논문 04-29-1B-9 한국통신학회논문지 '04-1 Vol.29 No.1B

A Modified REDP Aggregate Marker for improving TCP Fairness of Assured Services

Kyeong Hur[*], **Doo-Seop Eom**[*], **Kyun-Hyon Tchah**[*] ***Regular Members***

ABSTRACT

To provide the end-to-end service differentiation for assured services, the random early demotion and promotion (REDP) marker in the edge router at each domain boundary monitors the aggregate flow of the incoming in-profile packets and demotes in-profile packets or promotes the previously demoted in-profile packets at the aggregate flow level according to the negotiated interdomain service level agreement (SLA). The REDP marker achieves UDP fairness in demoting and promoting packets through random and early marking decisions on packets. But, TCP fairness of the REDP marker is not obvious as for UDP sources. In this paper, to improve TCP fairness of the REDP marker, we propose a modified REDP marker where we combine a dropper, meters and a token filling rate configuration component with the REDP marker. To make packet transmission rates of TCP flows more fair, at the aggregate flow level the combined dropper drops incoming excessive in-profile packets randomly with a constant probability when the token level in the leaky bucket stays in demotion region without incoming demoted in-profile packets. Considering the case where the token level cannot stay in demotion region without the prior demotion, we propose a token filling rate configuration method using traffic meters. By using the token filling rate configuration method, the modified REDP marker newly configures a token filling rate which is less than the negotiated rate determined by interdomain SLA and larger than the current input aggregate in-profile traffic rate. Then, with the newly configured token filling rate, the token level in the modified REDP marker can stay in demotion region pertinently for the operation of the dropper to improve TCP fairness. We experiment with the modified REDP marker using ns2 simulator for TCP sources at the general case where the token level cannot stay in demotion region without the prior demotion at the negotiated rate set as the bottleneck link bandwidth. The simulation results demonstrate that through the combined dropper with the newly configured token filling rate, the modified REDP marker also increases both aggregate in-profile throughput and link utilization in addition to TCP fairness improvement compared to the REDP marker.

Keyword : QoS, DiffServ, Assured Services, TCP, Fairness

Ⅰ. INTRODUCTION

With the proliferation of multimedia and real time applications, it is becoming more desirable to provide certain quality of service (QoS) guarantee for Internet applications. The Differentiated Services (DiffServ) architecture has been proposed as a scalable way of providing QoS in the Internet [1]. Scalability is achieved by moving complicated functionalities such as Per-flow or flow aggregate marking, shaping, and policing toward the edge router and leaving the core router with very simple functionality. With DiffServ, packets are marked at the ingressive edge router of the network with a DiffServ codepoint (DSCP) [2] and at the core router they are given a forwarding treatment according to their DSCP. Each DSCP corresponds to a Per-Hop Behavior (PHB). DiffServ provides packet level service differentiation through simple and predefined PHBs. Currently, the IETF has defined one class for Expedited Forwarding (EF) PHB [3] and four classes for Assured Forwarding (AF) PHB [4].

An Internet connection can span through a path involving one or more network domains. If we want to guarantee the end-to-end minimum

---

* 고려대학교 전자컴퓨터공학과 대학원 차세대 이동통신연구실(hkyeong@.korea.ac.kr)
논문번호 : 030329-0801, 접수일자: 2003년 8월 1일

throughput of the connection, we have to make sure that the aggregate traffic along the path does not exceed any of the interdomain negotiated service level agreements (e.g., the traffic rate) after this flow joins. This is very hard to ensure since the interdomain service level agreement (SLA) is not usually renegotiated at the initiation of each new connection. For assured services, the interdomain traffic rates are usually negotiated statically based on statistical estimation or updated periodically to avoid signaling overhead and scalability problem. So, the instantaneous aggregate in-profile traffic rate may be higher or lower than the negotiated rate determined by interdomain SLA [5]. To provide the end-to-end service differentiation in DiffServ model, it is needed to extend the per-hop behavior over multiple network domains. Then, in the case of higher incoming aggregate in-profile traffic rate, the intermediate marker in the edge router demotes some of the "in" packets to "out" so that the aggregate rate of "in" packets conform to the negotiated rate. The initial marking of packets at host markers can be done on a per-flow basis. But in order to implement the per-flow marking in a intermediate marker, all the intermediate markers should know the original contracted rate of each flow and tokens assigned to each flow should be proportional to its original contracted rate. So, the per-flow marking in the intermediate marker needs per-flow monitoring and signalling where scalability problem occurs. Then, the intermediate marking must be done on the aggregate flow level for the ease of scalability. Furthermore, the demotion, although exercised at the aggregate flow level, should affect all the connections proportionally to their current usage (i.e. fair demotion). On the other hand, if the incoming aggregate in-profile traffic rate is lower, ideally the intermediate marker shou

In the random early demotion and promotion (REDP) marker for assured services, identification of a previously demoted "in" packet is ensured using the AF PHB specified packet markings. In order to support this, the REDP intermediate marker in the edge router uses a three-color (red, yellow, and green) marking process [4], where yellow is used as an indicator for temporary demotion. In promoting packets, it provides better assured services than the two-color promotion through the tricolor promotion [5]. It uses two AF classes to isolate UDP and TCP traffics. The REDP marker was proposed to remove the phase effect of periodical flows by detecting the arriving rate of green packets early and by demoting or promoting packets randomly at the aggregate flow level [5], [6]. It uses a leaky bucket with a token filling rate equal to the negotiated traffic rate determined by interdomain SLA. The instantaneous token level in the leaky bucket belongs to one of three regions which are demotion, balanced and promotion regions. In the demotion region, before the leaky bucket is emptied, incoming green packets are demoted to yellow packets randomly with a probability which is inversely proportional to the number of remaining tokens that have not be consumed by arriving green packets. In the promotion region, before the leaky bucket is overflowed, incoming yellow packets are promoted to green packets randomly with a probability proportional to the number of remaining tokens.

The early and random marking decision of the REDP marker removes the deterministic phase effect common for UDP sources that brings about unfairness when using the leaky bucket marker [5]. Consequently, the REDP marker demotes green packets of each flow proportionally to its current green transmission rate through the early and random marking decisions on packets. Therefore, in the case of demotion for UDP sources having the same contracted rate, a fair amount of yellow packets could be generated to each UDP flow through the REDP marker because those UDP flows have the same current green transmission rate. And if the aggregate contracted rate of UDP flows is larger than the bottleneck link bandwidth on their path and the negotiated rate is set as the bottleneck link bandwidth, it is sure that the demotion occurs and the aggregate green packets of UDP flows uses completely the bottleneck link where almost all the yellow and red packets would be dropped by the RIO buffer [7]. Then, each UDP flow get a fair throughput composed of almost geen packets. Otherwise if the aggregate contracted rate of UDP flows is larger than the bottleneck link bandwidth and the negotiated rate is set smaller than the bottleneck link bandwidth, each UDP flow get a fair throughput composed of a fair amount of geen and that of yellow packets. This UDP fairness in demotion of the REDP marker results in the UDP fairness in promotion where a fair amount of yellow packets of each UDP flow through the prior demotion are promoted to a fair amount of green packets through the early and random marking decision. The UDP fairness of the REDP marker is almost as good as the per-flow marking while it works at the aggregate flow level without any scalability problem [5]. However, the REDP marker cannot remove the phase effect completely for TCP sources. As a result, TCP fairness of the REDP marker is not obvious as for UDP sources [5].

