A Self-Configuring RED Gateway for Quality of Service (QoS) Networks

Chin-Hui Chien and Wanjiun Liao

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: wjliao@cc.ee.ntu.edu.tw

Abstract This paper studies self-configuring Random Early Detection (RED) gateways for Quality of Service (QoS) networks. Existing work only considers the adaptive parameter setting of RED gateways for networks with best effort flows. This paper focuses on QoS networks in which both assured and best effort flows co-exist. RED with in/out (RIO) is an extension of RED for QoS networks, which maintains two parameter sets (i.e., *in-profile* and *out-of-profile*) for different classes of traffic. We find that fixed and low *out-of-profile* thresholds limit best-effort throughput when assured flows are frequently absent or the amount of assured flows is small. We then propose an adaptive scheme called Adaptive-RIO to adjust the parameters according to network dynamics. From the simulation, Adaptive-RIO performs better than RIO in terms of higher best-effort throughput and total throughput while ensuring the committed rate of assured flows.

Keywords: RED, RIO, Active Queue Management

1. Introduction

In this paper, we study the self-configuring Random Early Detection (RED) [1] gateway for differentiated service networks. RED is a promising solution for active queue management in the Internet. Each RED gateway monitors the average queue length, denoted as avg_q. If avg_q is relatively large, the network is inferred as being congested. To maintain average queue length at a reasonable length, there are three key parameters used by RED, including min_{th}, max_{th} and max_p. When avg_q is less than min_{th}, no packets are dropped. When avg_q exceeds max_{th}, every incoming packet is discarded. When avg_q is between min_{th} and max_{th}, packets are dropped with a dropping probability p which is a linear function of avg_q up to a value of max_p, at which point it jumps to unity.

Previous studies have shown that no fixed setting of RED parameters can meet all network conditions. [2] and [3] are two major efforts on dynamically configuring the parameters of RED gateways for different network conditions. Feng *et al.* [2] has found that if the average queue length avg_q oscillates around min_{th}, the RED gateway behaves too aggressively, i.e., max_p is set to too high. On the other hand, if it oscillates around max_{th}, the RED gateway behaves too conservatively. To solve this problem, max_p is adjusted according to the dynamics of avg_q. When avg_q oscillates around min_{th}, the value of max_p is reduced to allow more packets to be accumulated in the queue. On the contrary, when avg_q oscillates around max_{th}, the value of max_p is increased so as to drop more packets.

In [3], the authors have argued that it is problematic to set fixed values to the two queue thresholds, i.e., min_{th} and max_{th}. If the thresholds are set too small, there will be many timeouts when the network is heavily loaded; on the other hand, if the thresholds are set too large, the large amount of accumulated packets in the queue will incur large delay for each flow when the network is lightly loaded. Based on the "drop rate" as the measurement metrics of network load, they keep the actual drop rate as close to a pre-specified target drop rate as possible by adjusting the two thresholds dynamically.

RIO (RED-in/out) [4] is an extension of RED for Quality of Service (QoS) networks. In the context of QoS networks, two types of traffic are considered: assured traffic and best effort traffic. The network then provides different resource assurance to different classes of traffic flows. If the flows conform to the committed rates, the service will be assured (i.e., *in-profile*); otherwise, the extra packets are marked *out-of-profile*, and treated as best effort flows. For RIO, *two RED parameter sets* are used: one is for *in-profile* packets (i.e., min_{th}^{in}, max_{th}^{in}, and max_{p}^{in}) and the other is for *out-of-profile* packets (i.e., min_{th}^{out}, max_{th}^{out}, and max_{p}^{out}). The *out-of-profile* thresholds are typically set to values lower than the *in-profile* thresholds, i.e., $min_{th}^{in} > min_{th}^{out}$ and $max_{th}^{in} > max_{th}^{out}$, and the dropping probability of *out* (i.e., *out-of-profile*) packets is larger than *in* (i.e., *in-profile*) packets, i.e., $max_{p}^{in} < max_{p}^{out}$. As such, routers can preferentially drop the packets that are marked as *out* during network congestion. An RIO gateway also monitors the values of two average queue lengths: avg_q^{in} for the queue length of *in* packets and avg_q^{total} for the total queue length including *in* and *out* packets. Based on avg_q^{in} and avg_q^{total}, an RIO gateway controls the dropping of *in* and *out* packets, respectively, in an attempt to meet QoS demands for different classes of traffic.

