BIO Revisited*

Bahri Okuroglu, Sema Oktug
Department of Computer Engineering, Istanbul Technical University, Maslak Istanbul Turkey
okuroglu@itu.edu.tr oktug@itu.edu.tr

Abstract: The most popular active queue management algorithm, RED, is already shown to have parameter selection and adaptivity problems. The BLUE algorithm is an adaptive alternative to the RED algorithm, which behaves quite better for ECN supported networks with high number of flows active. RED based RIO algorithm is designed to offer different levels of services over a single queue and is used to provide AF service of DS architecture. Due to its ancestor's problem, RIO should be re-tuned for any change on the network and traffic patterns. Since this is not practical, BLUE-based BIO algorithm was proposed as an alternative to RIO with its adaptive nature. In this paper, BIO algorithm is re-studied for DS networks and several alternatives are considered for different parts of the algorithm. It is shown that with these modifications, BIO is a promising alternative to RIO with loss-rates around zero and minimized delay values.

1. INTRODUCTION

The benefits of the stateless architecture of the IP (Internet Protocol) have enabled the rapid growth of the Internet. With this enormous growth, network congestion, caused by the stateless architecture of IP has become more apparent. As network efficiency and reduction of the loss rate became major problems, new mechanisms are required to meet the expectations of today's applications since the architecture of the Internet is not designed to support these kinds of applications.

Jacobson's congestion control algorithms [1], which had been developed upon the first observations of Nagle's congestion predictions [2] on a real network, are still being used on the current Internet. Jacobson proposed slow start and congestion avoidance algorithms to be run by the end-node TCPs to prevent network congestion. The slow start algorithm increases the data in transit to start the self-clocking [1]. Congestion avoidance algorithms ensure that the end-nodes take the necessary action on packet loss. Congestion grows exponentially and early detection of the congestion helps to prevent it. Such a detection decreases the drop rate caused by congestion. Gateways experiencing congestion are the only nodes, which can detect it and take proper actions earliest.

Drop-tail queues, employing the traditional queue mechanism, do not require any special processing on the queue for congestion control. Packets are accepted while the queue length is less than a pre-defined limit, and all the packets are dropped after this limit. Several problems are inherent in this architecture as the queue becomes full almost all the time. Active Queue Management (AQM) mechanisms, such as RED [5] and BLUE [6], are congestion control algorithms, which are run on gateways to detect congestion earlier and to send implicit or explicit feedback to the end-nodes. Due to the advantages, the use of active queue management architectures on gateways is recommended by IETF [3].

New application types, born with the growth of the Internet, need new kind of services over Internet. IETF has been developing Differentiated Services (DiffServ, DS) architecture [4] to offer different levels of services to the applications beyond the best-effort service. Besides the drop-tail and AQM mechanisms, new queue management algorithms, like RIO [13], which are developed especially for service differentiation are used on scheduled queues to support DS.

BIO [15] was proposed as an alternative to RIO, and was expected to perform better for ECN supported networks with high number of flows active. Initially proposed BIO algorithm [15] was marking packets too aggressively, hence resulting in low link utilization. In this paper, BIO algorithm is re-studied for DS networks and several alternatives are proposed for different parts of the algorithm. Among these alternatives, the one which is considered as the most promising by the authors is studied extensively. It is shown that with these modifications, BIO overcomes RIO with loss-rates around zero and minimized delay values.

The rest of the paper is organized as follows: First, RED and BLUE AQMs are described. Next, DS architecture and related queue management algorithm, RIO, are explained. Modifications on the BIO algorithm are proposed to recover its problems. Finally the performance of a proposed BIO algorithm is compared with RIO using simulations.

2. ACTIVE QUEUE MANAGEMENT ALGORITHMS

2.1 RED

RED manages the queue based on the average queue length information. It calculates the average queue length using both the actual queue length and the average queue length, and marks packets with a probability proportional to the average queue length. w_q is the first parameter of RED which is used as weight of the average queue length in calculation of the

* This work is supported by the Istanbul Technical University Research Fund under Grant 1872.

new average queue length. By using average queue length at the marking probability calculations, instant bursts can be tolerated. Marking packets provides feedback information to source nodes on the congestion level of the gateways through the path.

RED uses two parameters, minth and maxth, on the queue length that show threshold values. Once the queue length exceeds the minth parameter, packets are marked with the randomized marking probability. maxp is the maximum packet marking probability. Last parameter, is the queue weight in the calculation of the average queue length

The behavior of the algorithm can be quite different for different set of parameters. These parameters should be tuned considering the network and traffic architecture.

