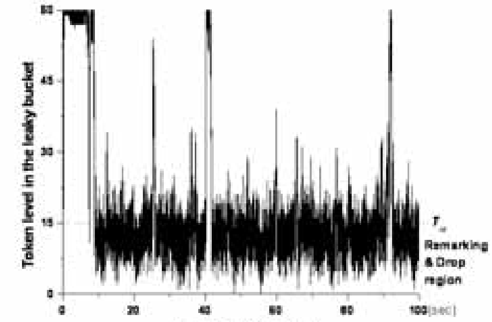

Fig. 7(c) ARD with TPR

Fig. 7. Comparison of the token level variations

the TPR during the $n\tau$ interval, has some analogy with the RED gateway in control ability over the aggregate packet transmission rate of TCP flows, as previously explained in Subsection 2.2. Furthermore, it shows the evidence of mitigation of the TCP phase effect, which increases TCP fairness.

In the previous simulation results for the proposed ARD marker we have shown its regulative control ability over the packet transmission rate of the greedy TCP flow and its mitigation ability over the TCP phase effect. Now, we compare TCP fairness and aggregate TCP throughput for aggregate TCP flows. Figures 8(a) and 8(b) compare the standard deviations in the IN and total throughputs of TCP flows at each simulation topology, respectively. The standard deviation(STD) of the throughput defines the degree of fairness. In addition, in Figs. 8(c) and 8(d), we compare the aggregate IN and total throughputs of TCP flows at each simulation topology shown in Fig. 3, respectively. Note that to increase TCP fairness, the ARM scheme performs only the adaptive fair remarking according to the TPR. So, the excessive use of greedy TCP flows is reduced and the throughput of the damaged TCP flow is increased as shown in Fig. 5. On the other hand, the ARD marker is proposed to enhance the ARM's effect of the adaptive fair remarking on TCP fairness improvement. As shown in Fig. 1, the ARD marker is a structure where an aggregate dropper is combined with the ARM scheme, to increase fair and constant packet droppings for the remarked packets to OUT of greedy TCP flows rather than unfair and sporadic packet droppings in the RIO buffer. From these reasons, the STDs in both IN and total throughputs of TCP flows become lower when using the ARM scheme than when using only the RIO-based scheme. Furthermore, the STDs also become lower when using the ARD scheme than when using the ARM scheme as shown in Figs. 8(a) and 8(b), irrespective of the number of TCP flows.

For the link utilization performance, we have described that the ARD marker improves TCP fairness without a decrease in the link utilization. Firstly, the adaptive remarking of the ARM scheme according to the current network traffic, using the TPR, can avoid excessive remarking of packets to OUT, which decrease the link utilization by excessive OUT packet droppings at the RIO buffer. That is, the TPR is adaptively set to be larger than the current aggregate IN packet rate λ_{IN}, and to be larger to increase the link utilization according to thecurrent degree of link utilization as shown in Eq.(4). As the TPR becomes larger, the amount of arriving IN packets of greedy TCP flows to

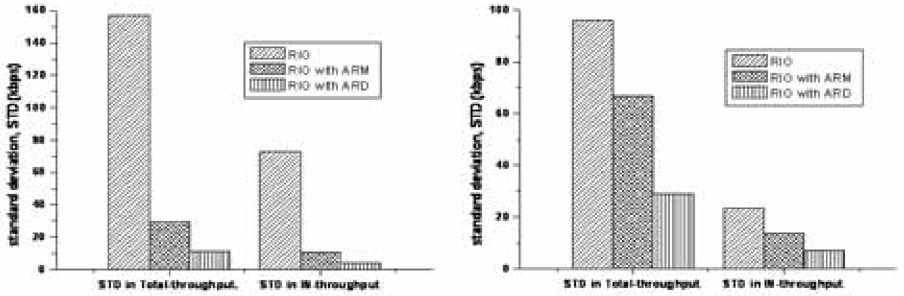

Fig. 8(a) Fairness in throughputs at 4 flows

Fig. 8(b) Fairness in throughputs at 20 flows

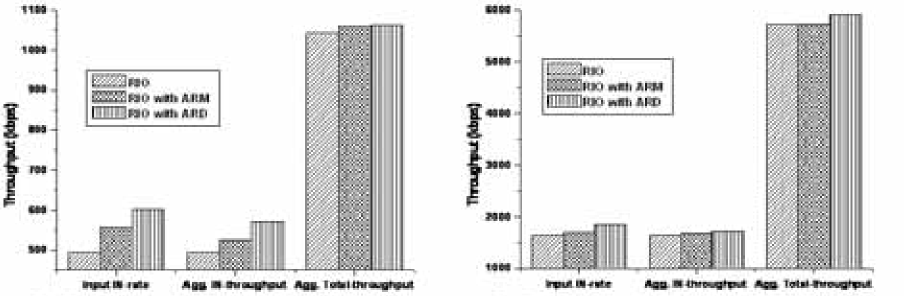

Fig. 8(c) Comparison of aggregate throughputs at 4 TCP flows

Fig. 8(d) Comparison of aggregate throughputs at 20 TCP flows

Fig. 8. Comparison of TCP fairness and aggregate TCP throughput

control becomes smaller. So, if the link utilization becomes lower, the TPR is set more larger than the λ_{IN}. Thus, a smaller amount of IN packets of greedy TCP flows is remarked in the remarking region. In the ARD marker, some of the excessive IN packets, which are likely to be remarked and dropped in the core routers, are dropped instead of being remarked to OUT. So, this fair early dropping gives little impact on the throughputs of TCP flows. Consequently, as shown in Figs. 8(c) and 8(d), there is no large decrease in the link utilization, in the aggregate IN throughput, and in the input IN packet rate when comparing the results of RIO-based scheme with the results of the ARM scheme and ARD marker, respectively.