Existing work on self-configuring RED gateways considers an environment with only best-effort flows (e.g., [2] and [3]). They do not discuss parameter adaptation of RED/RIO gateways for QoS networks with both assured and best effort traffic. There is much work on QoS provision over the Internet. Nevertheless, best effort (or low priorities) traffic still plays an important role in the network and contributes a large portion of traffic even though certain guarantees are demanded for real-time applications. It is very likely that best effort and guaranteed flows may coexist in the network. The focus on this paper will be on the co-existence of both types of traffic. We discuss the impact of wrongly setting the parameters of RED gateways on the performance of best effort flows, and propose an adaptive scheme to adjust *out-of-profile* parameters according to network dynamics, in an attempt to avoid the wastage of bandwidth when assured traffic is frequently absent or when the amount of assured traffic is small.

The rest of the paper is organized as follows. Section 2 describes the problem of wrong settings of RIO parameters for the performance of best traffic. Section 3 presents the proposed Adaptive RIO (A-RIO) gateway. Section 4 shows the simulation results to evaluate the performance of the proposed mechanism. Finally, Section 4 concludes the paper.

2. Performance problem with wrong setting of RIO parameters for best-effort traffic

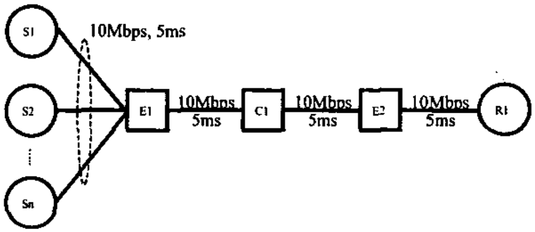

Figure 1. Network topology for simulation

In this section, we describe the problem with the wrong setting of RIO parameters for best effort traffic in DiffServ networks. To observe the effect of parameter settings on the performance of best-effort traffic, we conduct a simulation using *ns-2* [5] network simulator with an extension of the DiffServ module[1]. Fig. 1 shows the network topology used in the simulation. The squares indicate routers and the circles, the senders and the receiver. There are several senders (i.e., S1 to Sn) and one receiver (i.e., R1). Each sender has a greedy FTP connection with infinite data to send. There are two edge routers (i.e., E1 and E2) and one core router (i.e., C1). The two edge-routers use token bucket with a token bucket size of 50 ms[2] as their policy models. The core router implements the Assured Forwarding (AF) Per-Hop Behavior (PHB).

The simulation is conducted to see the effect on the throughput of best effort traffic, varying the values of the two best-effort thresholds, i.e., min_{th} and max_{th}. The simulation runs ten TCP Reno best-effort connections. Fig. 2 shows the utilization as a function of max_{th} values. The utilization (i.e., U) is defined as $U = \Sigma T_i / C$, where T_i is the throughput received by connection i and C is the bandwidth of the bottleneck link. We see that the lower the value of max_{th}, the less utilization the router can achieve. However, in RIO, best-effort thresholds (i.e., min_{th}^{out} and max_{th}^{out}) should always be set lower than assured parameters (i.e., min_{th}^{in} and max_{th}^{in}). Once the thresholds are set, the low threshold values for best-effort traffic will limit the total best-effort throughputs even though there are only best-effort flows in the network. Note that in this simulation, max_{th} is set to three times of min_{th} (i.e., max_{th}=3min_{th}). Due to space limitations, only the results of max_{th} is shown. The utilization curve of min_{th} is consistent with that in Fig. 2.

