A modified RIO queue management scheme that reduces the bandwidth skew problem in Assured Service

Hyogon Kim**, Won Hyoung Park* and Saewoong Bahk*
**Ajou University
hyogon@madang.ajou.ac.kr
*Seoul National University
{pwh, sbahk}@netlab.snu.ac.kr

***Abstract*—In offering a statistical end-to-end bandwidth guarantee service, typically called Assured Service, in Differentiated Services (Diff-Serv) framework, the biggest issue is its inconsistency. Larger profile TCP flows fail to achieve the guaranteed rate when competing with many smaller profile flows. This phenomenon, which we call "bandwidth skew", stems from the fact that larger profile flows take longer time to recover from the congestion window size backoff after a packet drop. Proposed solutions to this problem, therefore, are focused on modifying the TCP behavior. However, TCP modification is not practicable, mainly due to its large installation base. We look to other mechanisms in the Diff-Serv framework to find more realistic solutions. In particular, we demonstrate that RIO, the de facto standard packet differentiation mechanism used for Assured Service, also contributes to the bandwidth skew. Based on this new finding, we design a modified RIO mechanism called RI+O. RI+O uses OUT queue length in addition to IN and IN+OUT queue lengths to calculate OUT packet drop probability. We show through extensive simulation that RI+O significantly alleviates the bandwidth skew, expanding the operating regime for Assured Service.**

***Keywords*— Differentiated Services, Quality of Service, throughput guarantee, RIO, TCP**

1. INTRODUCTION

The current Internet delivers only the best-effort (BE) service. There is a demand to provide a variety of performance guarantees. Recently, the IETF Differentiated Services working group proposed the Differentiated Services Architecture, relatively simple and coarse methods of providing differentiated classes of service for Internet traffic, to support various types of applications and specific business requirements.

The Differentiated Services approach to providing Quality of Service (QoS) in networks employs a small, well-defined set of building blocks from which a variety of services may be built. A small bit-pattern in each packet, in the IPv4 TOS octet or the IPv6 Traffic Class octet, is used to mark a packet to receive a particular forwarding treatment, or per-hop behavior (PHB), at each network node. The Working Group standardized a small number of specific per-hop behaviors, the Expedited Forwarding (EF) PHB (RFC 2598) [3] and the Assured Forwarding (AF) PHB group (RFC 2597) [4].

The AF PHB group is a means for a provider DS domain to offer different levels of forwarding assurances for IP packets received from a customer DS domain. The customer can expect an assurance that the IP packets sent are forwarded with high probability as long as the traffic does not exceed the subscribed information rate (service profile). The packets that exceed the subscribed profile rate are also forwarded but not as high probability as the traffic that is within the profile rate. This service was originally proposed by Clark and Fang[6], and it is referred to as Assured Service (AS) in this paper.

Classified "in profile" or "out of profile" at the network boundary, the IP packets receive different treatment from IP routers in the Diff-Serv network. The AF PHB group RFC suggests using an active queue management algorithm like RIO (RED with IN (in-profile) and OUT (out-of-profile)) proposed by Clark and Fang in [6] in the routers. RIO uses twin RED algorithms for dropping packets, one for INs and the other for OUTs. Upon each packet arrival at the router, the router checks whether the packet is marked as IN or OUT. If it is an IN packet, the router calculates *avg_in*, the average queue size for the IN packets; if it is an OUT packet, the router calculates *avg_total*, average total queue size for both IN and OUT arriving packets. The probability of dropping an IN packet depends on *avg_in*, and the probability of dropping an OUT packet depends on *avg_total*. There are three parameters for each of the twin algorithms. We will call these sets of parameters as IN parameters ($min\_in/max\_in/P_{max\_in}$) and IN+OUT parameters ($min\_total/max\_total/P_{max\_total}$), respectively. In addition, the OUT parameters are used in our modified RIO mechanism. RIO mechanism can be extended easily to three levels of drop precedence or more.