$$FI = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \times \sum_{i=1}^{N} x_i^2} \tag{5}$$

Figures 8(a) and 8(b) have shown the TCP fairness improvement of the ARD marker by comparing the STD value in total throughputs. On the other hand, to evaluate TCP fairness performance of the proposed ARD marker, we used Jain's metric of fairness as shown in Eq.(5)[10]. In Eq.(5), FI is the fairness index and it is ranged from 0 to 1. For N flows, with flow i receiving a fraction throughput x_i on a given link, the fairness of the allocation is defined as the above FI. According to this definition, the closer the fairness index is to 1, the fairer the bandwidth distribution between flows is. Table 1 shows the fairness index at each scheme simulated in Figs. 3(a) and 3(b). In Table 1, we can see that the FI values at the ARD marker are the largest among the schemes and they are almost equal to 1. That is, like the FI values at other schemes using per- flow information, this result indicates that TCP flows share the bottleneck link bandwidth equally when using the ARD marker, although it operates at the aggregate flow level without per-flow information.

Table 1. Comparison of the fairness index.

	RIO-based scheme	ARM scheme	ARD Marker
Case of Fig. 3(a)	0.786	0.990	0.999
Case of Fig. 3(b)	0.852	0.950	0.991

Ⅳ. Conclusions

In this paper, we have proposed a novel dropping strategy at the ingressive edge router to improve TCP fairness without a decrease in the link utilization. The proposed Adaptive Regulating Drop(ARD) aggregate marker needs only the simple per-flow host marker that marks simply the packets of a TCP flow as IN or OUT packets according to only the contract rate. The ARD marker introduces the configuration method of the Temporary Permitted Rate(TPR). By using the new configuration method of TPR, the ARD marker remarks and drops IN packets of greedy TCP flows pertinently and constantly whenever the network traffic changes.

The simulation results indicate that the ARM scheme improves TCP fairness compared to the RIO-based scheme and that the ARD marker increases TCP fairness more than the ARM scheme by mitigating the TCP phase effect through fair early dropping. Furthermore, for the link utilization performance, it is shown that the ARM scheme does not decrease the link utilization due to avoidance of the excessive remarking according the TPR while the fair early dropping in the ARD marker gives little impact on the link utilization. From the simulation results, we can see that the ARD marker can achieve TCP fairness without a decrease in the link utilization, although it operates at the aggregate flow level without per-flow information.

REFERENCES

[1] K. Nichols, S. Blake, F. Baker, and D.L. Black, “Definition of the Differentiated Service Field(DS Field) in the IPv4 and IPv6 headers,” RFC2474, Network Working Group, Dec. 1998.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for Differentiated Services,” RFC2475, Network Working Group, Dec. 1998.

[3] V. Jacobson, K. Nichols, and K. Poduri, “An Expedited Forwarding PHB,” RFC 2598, Network Working Group, June 1999.

[4] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB Group,” RFC 2597, Network Working Group, June 1999.

[5] D. Clark and W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service,” IEEE/ACM Transactions on Networking, vol.6, no.4, pp.362-373, Aug. 1998.

[6] S. Floyd and V. Jacobson, “On traffic phase effects in packet switched gateways,” Internetworking: Research and Experience, vol.3, no.3, pp.115-156, Sept. 1992.

[7] W. Feng, D. Kandlur, D. Saha, and K. Shin, “Understanding and improving TCP performance over networks with minimum rate guarantees,” IEEE/ACM Transactions on Networking, vol.7, no.2, pp.173-187, Apr. 1999.

[8] I.Yeom and A. L. Narasimha Reddy, “Marking for QoS Improvement,” Computer Communications, vol. 24, no. 1, pp. 35-50, Jan. 2001.

[9] 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 Transactions on Networking, vol.7, no.5, pp.685-697, Oct. 1999.

[10] R. Jain, The Art of Computer Systems Performance Analysis, John Wiley and Sons Inc., 1991.

허 경 (Kyeong Hur) 종신회원

1998년 2월 고려대 전자공학과 학사
2000년 2월 고려대 전자공학과 석사
2004년 8월 고려대 전자공학과 박사
2004년 9월~2005년8월 삼성종합기술원 전문연구원
2005년 9월~현재 경인교육대학교 컴퓨터 교육과 전임강사
<관심분야> 컴퓨터 네트워크, All-IP 네트워크, Wireless MAC 시스템