In the case that TCP sources, having a same contracted rate and different connection round trip times (RTT), share the bottleneck link bandwidth through the RIO buffer, each TCP flow will have an unfair green transmission rate due to the TCP's congestion control worked with a different RTT. And if the REDP marker demotes green packets of those TCP flows with unfair green transmission rates, an unfair amount of yellow packets having a larger drop probability than green packets will be generated to each TCP flow. Even if each TCP flow has the same green transmission rate, TCP fairness is not obvious because the bottleneck link cannot be utilized completely by green packets of those TCP flows due to the TCP's congestion control and then some amount of red packets of TCP flows having different packet transmission rates would not be dropped at the bottleneck link. Consequently, the unfair packet transmission rate due to TCP's congestion control results in TCP unfairness of the REDP marker in demotion and promotion. Therefore, to improve TCP fairness of REDP marker, a controller which can make packet transmission rates of TCP flows fair is needed. Then, more fair packet transmission rates of TCP flows than those using only the REDP marker can improve TCP fairness of the REDP marker in demotion and promotion.

In this paper, to improve TCP fairness of the REDP marker, we propose a modified REDP marker where we combine a dropper, meters and a token filling rate configuration component with the REDP marker. The modified REDP marker performs a dropping in the demotion at a domain boundary only if there is no prior demotion. Otherwise if there is a prior demotion, it performs the REDP marking process. The concatenate dropping at multiple domains is avoided to manifest the effect of a dropping at a domain boundary on TCP fairness. At the aggregate flow level the combined dropper drops incoming excessive green packets randomly with a constant probability when the token level in the leaky bucket stays in demotion region without incoming yellow packets. Then, incoming some green packets of each TCP flow will be dropped proportionally to its current green transmission rate. So, the packet transmission rates and green transmission rates of TCP flows can be more fair and TCP fairness can be improved through the combined dropper.

Considering the case where the token level cannot stay in demotion region without the prior demotion, we propose a token filling rate configuration method using traffic meters. By using the token filling rate configuration method, the modified REDP marker newly configures a pertinent token filling rate which is less than the negotiated rate determined by interdomain SLA and is larger than the current input aggregate green rate. Then, with the newly configured token filling rate, the token level in the modified REDP marker can stay in demotion region pertinently for the operation of the dropper to improve TCP fairness. Input traffic meters in the modified REDP marker are Green-meter, Red-meter, and Yellow-meter. Each input traffic meter measures the incoming aggregate rate of each colored packets for a time interval. An output Yellow-meter is also used for identification of the demotion. From measured values by the combined meters for a time interval, if there is no incoming yellow packets, that means there is no prior demotion improving TCP fairness and then the modified REDP marker newly set the token filling rate as a smaller value between the negotiated rate and the pertinent token filling rate computed from measured values by input Green and Red meters considering utilization of bottleneck link.

We have experimented with the modified REDP marker using *ns2* simulator at the general case for TCP sources where the token level cannot stay in demotion region without the prior demotion at the negotiated rate set as the bottleneck link bandwidth. The simulation results show that through the combined dropper with the newly configured token filling rate, the modified REDP marker also increases both aggregate in-profile throughput and link utilization in addition to TCP fairness improvement compared to the REDP marker.

The rest of the paper is organized as follows. Section 2 proposes a modified REDP marker to improve TCP fairness of the REDP marker. The REDP marker is also explained in this section. Section 3 studies the performance of the modified REDP marker using ns2 network simulator. Section 4 is devoted to concluding remarks.

Ⅱ. REDP Marker

In this section, we propose a modified REDP intermediate marker to improve TCP fairness of the REDP marker. The REDP scheme uses a variation of the tricolor marking model for the tricolor promotion. Therefore, each packet can be marked as green, yellow, or red. Suppose an end user submits an expected rate r. Initially, the local domain configures a leaf marker for the flow. A packet of this flow is marked as green if it is in-profile and red if it is out-of-profile. None of the packets is marked as yellow. While crossing a domain boundary through the REDP marker, a green packet is demoted to yellow if the aggregate green packet rate exceeds the negotiated rate at the intermediate marker. A yellow packet is promoted to green if the aggregate green packet rate is lower than the negotiated rate. A yellow packet is never demoted to red and a red packet is never promoted to yellow. Thus, yellow is specifically used to memorize the demoted green packets. In promoting packets, the REDP marker provides better assured services than the two-color promotion through the tricolor promotion [5]. Figure 1 shows the state diagram of the demotion-promotion algorithm in the REDP marker.

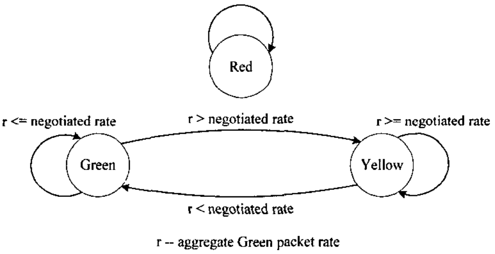


Fig.1. State diagram of demotion and promotion.

The REDP marker introduces randomness and early decisions on the packet marking process at the aggregate flow level to remove the phase effect that could bring about the unfairness in the demotion and promotion among different microflows [5], [6]. The REDP marker is implemented using a leaky bucket where the token filling rate R is equal to the negotiated rate determined by interdomain SLA. A promotion threshold is set in the leaky bucket. If the tokens in the leaky bucket exceed the promotion threshold and an arriving packet is yellow, it is randomly promoted to green. Similarly, a demotion threshold is used in the leaky bucket. If the number of tokens in the leaky bucket is less than the demotion threshold, an arriving green packet is randomly demoted to yellow. Using this scheme, the REDP marker can also detect whether the aggregate rate of the arrival of green packets is lower or higher than the negotiated rate. The REDP marking model is shown in Fig.2. Two thresholds, T_L and T_H, divide the leaky bucket into three regions-demotion, balanced, and promotion regions. If the arriving rate of green packets is equal to the token filling rate R, the token consumption rate is the same as the token filling rate. Therefore, the number of tokens in the bucket remains in the balanced region and each packet is forwarded without changing the color. If the arriving rate of green packets exceeds R, the token consumption rate exceeds the token filling rate. Then, the token level in the bucket falls into the demotion region. In the demotion region, each arriving green packet is randomly demoted to yellow with a probability of P_{demo}, where P_{demo} is a function of the token count in the leaky bucket TK_{num} as shown in Eq.(1). In Eq.(1), MAX_{demo} is the maximum demotion rate. When the leaky bucket runs out of tokens, each arriving green packet is demoted to yellow. If the arriving rate of green packets is less than R, the token filling rate exceeds the token consumption rate and the token level reaches the promotion region. In the promotion region, each arriving yellow packet will be randomly promoted to green with a probability of P_{promo}, where P_{promo} is a function of TK_{num} as shown in Eq.(1).

$$\begin{aligned} P_{demo} &= (T_L - TK_{num}) \cdot MAX_{demo} / T_L \\ P_{promo} &= (TK_{num} - T_H) \cdot MAX_{promo} /(b - T_H) \end{aligned} \qquad (1)$$