2. PROBLEMS OF ASSURED SERVICE

Ibanez and Nichols [12] showed that Assured Service does not provide clearly defined and consistent rate guarantees. In particular, they showed that if there are flows having different profile rates, together with best-effort flows, the flows with small profile rates reach or exceed their profile rates while the flows with large profile rates cannot reach their profile rates. This problem, that we call "bandwidth skew", is significant because if the rate guarantee of Assured Service is severely violated, the Differentiated Services cannot be deployed as the IETF plans. So several people proposed some schemes to solve these problems. Feng et al. [14] proposed a two-window TCP scheme which modifies the TCP congestion avoidance algorithm. In this scheme, the TCP congestion window is differently adjusted depending on the detected marking of dropped packets. Yeom and Reddy [13] studied schemes such as limiting OUT packets, inverse-rate drop policy, three drop precedences (not the same way as AF PHB), and two-windows TCP. In the limiting OUT packets sceme, when the sender gets the information that one of its packets is marked OUT from the marker, it reduces the window by a packet. Inverse-rate drop policy requires that every packet is stamped with the profile rate and the higher the profile rate, the lower the probability for dropping a packet of that flow. In the three drop precedences scheme, the marker keeps track of the long-term sending rate of the sender. When a packet has to be marked as OUT, if the-long-term sending rate is greater than the profile rate the packet is marked as OUT-OUT, otherwise OUT-IN.

Unfortunately, these proposals are not practical enough because they require modifications of the existing TCP mechanisms [13, 14], the signaling scheme between a sender and a router [13], and/or an additional IP header field [13]. In this paper, therefore, we seek to find solutions that do not require significant changes in the already deployed infrastructure. We try modifying only the newly implemented components of the Differentiated Services architecture, in particular, the RIO mechanism. Although we cannot solve the bandwidth skew problem for all conceivable situations, we can expand the regime where the Assured Service is feasible.

3. THE MODIFIED RIO

It is undeniable that the inherent nature of TCP window size backoff after packet drop is the source of the bandwidth skew problem. However, the role of RIO in the problem has been glanced over. We find that it significantly contributes to the discrimination of larger profile flows. No matter how the IN+OUT parameters are set, RIO cannot strongly control the throughputs generated by OUT packets for different profile flows. This is because both IN and OUT packets are used to compute *avg_total*, and the composition of IN and OUT packets differ for different profile flows. It is likely that for a large profile flow, the OUT packets arrive immediately following many IN packets of the same flow due to the large congestion window size. So when OUT packets begin to be generated, *avg_in* is closer (than when smaller profile flows transmit) to the *avg_total* value that causes packet drop. The larger profile flows are likely to face with larger OUT packet drop probabilities, resulting in the unfairness of throughputs of the OUT packets. Larger profile flows can further suffer due to their larger window sizes. After the packet drop and window back-off, the time in which the flow can recover its original window size is longer. Meanwhile, the window size is not large enough to send IN packets at its full profile rate. On the other hand, the flows with small profile rates and best-effort flows can have many chances to send packets enough to get the subscribed bandwidth plus some excess bandwidth that the flows with large profile rates missed. So the distribution of the throughputs of IN packets among the flows can become unfair as well.

```
For each packet arrival
        Calculate the average IN queue size avg_in;
        Calculate the average queue size avg_total;

If it is an IN packet
        If min_in < avg_in < max_in
                Calculate probability $P_{in}$;
                With probability $P_{in}$, drop this packet;
        else if max_in < avg_in
                Drop this packet.
If it is an OUT packet
        If min_total < avg_total < max_total
                Calculate probability $P_{total}$;
                With probability $P_{total}$, drop this packet;
        else if max_total < avg_total
                Drop this packet.
    If it is not dropped
        If min_out < avg_total – avg_in < max_out
                Calculate probability $P_{out}$;
                With probability $P_{out}$, drop this packet;
        else if max_out < avg_total – avg_in
                Drop this packet.
```

Figure 1. RI+O algorithm.

Therefore, there is a need to provide some mechanisms to separately control OUT packets depending only on the number of OUT packets in the queue. Fig. 1 shows the modified RIO algorithm that separately control the OUT queue length. We call the modified RIO mechanism "RI+O" (RED with IN, IN+OUT and OUT). RI+O is a straightforward extension of RIO. In RI+O, whenever a packet arrives, the router calculates *avg_in* and *avg_total*, and as a fallout, obtains *avg_out* (= *avg_total* – *avg_in*) assuming the same weighted averaging constant *w_q* is used for *avg_in* and *avg_total*. In RI+O, three sets of RED parameters are used: IN, IN+OUT, and OUT parameters ($min\_out/max\_out/P_{max\_out}$). When an OUT packet arrives, *avg_total* is compared with the IN+OUT threshold and then if it is not dropped, *avg_out* is compared with the OUT threshold. Fig. 4 shows RI+O algorithm. The bold-faced part in Figure 1 is where the OUT queue length is separately controlled.