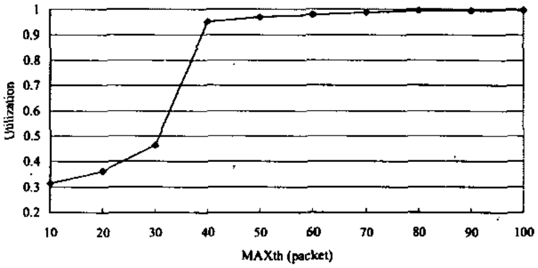

Figure 2. The throughput of best effort traffic in different settings of max_{th} values

3. An Adaptive-RIO gateway

Adaptive-RIO is an RIO with the capability of self-configuring *out-of-profile* thresholds. The main objective of Adaptive-RIO is to increase best-effort throughputs by utilizing the available buffer spaces of the core routers in DiffServ networks. In Adaptive-RIO, both max_{th} and min_{th} of *out-of-profilet* thresholds are not fixed values anymore, but can be any values in the range of a lower and an upper bound. As in RED, there is a linear relation between max_{th}^{out} and min_{th}^{out} , i.e., $max_{th}^{out} = \alpha$ min_{th}^{out} . α>1. Typically, α is set to 3 [1]. To save space, we will only show the portion corresponding to max_{th}^{out} (and will use "max" to represent max_{th}^{out}) in the rest of the paper.

Adaptive-RIO is implemented at core routers. It operates as follows. The core router performs the original operation of RIO upon receipt of a new packet, i.e., calculates the average queue length and drops this incoming packet if the dropping condition is met. The core router then estimates the arrival rate using the Timed Sliding Window (TSW) [4] and determines if the thresholds should be adjusted. The basic concept behind Adaptive-RIO is simple. The router operates as a normal RIO gateway when both types of traffic co-exist in the network or the resource is abundant. If assured traffic is

---

[1] The DiffServ module is ported by Nortel Networks.

[2] The bucket size (in terms of bytes) = token generating rate * 50 ms.

absent, best-effort traffic can use all the resource by increasing the thresholds. Once the arrival rate of best-effort packets has exceeded the maximum outgoing capacity, the network is inferred as being congested, and the thresholds of *out* parameters should be lowered down.

In our mechanism, the *out-of-profile* threshold "max" ranges between two bounds, i.e., $\max_L$ and $\max_U$, where $\max_L$ is the lower bound and $\max_U$ is the upper bound. ΔT is an increment or a decrement step at each update, defined as $\Delta T = \beta \max_U$ where $0< \beta <1$. Once the total arrival rate (A_{total}) exceeds the maximum outgoing capacity (C), the queue will be filled up rapidly. The accumulated queue will affect the service assurance of assured traffic. To solve this problem, we make use of "A_{total} exceeds C" as a signal to detect if queue build-up is imminent. When the condition holds, the *out-of-profile* thresholds are decreased.

```
Initially
  $C' \bullet C_{init}$
Every update interval
if assured flows exist
  if $A_{total} > C' + \varepsilon$
    if $A_{total}(now) > A_{total}(previous)$
        if $A_{total} > C' + \triangle C$
          $C' \leftarrow C' + \triangle C \big|_{Cinit}^{C}$
        max $\leftarrow$ max$-\triangle T \big|_{\max_L}^{\max_U}$
  else if $A_{total} < C' - \varepsilon$
    if $A_{total}(now) < A_{total}(previous)$
        if $A_{total} < C' - \triangle C$
          $C' \leftarrow C' - \triangle C \big|_{Cinit}^{C}$
        max $\leftarrow$ max$+\triangle T \big|_{\max_L}^{\max_U}$
else
  if max $< \max_U$
    max $\leftarrow$ max $+\triangle T \big|_{\max_L}^{\max_U}$
    $C' \leftarrow C' - \triangle C \big|_{Cinit}^{C}$
```