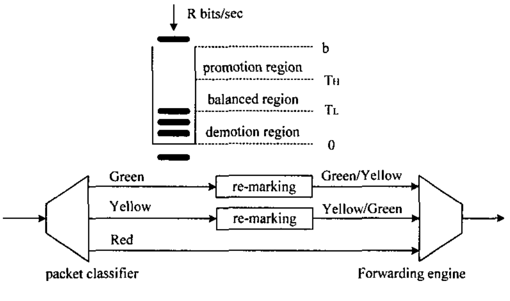


Fig.2. REDP Marker

The DiffServ core routers could support either two or three drop precedences. If it supports two drop precedences (e.g., RIO), green is deemed as "in". Both yellow and red are deemed as "out" [7]. In this paper, we consider the case that it supports two drop precedences. As described before, the REDP marker was proposed to remove the phase effect of periodical flows by detecting the arriving rate of green packets early and by promoting or demoting packets randomly. The early and random marking decision of the REDP marker removes the deterministic phase

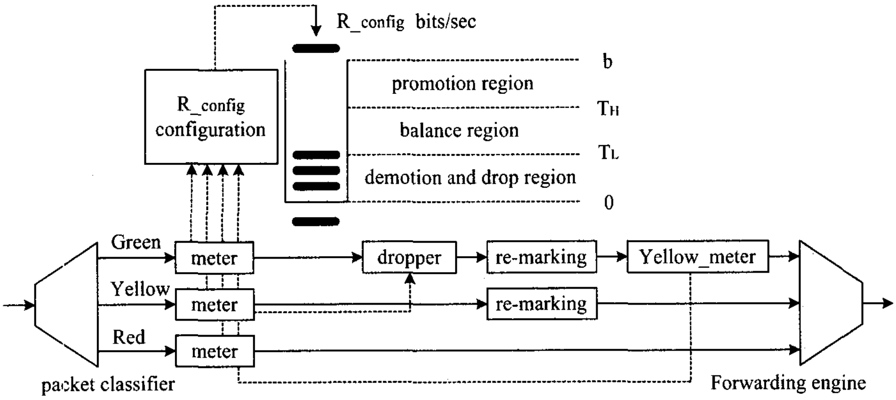


Fig.3. Modified REDP Marker

effect common for UDP sources which generate constant data rates. Consequently, the REDP marker demotes green packets of each flow proportionally to its current green transmission rate through the early and random marking decisions on packets. As a result, UDP fairness of the REDP marker in demoting and promoting packets is almost as good as the per-flow marking while it works at the aggregate flow level without any scalability problem. However, the REDP marker cannot remove the phase effect completely for TCP sources and TCP fairness of the REDP marker is not obvious as for UDP sources [5].

Ⅲ. Modified REDP Marker

Proposed modified REDP marker is shown in Fig.3. In modified REDP marker, to improve TCP fairness of the REDP marker, the REDP marker is combined with a dropper. At the aggregate flow level the dropper drops incoming excessive green packets randomly with a constant probability when the token level in the leaky bucket stays in demotion region without incoming yellow packets. If TCP sources with a same contracted rate and different connection RTTs share the bottleneck link bandwidth through the RIO buffer, each TCP flow will have an unfair green transmission rate through the bottleneck link. And if the REDP marker demotes green packets of those TCP sources with unfair green transmission rates, it will demote green packets of each TCP flow proportionally to each TCP flow's current unfair green transmission rate. To improve TCP fairness in demoting green packets, yellow packets should be generated to each TCP flow proportionally to its current green transmission rate and a portion of those yellow packets of each TCP flow proportional to that rate should be dropped at the RIO buffer. So, the packet transmission rates and green transmission rates of those TCP flows can be more fair. Then, the REDP marker will demote green packets of each TCP flow proportionally to each TCP flow's more fair green transmission rate. But, those yellow packets of each TCP flow are randomly dropped at the RIO buffer with a variable probability which is based on congestion state at the bottleneck link [7]. Then, a yellow packet may or may not get dropped depending on the actual network traffic and those yellow packets of each TCP flow would not be dropped proportionally to its current green transmission rate. But, the combined dropper drops incoming excessive green packets with a constant probability P_{drop} at demotion region. Therefore, incoming some green packets of each TCP flow will be dropped proportionally to its current green transmission rate and the modified REDP marker can improve TCP fairness in demoting green packets compared to the REDP marker. However, in both cases where the aggregate contracted rate of TCP flows is larger or smaller than the bottleneck link bandwidth with the negotiated rate set as the bottleneck link bandwidth, the demotion almost cannot occur because the bottleneck link cannot be utilized completely by green packets of those TCP flows due to the TCP's congestion control. So, the operation of the dropper to improve TCP fairness will be not performed in those cases.

Then, considering the case where the token level cannot stay in demotion region without the prior demotion, we propose a token filling rate configuration method using traffic meters as shown in Fig.3 and Fig.4. In the modified REDP marker, the demotion means the improvement of TCP fairness. Input traffic meters in modified REDP marker are Green-meter, Red-meter, and Yellow-meter. Each input traffic meter measures the incoming aggregate rate of each colored packets for a time interval. An output Yellow-meter is also used for identification of the demotion. Using traffic meters the modified REDP marker ensures the configuration of a pertinent token filling rate for the demotion of TCP green packets. For example, if the token level in the leaky bucket stays in promotion region without any incoming yellow packets for a time interval, the token filling rate of the leaky bucket is newly set by the token filling rate configuration method. Then, the token level in the modified REDP marker can stay pertinently in demotion region with the newly configured token filling rate. Now, through dropping green packets randomly in the demotion region, the modified REDP marker improves TCP fairness. In proposed token filling rate configuration algorithm, we express the negotiated rate determined by interdomain SLA as R_{nego} and the configured token filling rate of the leaky bucket as R_{config}. In the REDP marking process, the R_{config} is always equal to the R_{nego}. The pertinent token filling rate for the demotion, which is computed by the token filling rate configuration component in Fig.3, is expressed as R_{comp}. If the R_{config} is newly set by the token filling rate configuration component through modified REDP marking process, we set the value of Id_{config} as 1. Otherwise the Id_{config} is set as 0. We define a time interval having $n\tau$ duration and we observe the number of tokens in the leaky bucket every τ time.