To put the performance of RI+O in perspective, we compare the throughputs achieved by different profile flows with RIO and RI+O, respectively. The simulation setting is the same as in [12], where they simulate 40 TCP flows, 20 of which are Assured Service flows and the remaining 20 are best-effort flows. An RTT of 100ms was used for all the connections. The bottleneck link bandwidth is 20Mbps and RIO parameters are 390/694/0.02 for IN, 347/694/0.05 for IN+OUT. The profile rates of Assured Service flows are 2Mbps for flows 0 ~ 3, 1Mbps for flows 4 ~ 7, 0.5Mbps for flows 8~11, 0.2Mbps for flows 12 ~ 15, and 0.1Mbps for flows 16~19. Flows 20 ~ 39 are for best-effort. The total subscription level is 76%. For each flow, a token bucket marker is attached, and the size of the token bucket is 160Kbits. In [12], there is a policer at R0 with the token bucket size of 200Kbits, but we don't use the policer. We use the ns 2.1a4 simulator [15]. All simulation results are averaged over 6 runs and each simulation runs for 100s. In our simulations, we use the TCP NewReno which always has packets to send.

The results are shown in Table 1. The fixed IN parameters are 390/694/0.02. And all the token bucket sizes are set to 320Kbits.

Table 1. RIO vs. RI+O.

Rate(Mbps)	**2.0**	**1.0**	**0.5**	**0.2**	**0.1**	**0.0 (BE)**
Best RIO	1.765	1.009	0.568	0.333	0.276	0.253
200/600/0.01	**1.633**	**0.912**	**0.557**	**0.322**	**0.287**	**0.252**
100/300/0.01	1.657	0.962	0.586	**0.346**	0.277	0.229
30/90/0.01	**1.751**	**1.013**	**0.594**	**0.342**	**0.276**	**0.198**
10/30/0.01	1.781	1.023	0.592	0.345	0.272	0.186
5/15/0.01	**1.798**	**1.025**	**0.591**	**0.339**	**0.271**	**0.179**
Fair share	2.120	1.120	0.620	0.320	0.220	0.120

When we compare the best case of RIO with the best case of RI+O, we can see that RI+O always yields improvement.

Now, we explore the effect of the number of best-effort flows on RIO and RI+O. At 40% and 50% of the total subscription level (20Mbps bottleneck link), four Assured Service flows contract the same profile rates (2Mbps and 2.5Mbps respectively), and then various number of best-effort flows are competing with them. We changed the number of best-effort flows as 0, 1, 2, 6, 11, 16, 26, 46, 96 and simulated with RIO and RI+O. RIO and RI+O have 60/180/0.02 for IN parameters, 30/90/0.05 for IN+OUT parameters, and 10/30/0.1 for OUT parameters. The total queue size is 200 packets. Fig. 2 shows the results. In 40% subscription level (Fig. 2(a)), Assured Service flows can maintain their profile rates until the number of best-effort flows is 11 with RIO and 26 with RI+O, respectively. In 50% subscription level (Fig. 2(b)), 6 with RIO and 16 with RI+O. Use of RI+O makes Assured Service flows overcome the impacts of best-effort flows.

Fig. 3 shows another simulation results. The number of best-effort flows is fixed to 20. Assured Service flows with same profile rates subscribe 20Mbps bottleneck link. The total subscription level varies from 10% to 100%. A flow's profile rate is set to 0.5Mbps, 1Mbps, and 2Mbps for each simulation set. With low profile rates (Fig. 3(a)), the performances of RIO and RI+O are similar. They can guarantee at least the profile rates until the subscription level reaches 90%. But as the profile rate of each Assured Service flow becomes larger, the performance gap between RIO and RI+O also becomes larger. With 1Mbps profile rate (Fig. 3(b)), RIO can guarantee the profile rate up to 50% subscription level, but RI+O can guarantee it up to 70% subscription level. Even with 2Mbps profile rate (Fig. 3(c)), RI+O can guarantee the profile rate until the subscription level reaches 50%, but RIO always fails to guarantee it. So, with RI+O, the profile rate guarantee can be possible until the subscription level reaches 50%. In practical situations the total subscription level may be lower than 50%, so RI+O can guarantee profile rates of Assured Service flows.