Figure 3. Adaptive parameter setting

In Adaptive-RIO, the *reference point*, denoted as C', is defined for early detection of buffer overflow. C' ranges between C_{init} and C, instead of a fixed value. If C' is fixed at C, the *out-of-profile* thresholds may not be decreased *in time*, resulting in degradation of the guarantees on assured flows. Thus, the *out-of-profile* thresholds should start decreasing when the consumed capacity has exceeded a predefined percentage of the original capacity, i.e., $C_{init.}$ $C_{init} = \delta C$, where $0 < \delta < 1$. On the other hand, if C' is fixed on C_{init}, the buffer space cannot be fully utilized; more specifically, an amount of $C - C_{init}$ bandwidth will be wasted. To solve this problem, we adjust the reference point C' within the range of C_{init} and C, with a ΔC at each update, where $\Delta C = \omega(C - C_{init})$, $0 < \omega < 1$.

Fig. 3 shows the details of self-configuring *out-of-profile* thresholds for Adaptive-RIO gateways. $\Delta C\big|_{C_{init}}^{C}$ means ΔC only ranges between C and C_{init}; similarly, $\Delta T\big|_{\max_L}^{\max_U}$ means ΔT only ranges between $\max_L$ and $\max_U$. The network is inferred as being congested if assured flows exist and A_{total} exceeds $C' + \varepsilon$ plus deviating from C' (i.e., $A_{total}^{now} > A_{total}^{previous}$). Thus, max is decreased and C' is increased, and vice versa. The condition of $C' \pm \varepsilon$ is set to avoid overly frequent updates on the threshold when the changes are relatively small and oscillate between $C' \pm \varepsilon$. Our mechanism needs to detect whether any assured flow exists in the network. If there are no assured flows during a pre-specified period, assured flows are assumed to be absent temporarily in the network. If no assured flows exist, best-effort flows can occupy more buffers to improve network utilization. Thus, the *out-of-profile* threshold is increased and C' is decreased.

4. Performance Evaluation

In this section, some simulation results are shown. The simulation is performed in the same environment as shown in Fig.1. But this time, we consider both assured and best effort flows in the network. Table 1 shows the initial setting for the simulation. We use TSW to estimate arrival rate, with a TSW window of 5 sec.

Parameter	Value
C_{init}	0.9C
$\triangle C$	0.0002C'
$\triangle T$	$0.02\max_{th}$
ε	0.025C'
UI	200ms
TSW window	5sec

Table 1. The initial setting for the simulation

In the first experiment, we compare the performance of Adaptive-RIO with traditional RIO gateways in terms of the best-effort throughput (i.e., the throughput of best effort flows), assured throughput (i.e., the throughput of assured flows), and total throughput. In this simulation, there are one assured flow with a committed rate of 4Mbps and four best-effort flows. The assured flow is idle during 0-100sec and 200-300sec in simulation time. Fig. 4 shows that Adaptive-RIO can guarantee the assured throughput and Fig. 5 shows that Adaptive-RIO have better total throughput than RIO. The solid curve is for Adaptive-RIO, and the dashed curve is for RIO. Note that Adaptive-RIO also has better best-effort throughput. We do not include the figure in this paper due to the space limitation.

The second simulation is conducted to observe the best-effort utilization improvement achieved by Adaptive-RIO when the committed rate guarantee of assured flows is not violated. Again, we simulate one assured flow with a committed rate of 4Mbps and four best-effort flows. The assured flow is modeled by an exponential ON-OFF traffic[3] with three parameters, i.e., *on_time*, *off_time* and *rate*. Both *on_time* and *off_time* are exponential random variables whose parameters are set to 10 sec; *rate* is set to 4Mbps. The simulation result is averaged over ten runs. The best-effort utilization is defined as $U_{BE} = \Sigma T_{BE} / C$, where T_{BE} is best-effort throughput and C is maximum outgoing capacity. The *on_time* is fixed at 10 sec and the *off_time* varies from 0 sec to 60 sec.