Initially, Id_{config} is set as 0 and conventional REDP marking process is performed for a time interval with the R_{config} equal to R_{nego}. If the input yellow traffic rate for the time interval from the input Yellow-meter is zero, it means that there is no prior demotion and fairness is not improved yet. So, the demotion for improving TCP fairness through the modified REDP marking process is needed. Therefore, the modified REDP marking process will be performed during the next time interval. If the input yellow traffic rate is nonzero, it means that there is the prior demotion through the modified REDP marker. Then, the conventional REDP marking process will be performed during the next time interval. We can convert the modified REDP marking process to the REDP marking process through making the dropper off and equating R_{config} to R_{nego} with changing Id_{config} to 0. The modified REDP marker performs a dropping through the demotion at a domain boundary. The concatenate dropping at multiple domains is avoided in proposed token filling rate configuration algorithm to manifest the effect of a dropping at a domain boundary on TCP fairness. At the ingressive edge router *ER* of TCP flows as shown in Fig.5, the input yellow traffic rate is zero. Therefore, at the ingressive domain of TCP flows, the demotion for improving TCP fairness through the modified REDP marker is always performed. In modified REDP marking process, if the input yellow traffic rate is zero, the value of Id_{config} is checked whether it is zero or one. If Id_{config} has been set as 0, R_{config} will be newly set to be pertinent for the demotion. In the case that Id_{config} has been set as 1, it notifies that the R_{config} was once set pertinently by the token filling rate configuration component. However, if output yellow traffic rate from the output Yellow-meter is zero, it proves that the demotion did not occur during the last time interval due to the decrease in number of TCP flows which are routed through the edge router. So, R_{config} also should be newly set to be pertinent for the demotion in this case. We set newly R_{config} through comparing R_{comp} and R_{nego}. The R_{config} cannot be larger than the R_{nego}. If the R_{comp} pertinent for the demotion is larger than the R_{nego}, R_{config} is set as R_{nego}. Otherwise it is set as R_{comp}. At both cases the Id_{config} is changed to 1. The R_{config} less than the R_{comp} ensures the occurrence of the demotion. The R_{nego} can vary every time according to the interdomain SLA negotiation and the variation of the R_{nego} implies that the aggregate input green traffic rate has changed or will change. So, we have to prevent the case where the R_{config} is not changed because the Id_{config} has been set constantly as 1 while R_{nego} has increased with the increase in the input green traffic rate. And we also have to prevent early the case where the demotion has not occurred even though the Id_{config} has been set as 1. For those purposes, the Id_{config} is also changed to 0 whenever R_{nego} changes.

The R_{comp} is computed pertinently for the demotion as follows. We used λ_{green} to denote the aggregate input green traffic rate measured during a $n\tau$ interval at the input Green-meter. λ_{red} and λ_{yellow} denote the aggregate input red and yellow traffic rates, respectively. In Fig.5 showing a DiffServ domain, there is a bottleneck link on a path with 1.2 Mbps bandwidth. Then,

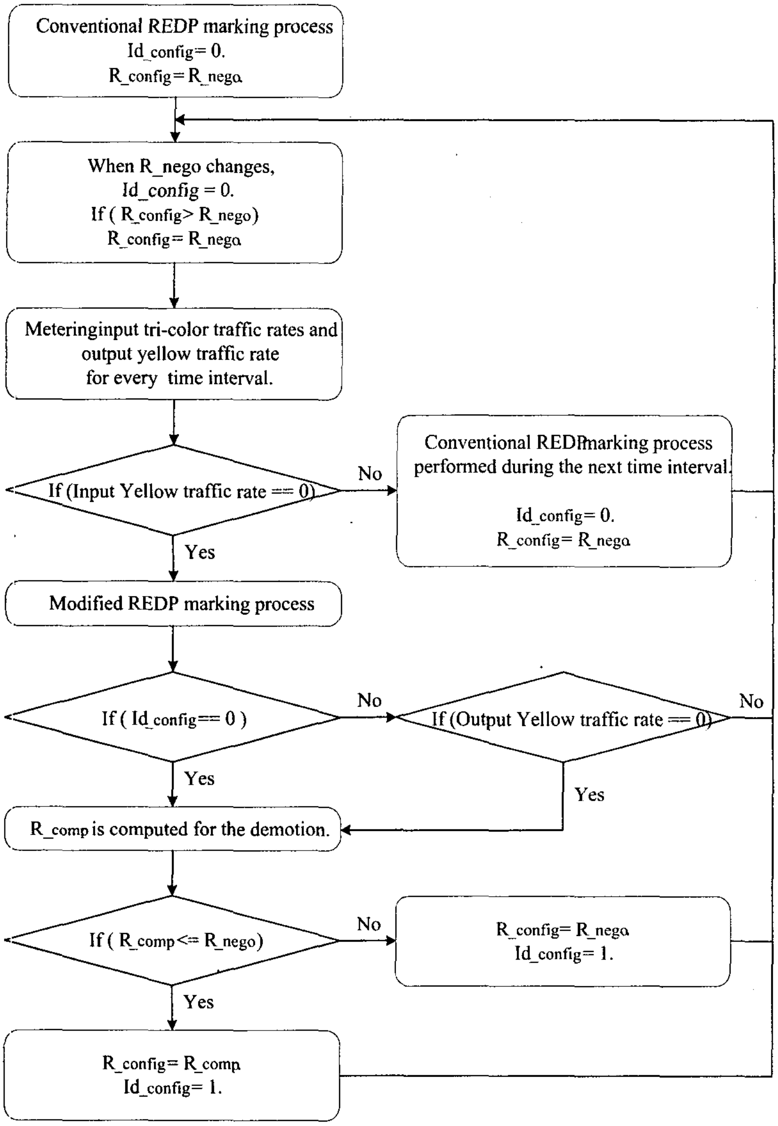


Fig.4. Token filling rate configuration algorithm.

R_{nego} can be set maximally as the 1.2 Mbps bandwidth of the bottleneck link. We used R_{max} to denote the maximum R_{nego}. For TCP sources, the aggregate TCP flows cannot utilize the bottleneck link completely due to the TCP's congestion control. So, the average utilized bandwidth by those TCP flows is smaller than the 1.2 Mbps bandwidth of the bottleneck link. If the input yellow traffic rate is zero, the sum of the λ_{green} and the λ_{red} can be approximately equal to the average utilized bandwidth of the bottleneck link during the last $n\tau$ interval. In computing R_{comp}, R_{comp} should be larger than the λ_{green} measured during the last $n\tau$ interval. If the R_{nego} is larger than the λ_{green} and the R_{config} is set as the R_{comp} which is smaller than the λ_{green}, unreasonably the modified REDP marker will demote current green packets. To avoid that situation, as shown in Eq.(2), the R_{comp} can be computed through multiplying the R_{max} by the ratio of λ_{green} to the sum of λ_{green} and λ_{red}.

Therefore, the R_{comp} is larger than the λ_{green} because the sum of λ_{green} and λ_{red} is smaller than the R_{max} equal to the bandwidth of the bottleneck link. The R_{comp} also means the λ_{green} when TCP flows utilize the bottleneck link completely. Thus, by newly setting the R_{config} as the R_{comp}, the modified REDP marker does not demote green packets until the input aggregate green traffic rate exceeds the R_{comp} considering utilization of the bottleneck link.

In assured services, it is desirable that aggregate green packets of TCP flows utilize all the bottleneck link when the aggregate contracted rate of those TCP connections is larger than the bottleneck link bandwidth. For that purpose, the modified REDP marker is also proposed to increase input green rate and the aggregate green throughput of TCP flows. Through the proposed token filling rate configuration method and the combined dropper, the modified REDP marker can increase input green rate and aggregate green throughput. If the token level stays in promotion region without any incoming yellow packets at a token filling rate of R_{nego}, R_{config} is newly set as the R_{comp} smaller than the R_{nego}. When the R_{config} is newly set as the R_{comp} larger than the λ_{green}, the modified REDP marker does not demote green packets until the aggregate green traffic rate exceeds the R_{comp}. And the token level in the modified REDP marker can stay pertinently in demotion region. Then, through dropping green packets randomly in the demotion region, firstly it improves TCP fairness compared to the REDP marker. This improved TCP fairness of the modified REDP marker results in improved TCP fairness in following promotion where a fair amount of yellow packets of each TCP flow through the combined dropper with the R_{comp} are promoted to a fair amount of green packets through the early and random marking decision of the REDP marking scheme. In the following promotion, the input yellow traffic rate will be not zero because of the generation of yellow packets at the prior demotion. Then, according to the token filling rate configuration method shown in Fig.4, the conventional REDP marking process will be performed in the modified REDP marker. Furthermore, the modified REDP marker also increases aggregate green throughput and the utilization of the bottleneck link compared to the REDP marker through the increase in the input green rate. The increase in the input green rate can be seen from the result that the token level in the modified REDP marker stays in demotion region with the R_{comp} larger than the λ_{green}. The simulation results in Section 3-A of this paper support this argument.