4. CONCLUSIONS

It has been observed that Assured Service cannot guarantee the contracted profile rate consistently. In this paper, we sought to find practical solutions for this problem by exploring parameter values that impact on the performance of Assured Service flows. For this reason, we only examined the newly defined but not yet deployed Differentiated Services components, such as the RIO mechanism. We proposed RI+O, a modified RIO, and evaluated its performance comparing with that of RIO. This is still an on-going work which needs more experiments and analyses. Our future work will focus on further analyzing the simulation results and RI+O.

REFERENCES

[1] K. Nichols and S. Blake, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474.

[2] D. Black, S. Blake, M. Carlson, E. Davis, Z. Wang and W Weiss, "An Architecture for Differentiated Services ", RFC 2475

[3] V. Jacobson, K. Nichols and K. Poduri, "An Expedited Forwarding PHB", RFC 2598

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

[5] K. Nichols, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet ", draft-nichols-diff-svc-arch-02.txt, Internet Draft, April 1999.

[6] D. D. Clark and W. Fang, "Explicit Allocation of Best-effort Packet Delivery Service". IEEE/ACM Transactions on Networking, Vol. 6, Number 4, August 1998, pp. 362~373.

[7] M. Goyal, P. Misra and R. Jain, "Effect of Number of Drop Precedences in Assured Forwarding", draft-goyal-dpstdy-diffserv-00 txt, Internet Draft, March 1999.

[8] K. Nichols and B. Carpenter, "Format for Diffserv Working Group Traffic Conditioner Drafts". Internet Draft, February 1999.

[9] J. Heinanen and R. Guerin, "A Three Color Marker", draft-heinanen-diffserv-tcm-01.txt, Internet Draft, February 1999.

[10] J. Heinanen and R. Guerin, "A Two Rate Three Color Marker", draft-heinanen-diffserv-trtcm-01.txt, March 1999

[11] S. Floyd and V Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1, Number 4, August 1993, pp. 397~413.

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

[13] I. Yeom and A. L. N. Reddy, "Realizing Throughput Guarantees in Diff-serv Networks", IEEE ICMCS 99.

[14] W. Feng, D. D. Kandlur, D. Saha and K G. Shin, "Understanding TCP Dynamics in an Integrated Services Internet ", Proceedings of the International Workshop on NOSSDAV, May 1997.

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

[16] References on RED (Random Early Detection) Queue Management . Available via http://www.aciri.org/floyd/red.html.

[17] W. Feng, D. D. Kandlur, D. Saha and K. G. Shin, "A Self-Configuring RED Gateway", Infocom 99.

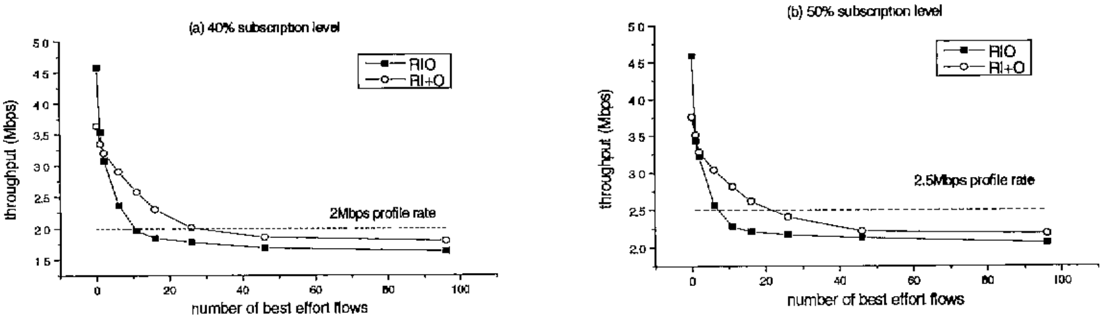

Figure 6. Throughput of AS flow vs. number of competing BE flows.