Fig. 6 shows the best-effort utilization achieved by Adaptive-RIO (i.e., the solid curve), which is always better than RIO (i.e., dashed curve). The longer the idle time of the assured flow, the better the improvement of best-effort utilization. Fig. 7 shows that both Adaptive-RIO and RIO meet the committed rate of the assured flow, which demonstrates Adaptive-RIO can achieve the committed rate as effectively as RIO while having better best-effort throughput.

4. Concluding remarks

In this paper, we have proposed a mechanism called Adaptive RIO to dynamically adjust the *out-of-profile* thresholds of RIO gateways according to network dynamics for QoS networks. We have found that fixed and low *out-of-profile* thresholds limit best-effort throughput when assured flows are frequently absent or when the amount of assured flows is small. From the simulation, we have proved that our mechanism performs better than RIO with fixed parameter settings in terms of higher best effort throughput while ensuring the committed rate of assured flows.

Acknowledgement

This work was supported in part by MOE program for Promoting Academic Excellence of Universities under grant number 89E-FA06-2-4-7, and in part by the National Science Council, Taiwan, under grant number NSC91-2213-E-002-067.

References

[1] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Trans. on Networking, pp. 397-413, 1993.

[2] W. C. Feng et al. "A Self-Configuring RED Gateway," IEEE INFOCOM'99.

[3] J. Aweya et al. "Enhancing TCP performance with a load-adaptive RED mechanism," Int. Journal of Network Management, Jan. 2001.

[4] D. D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," IEEE/ACM Trans. on Networking, Apr. 1998.

[5] http://www.isi.edu/nsnam/ns.

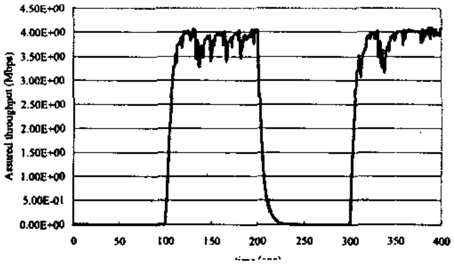

Figure 4. Assured throughput

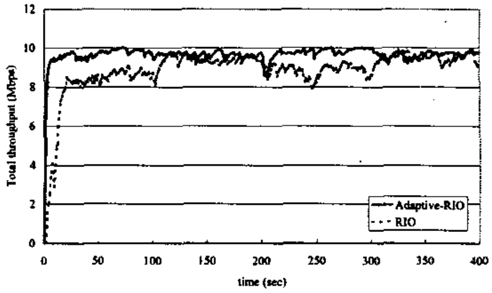

Figure 5. Total throughput

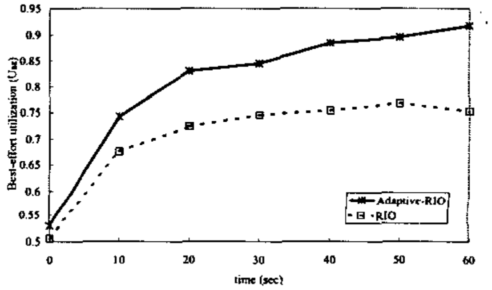

Figure 6. Best effort utilization

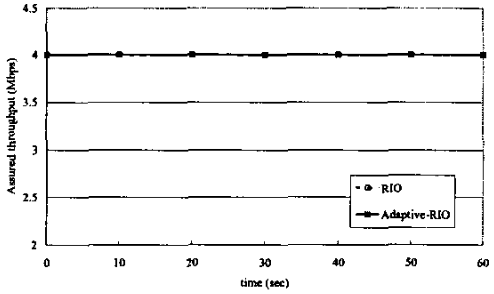

Figure 7. The committed rate of the assured flow

---

[3] Exponential ON-OFF traffic is a built-in traffic generator in ns2.