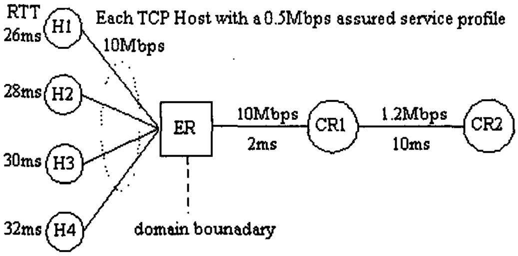
$$R_{comp} = R_{\max} \cdot \frac{\lambda_{green}}{\lambda_{green} + \lambda_{red}}, \quad at\ \lambda_{yellow} = 0$$
$$(R_{comp} > \lambda_{green} \quad for\ TCP\ flows) \tag{2}$$

IV. Performance Study

In this Section, we analyze the performance and the effectiveness of proposed modified REDP marker. Considering the general case for TCP sources where the token level cannot stay in demotion region without the prior demotion at the negotiated rate set as the bottleneck link bandwidth, we proposed a token filling rate configuration method. By adopting the proposed token filling rate configuration method, the modified REDP marker ensures the configuration of a pertinent token filling rate for the demotion where TCP fairness is improved through the combined dropper. In this paper, we assume that variation of the negotiated rate R_{nego} determined by the interdomain SLA has no relation to the token filling rate configuration method and we focused on the improvement of TCP fairness of the REDP intermediate marker. The intermediate marker in edge router is assumed to have no signalling for exchanging information with the leaf marker in each host for the ease of scalability. Then, we experiment with the modified REDP marker at two cases where the aggregate contracted rate of TCP flows is larger or smaller than the negotiated rate set as the bottleneck link bandwidth. At those cases, the demotion almost cannot occur because the bottleneck link cannot be utilized completely by green packets of those TCP flows due to the TCP's congestion control. In each case, we compare TCP fairness and aggregate throughput through experiments using the *ns2* simulator. Only TCP sources are analyzed to show the performance improvement. Finally, we briefly investigate the effectiveness of the R_{comp} with showing its impact on the performance of modified REDP marker in each case.

A. Effect of the combined dropper with token filling rate configuration method on TCP fairness

Figure 5 depicts the simulation topology used to study the effect of the combined dropper with token filling rate configuration method on TCP fairness. TCP hosts in our simulations used TCP Reno transport protocol and TCP hosts H1, H2, H3, H4 each has a leaf marker implemented inside. Each of the hosts has a 0.5 Mbps assured service profile. So initially each host could have up to 0.5 Mbps packets marked as green. The remaining packets are marked as red. Assured service is implemented in routers through the RIO scheme. In core routers, all the green packets are treated as "in," both red and yellow packets are treated as "out". We implemented a simple RIO queue [7] in *ns2* simulator. Both "in" and "out" packets are buffered in the same queue. We used two sets of RED parameters for "in" and "out" packets [9]. The RED parameters for "in" packets is : 45 packets, 60 packets, and 0.02 for min_{in}, max_{in}, and $P\max_{in}$, respectively, and 20 packets, 40 packets, and 0.05 for min_{out}, max_{out}, and $P\max_{out}$, 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 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. So almost all the green packets would be forwarded without being dropped in this configuration. The RTT (Round Trip Time) of each TCP flow is 26, 28, 30, 32 ms, respectively. Four TCP flows-tcp1, tcp2, tcp3, tcp4-originate from hosts H1, H2, H3, and H4, respectively and terminate at CR2. Throughput of each TCP flow is measured at CR2 core router. The edge router ER is located at the domain boundary. For both markers in all our simulations, we set the size of the leaky bucket, b, as 60 packets where T_L is set as 15 packets and T_H is set as 45 packets. Both MAX_{demo} and MAX_{promo} probabilities are set equally as 0.5 [5]. The time interval $n\tau$ during which modified REDP marker measures traffic rates is set as 100s where τ is set as 0.1 second and n is set as 1000. The drop probability P_{drop} in the combined dropper is set constantly as 0.02.



Marker	At ER1
REDP	$R_{config} = R_{nego} = 1.2Mbps$
Modified REDP	Modified REDP marking process($Id_{config} = 1$) $R_{config} = R_{comp} = 0.9Mbps$, $R_{comp} < R_{nego}$

Fig.5. Simulation topology used to study the effect of the combined dropper with token filling rate configuration method.

At ER, R_{neg}, is 1.2 Mbps equal to the bandwidth of the bottleneck link between CR1 and CR2. In Fig.5, the aggregate contracted green rate of TCP hosts is 2.0 Mbps larger than the R_{nego} where ideally demotion can occur. The input yellow traffic rate, measured at the modified REDP marker in ER during $n\tau$, was zero. The λ_{green} was 757 Kbps and the λ_{red} was 252 Kbps. As shown in Eq.(2), at the token filling rate configuration component, the R_{comp} is computed as 900 Kbps which is larger than the λ_{green} and is smaller than the R_{nego}. Then, by the proposed token filling rate configuration method shown in Fig.4, the R_{config} of modified REDP marker in ER is set as 0.9 Mbps equal to R_{comp} with changing Id_{config} to 1. In Fig.5, the λ_{green} of aggregate TCP flows at ER is much smaller than the bandwidth of the bottleneck link due to the TCP's congestion control even though the aggregate contracted green rate of those TCP hosts is larger than the bottleneck link bandwidth. Because the R_{nego} is larger than the λ_{green} at ER, the token level in the REDP marker cannot stay in the demotion region without the prior demotion at Fig.5. In Fig.6, to show the effect of the combined dropper with the newly configured token filling rate, we show the variations of token level in the leaky bucket respectively when using the REDP marker, REDP marker with R_{config} set as the R_{comp}, and modified REDP marker at ER in Fig.5. In Fig.6-(a), we can see that the token level in the REDP marker with the 1.2 Mbps R_{nego} stays in the promotion region with no arriving yellow packet even though the aggregate contracted green rate of TCP hosts in Fig.5 is larger than the R_{nego} set as the bottleneck link bandwidth. This result for TCP sources can occur generally and cannot be avoided due to the TCP's congestion control. Figure 6-(b) shows that the REDP marker with R_{config} set as the R_{comp} has more various regions than the REDP marker in Fig.6-(a) and generates yellow packets. Finally, in Fig.6-(c), we can see that most of time the token level in the modified REDP marker stays in the region between demotion region and balanced region and there are largest demotions among the above three cases. It means that the input green rate is increased by using the modified REDP marker because the R_{comp} is larger than the λ_{green} measured during the last $n\tau$ interval. In addition, the difference between the result of Fig.6-(b) and that of Fig.6-(c) is only the application of the dropper to the modified REDP marker. Therefore, at the case for TCP sources where the token level cannot stay in the demotion region without the prior demotion as in Fig.6-(a), the modified REDP marker makes the token level be able to stay pertinently in demotion region using proposed token filling rate configuration method and it increases input green rate through the combined dropper which works at demotion region.

The throughput for different TCP flows at Fig.5 using the REDP marker with the R_{nego} REDP marker with R_{config} set as the R_{comp}, and proposed modified REDP marker is shown in Fig.7, respectively. We show green and total throughputs of each TCP flow. In addition, aggregate throughput performances of those TCP flows at the above three cases are also compared to analyze the effect of the combined dropper with token filling rate configuration method on TCP fairness. In Fig.5, if the intermediate marker implemented in ER is ideally fair, as for UDP sources, ideally each TCP flow can get 300 Kbps green throughput. In results using the REDP marker with the R_{nego} in Figs.7-(a) and 7-(b), green and total throughputs of the four TCP flows are highly biased. This TCP unfairness occurs because TCP has its own congestion control algorithm and the RIO buffer drops arriving packets according to its congestion state [7], [10]. To improve TCP fairness, a control to make packet transmission rates of TCP flows more fair should be added. Using only the aggregate marking algorithm as the REDP at intermediate marker cannot avoid that TCP unfairness [5]. But, in results using the REDP marker with R_{config} set as the R_{comp}, fairness is improved compared to the case using the REDP marker with the R_{nego}. This results show that, at the case of Fig.5 where the token level in the REDP marker cannot stay in demotion region without the prior demotion, the REDP marker with only proposed token filling rate configuration method can provide better TCP fairness over the conventional REDP marker by demotion which can generate a minor drop effect proportional to the current packet transmission rate of each TCP flow. Furthermore, Figures 7-(a) and 7-(b) show that, by the combined dropper in the modified REDP marker, in addition to the improved TCP fairness, both green and total throughputs of each flow are increased to be approximate to 225 Kbps equal to the fair share for the R_{comp} and total 300 Kbps throughput, respectively. Figure 7-(c) compares aggregate green throughput, input green rate, and utilization of the bottleneck link at the above three cases. In Fig.7-(c), though TCP fairness is improved by using the REDP marker with R_{config} set as the R_{comp}, all the aggregate throughputs using that marker is smaller than those using the REDP marker with R_{nego}. However, in addition to the TCP fairness improvement, through the increase in input green rate shown in Fig.6-(c), the modified REDP marker also increases aggregate green throughput and utilization of the bottleneck link compared to those using the REDP marker with R_{nego}.

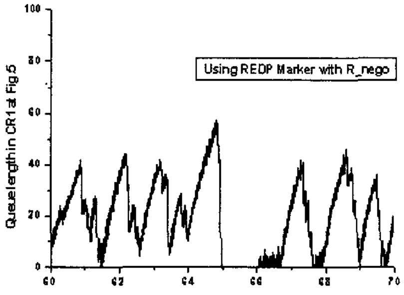


Fig.8-(a) Simulation time (sec)

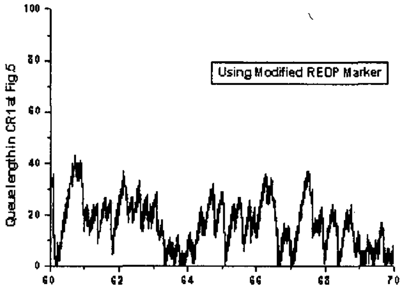
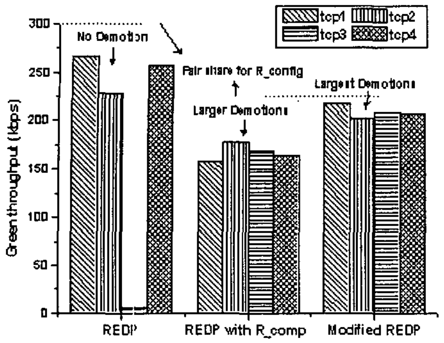


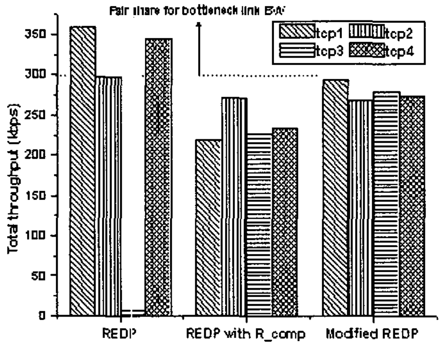
Fig.8-(b) Simulation time (sec)

Fig.8. Comparison of queue length variations

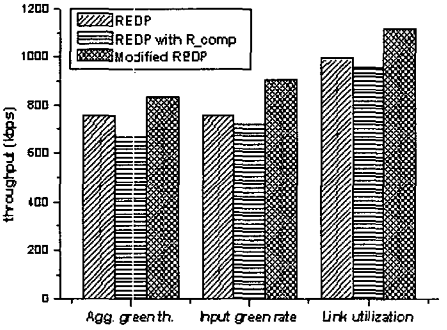
In Figs.8 and 9, we show the variations of queue length and average total queue length of RIO buffer in CR1 at Fig.5, respectively. In Fig.8-(a) using the REDP marker with R_{nego}, we can see the burst packet arrivals of TCP flows with the global synchronization due to the phase effect [6]. But, when using the modified REDP marker, the phase effect of TCP flows is reduced through the operation of combined dropper as shown in Fig. 8



(a) Green throughput

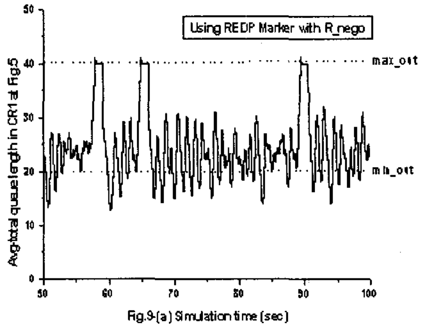


(b) Total throughput



c) Aggregate throughput

Fig.7. Comparisons of TCP fairness and aggregate throughput at Fig.5.



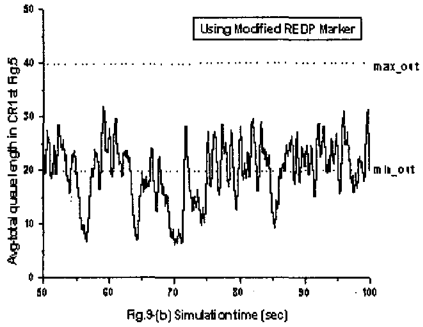
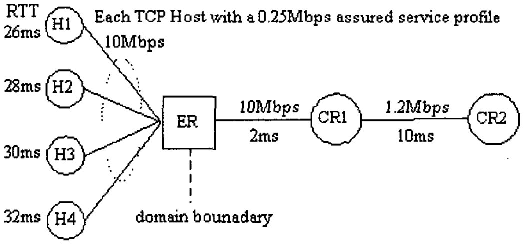


Fig.9. Average total queue length variations

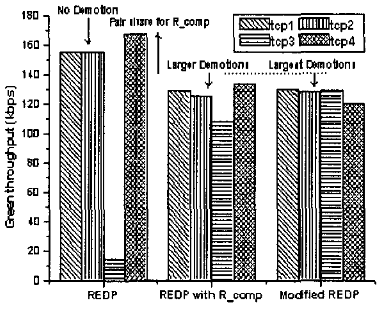


Marker	At ER1
REDP	$R_{config} = R_{nego} = 1.2Mbps$
Modified REDP	Modified REDP marking process ($Id_{config} = 1$) $R_{config} = R_{comp} = 0.57Mbps$, $R_{comp} < R_{nego}$

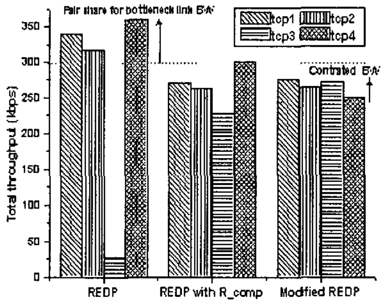
Fig.10. A case where inherently demotion cannot occur.

-(b) and then the average total queue length decreases as shown in Fig.9-(b), Therefore, by using the modified REDP marker, we can reduce the random drops at the bottleneck link through the decrease in the average total queue length. So, it can also increase the aggregate throughput of TCP flows as shown in Fig.7-(c) in addition to the improved TCP fairness by reduced TCP phase effect.

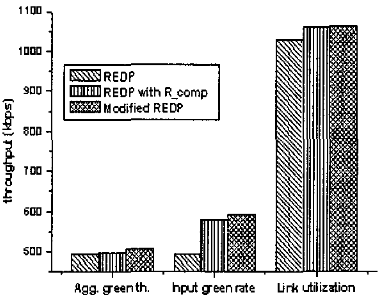
In Fig.10, we consider the case where the aggregate contracted green rate of TCP hosts is smaller than the R_{nego} where inherently demotion cannot occur and in this case we show the effect of the combined dropper with token filling rate configuration method on TCP fairness. The simulation topology in Fig.10 is identical to that in Fig.5 except that each of the TCP hosts has a 0.25 Mbps assured service profile at Fig.10. At ER, R_{nego} is 1.2 Mbps and the aggregate contracted green rate of TCP hosts is 1.0 Mbps smaller than the R_{nego}. Then, the token level in the REDP marker in ER at Fig.10 cannot stay in the demotion region without the prior demotion because the R_{nego} is larger than the aggregate contracted green rate of TCP hosts. The input yellow traffic rate, measured at the modified REDP marker in ER during $n\tau$, was zero. The λ_{green} was 494 Kbps and the λ_{red} was 546 Kbps. The R_{comp} is computed as 570 Kbps which is larger than the λ_{green} and is smaller than the R_{nego}. Then, by the token filling rate configuration method, the R_{config} of modified REDP marker in ER is set as 0.57 Mbps equal to R_{comp} with changing Id_{config} to 1.



(a) Green throughput



(b) Total throughput

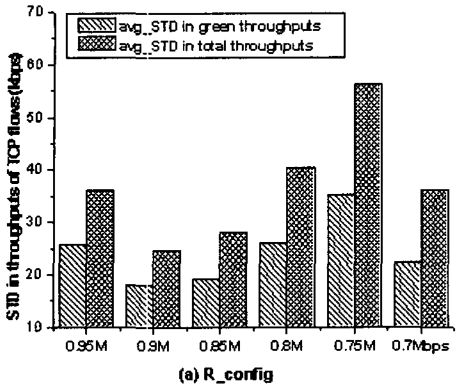


c) Aggregate throughput

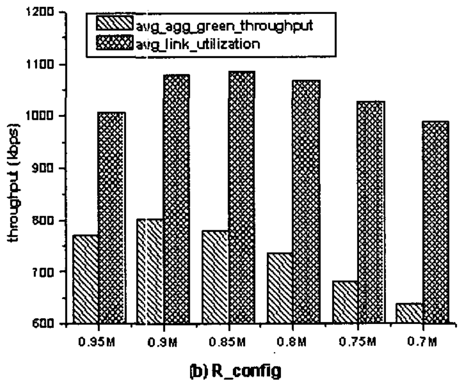
Fig.11. Fairness and aggregate throughput at Fig.10.

Figure 11 shows the throughput for different TCP flows at Fig.10 respectively when using the REDP marker with the R_{nego}, REDP marker with R_{config} set as the R_{comp}, and proposed modified REDP marker. At Fig.10, if the intermediate marker implemented in ER is ideally fair, ideally each TCP flow can get 250 Kbps green throughput. And if the REDP marker with R_{config} equal to the 570 Kbps R_{comp} is ideally Fair, ideally each TCP flow can get 142.5 Kbps green throughput as for UDP sources [5]. In results using the REDP marker with the R_{nego} in Figs.11-(a) and 11-(b), similar to the previous results, green and total throughputs of the four TCP flows are highly biased. But, in results using the REDP marker with R_{config} set as the R_{comp}, fairness is improved compared to the case using the REDP marker with the R_{nego}. This results show that, even at the case of Fig.10 where inherently the token level in the REDP marker cannot stay in demotion region without the prior demotion, the REDP marker with only token filling rate configuration method can provide better TCP fairness by the demotion with the R_{comp} larger than the λ_{green} input green rate.

Furthermore, Figures 11-(a) and 11-(b) show that, by the combined dropper in the modified REDP marker, fairness is more improved than the case using the REDP marker with R_{config} set as the R_{comp} and each TCP flow get a total throughput more than the 250 Kbps contracted rate. Figure 11-(c) compares aggregate green throughput, input green rate, and utilization of the bottleneck link at the above three cases. In Fig.11-(c), all the aggregate throughputs when using the REDP marker with R_{config} set as the R_{comp} or the modified REDP marker are larger than those using the REDP marker with R_{nego}. However, the increase in the aggregate throughputs is not obvious as that in the case of Fig.5. From simulation results, we show that, for cases that there is enough R_{nego} or not enough R_{nego} to forward all the aggregate contracted green packets as green at the edge router, through the combined dropper with proposed token filling rate configuration method in the modified REDP marker, TCP fairness is improved compared to the REDP marker without any scalability problem. Furthermore, if the leaf markers implemented in each host are configured based on the capacity of the first intermediate marker, as the result shown in Fig.11-(b), each TCP flow can get its contracted rate using the proposed modified REDP marker.

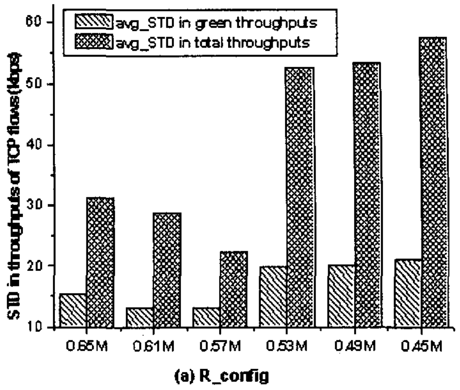


(a) R_config

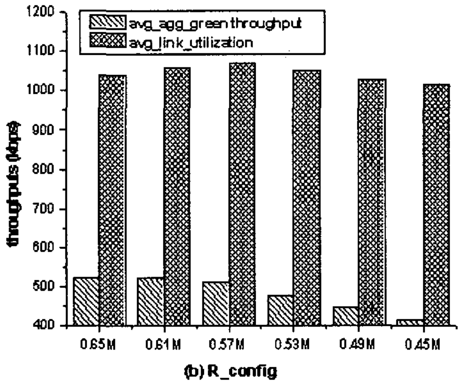


(b) R_config

Fig.12. Effectiveness of R_{comp} at Fig. 5.



(a) R_config



(b) R_config

Fig.13. Effectiveness of R_{comp} at Fig. 10.

B. Effectiveness of R_{comp} for the modified REDP marker on TCP fairness

In this Subsection, we briefly investigate the effectiveness of the R_{comp} for the modified REDP marker on TCP fairness through the simulation at each case considered in the above Subsection. In Fig.12-(a), to show the effectiveness of the R_{comp}, 0.9 Mbps, at Fig.5, we compare the average standard deviation value in green throughputs and that in total throughputs of TCP flows for each P_{drop} at each R_{comp} candidate. The standard deviation of the throughputs defines the degree of fairness and the standard deviation is abbreviated as STD in this paper. The average STD value in total throughputs for each P_{drop} is calculated through averaging STD values from results using different five P_{drop} probabilities such as 0.01, 0.02, 0.03, 0.04, and 0.05 with a R_{comp} candidate at Fig.5. That in green throughputs for each P_{drop} is also calculated in the same way. In Fig.12, we considered several R_{comp} candidates at Fig.5 such as 0.95 Mbps, 0.9 Mbps, 0.85 Mbps, 0.8 Mbps, 0.75 Mbps, and 0.7 Mbps which are smaller than the 1.2 Mbps R_{nego}.

In addition, Figure 12-(b) compares the average aggregate green and total throughputs of TCP flows for each P_{drop} at each R_{comp} candidate in Fig.5. We can find that the R_{comp}, 0.9 Mbps, which is larger than the 757 Kbps λ_{green} at Fig.5 and is calculated at the combined token filling rate configuration component in modified REDP marker, belongs to a effective value where the token level in the modified REDP marker can stay in demotion region pertinently for the performance improvement because each STD value in Fig.12-(a) is lower than others and each aggregate throughput in Fig.12-(b) is also higher than others when the R_{config} is set as 0.9 Mbps.

Figure 13-(a) shows the effectiveness of the R_{comp}, 0.57 Mbps at Fig.10 by comparing the average STD value in green throughputs and that in total throughputs of TCP flows for each P_{drop} at each R_{comp} candidate which are calculated in the same way as Fig.12. In Fig.13, we considered several R_{comp} candidates at Fig.10 such as 0.65 Mbps, 0.61 Mbps, 0.57 Mbps, 0.53 Mbps, 0.49 Mbps, and 0.45 Mbps which are smaller than the 1.2 Mbps R_{nego}. In addition, Figure 13-(b) compares the average aggregate green and total throughputs of TCP flows for each P_{drop} at each R_{comp} candidate at Fig.10. We can find that the R_{comp}, 0.57 Mbps, which is larger than the 494 Kbps λ_{green} at Fig.10, also belongs to a effective value for the performance improvement of the modified REDP marker because each STD value in Fig.13-(a) is a lower one and each aggregate throughput in Fig.13-(b) is also a higher one when the R_{config} is set as 0.57 Mbps. From results in Figs.12-(b) and 13-(b), we can see that the lower R_{config}, which is smaller than the R_{comp}, results in the lower average aggregate green throughput that also may result in the lower average utilization of the bottleneck link. From the simulation results in this Subsection for the cases of Fig.5 and Fig.10, it is inferred that the R_{comp} have the global availability for the performance improvement of proposed modified REDP marker.

V. CONCLUDING REMARKS

In this paper, to improve TCP fairness of the REDP marker, a modified REDP marker is proposed where we combine a dropper, meters and a token filling rate configuration component with the REDP marker. To make packet transmission rates of TCP flows more fair, at the aggregate flow level the combined dropper drops incoming green packets randomly with a constant probability when the token level stays in demotion region without the prior demotion. On the other hand, considering the general case where the token level cannot stay in demotion region without the prior demotion at the negotiated rate set as the bottleneck link bandwidth, we proposed a token filling rate configuration method.

By adopting the proposed token filling rate configuration method, the modified REDP marker ensures a pertinent token filling rate configuration for the demotion by which TCP fairness is improved. The modified REDP marker performs a dropping through the demotion at a domain boundary only if there is no prior demotion. Otherwise if there is a prior demotion, it performs the REDP marking process through the token filling rate configuration method. Therefore, at the ingressive domain of TCP flows, the demotion for improving TCP fairness through the modified REDP marker is always performed. Simulation results show that by the effect of the combined dropper with token filling rate configuration method, the modified REDP marker also increases the aggregate green throughput and the utilization of the bottleneck link compared to the REDP marker in addition to the improvement of TCP fairness. The effectiveness of the R_{comp} is also investigated with showing its impact on the performance of the modified REDP marker. From the simulation results, it is inferred that the R_{comp} calculated by the token filling rate configuration method is an effective value for the performance improvement of the modified REDP marker. Then, if we use the REDP marker for the AF class of UDP flows and we use the modified REDP marker for the AF class of TCP flows, the fairness in assured services can be provided without scalability problem.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC2475, *IETF*, 1998.

[2] 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, *IETF*, Dec. 1998.

[3] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," RFC 2598, *IETF*, June 1999.

[4] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," RFC 2597, *IETF*, June 1999.

[5] F. Wang, P. Mohapatra, S. Mukherjee, and D. Bushmitch, "A Random Early Demotion and Prmotion Marker for Assured Services," *IEEE Journal of Selected Areas on Communications*, vol.18, no.12, pp.2640-2650, Dec. 2000.

[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] 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.

[8] UCB/LBNL/VINT, "Network Simulator-ns (ver sion2)," *http://www-mash.cs.berkeley.edu/ns*, 1998.

[9] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol.1, no.4, pp.397-413, Aug. 1993.

[10] 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.

허 경(Kyeong Hur) 정회원

1998년 2월: 고려대학교 전자공학과 학사
2000년 2월: 고려대학교 전자공학과 석사
2000년 3월~현재: 고려대학교 전자공학과 박사과정 재학 중

<주관심분야> 통신네트워크 설계 및 성능분석, IP 네트워크, 이동 멀티미디어 시스템

엄 두 섭(Doo-Seop Eom) 정회원

1987년 2월: 고려대학교 전자공학과 학사
1989년 2월: 고려대학교 전자공학과 석사
1999년 3월: 일본오사카대학 정보통신공학과 박사
1989년 2월~1999년 8월: 한국전자통신연구소 연구원
1999년 9월~2000년 8월: 원광대학교 전임강사
2000년 9월~현재: 고려대학교 전자공학과 부교수

<주관심분야> 통신네트워크 설계 및 성능분석, 무선 Ad-hoc 망, IP 네트워크

차 균 현(Kyun-Hyon Tchah) 정회원

1965년 2월: 서울대학교 전기공학과 학사
1967년 6월: 미국 일리노이 공과대학 석사
1976년 6월 : 서울대학교 전자공학과 박사
1977년 3월~현재: 고려대학교 전자공학과 교수
1998년 1월~1998년 12월: 한국통신학회 회장
1998년 4월~현재: 한국전자통신연구원 부이사장

<주관심분야> 통신 이론, 이동 통신, 위성 통신, 이동 멀티미디어 시스템