The effectiveness of the RED algorithm on congestion control and reduction of the loss rate is proven by both simulations and real-world experiences [5, 7]. However, the algorithm has parameter selection problems [8]. In order to benefit from the algorithm its parameters must be arranged properly. Moreover, the marking aggressiveness of the algorithm is insensitive to the number of active flows on the gateway. When the bottleneck link is shared equally between the active flows, marking one packet to send congestion feedback decreases the total throughput by the rate of 1-1/(2N) [9]. Since only the average queue length and maxp parameter are used in calculation of RED's packet marking probability, RED cannot mark packets proportionally to the number of active flows. This may result in the loss of queue control.

2.2 BLUE

BLUE [6] is a recently developed active queue management mechanism which has a completely different marking strategy compared to RED. BLUE algorithm assumes that queue length does not directly reflect the congestion level. Hence it does not update the packet marking probability with the queue length. Instead it uses queue overflow and idle event history to update the packet marking probability (p_m). Packet loss due to the queue overflow means that the marking is not aggressive enough and p_m should be increased. Similarly, the queue idle event occurs as a result of the aggressive marking policy therefore the p_m parameter should be decreased. This mechanism effectively allows BLUE to learn the correct rate it needs to send back congestion notification.

BLUE uses three parameters: The first two parameters determine the amount by which p_m is incremented in case of the queue overflows (d_i) or is decremented when the link is idle (d_d). The last parameter is the minimum time interval between two successive updates (freeze_time). BLUE is a simple algorithm compared to RED.

Performance of the algorithms is evaluated by simulations and an experimental testbed at [6]. It is shown that BLUE successfully controls the queue and overcomes RED when ECN is used in [6].

3. DIFFERENTIATED SERVICES MECHANISMS ON ROUTERS

Differentiated Services (DS) is a new architecture [4], which is developed by IETF to support different levels of services over IP networks. It is more scalable than the previous solution, Integrated Services [10], which requires complex checks on each packet and maintenance of state information for each active flow on each gateway.

DS provides Expedited Forwarding (EF) [11] PHB (Per-Hop Behavior) which offers "low loss, low latency, low jitter, assured bandwidth" end-to-end service and Assured Forwarding (AF) [12] PHB which provides delivery of IP packets in four independently forwarded AF classes. Within each AF class, an IP packet can be assigned one of three different levels of drop precedence. AF PHB should also tolerate short-term congestion.

3.1 Red with In and Out (RIO)

To offer this kind of services a router may use different number of queues for each output interface served with a scheduler. Gateway may have an EF queue, a BE queue and four AF queues. Each AF queue should run an algorithm like RIO (RED with In and Out) [13] to be able to serve packets with different priorities in one queue.

RIO runs N (3) different parameterized RED algorithm on the queue to mark each packet respective to its priority. It begins to drop low priority packets much before the high priority packets. When the congestion increases high priority packets are dropped. It is shown that RIO can provide different levels of services to different priority packets [13].

3.2 BLUE with In and Out (BIO)

As described in Section 2.1, RED has parameter selection problems and its inadequacy to adapt to new conditions [9], and [14]. Since RIO is an algorithm based on RED, it has the same problems inherently. As an alternative, a BLUE based algorithm, BIO, was proposed by [15]. BLUE is an adaptive algorithm for a single precedence level queue, but it is not designed to offer different levels of services over a single queue as RIO.

Initial BIO algorithm was marking packets too aggressively, hence resulting in low link utilization. Below, the BIO algorithm is re-studied and several alternatives are considered for different parts of the algorithm.

BLUE does not have thresholds as parameters; it simply acts on queue idle and queue overflow events. However, to reserve different levels of resources to each precedence level, BIO algorithm should have minth and maxth parameters for each precedence level. With these two parameters, basic BIO algorithm will be as follows:

- $Q_p < minth_p$ no drop and dec pm_p
- $Q_p > maxth_p$ forced drop and inc pm_p

Three different alternatives exist for Q_p:

1. $Q_p = q$
2. $Q_p = q_p$
3. $Q_p = \text{sum}(q_i)\ [i=0..p]$ <0 is the highest precedence>

Where q_p is the number of precedence p packets in queue

Selecting the first option may result in marking higher precedence packets while low precedence packets still exist in queue. Since this is not compliant with QoS requirements, this option is ignored. Selecting the second option for minth comparison may result in the same affect, it is also ignored for minth comparison. With these restrictions we have only the last alternative for the lower threshold comparison.

Parameter selection is a problem of the algorithms since it is hard to tune them for different conditions. The following alternative may be considered for the minth comparison, to remove a parameter from the BIO algorithm:

$\text{sum}(q_i)\ [i=p+1..N] = 0$ mark precedence p packet (where N is the number of precedence levels)

This alternative means that we mark precedence-p packets only if we have no lower level packets in queue. With these, all possible changes to BLUE to form BIO is as follows:

Decrement pm_p if:

1.	$q_p = 0$	Single	[S]
2.	$\text{sum}(q_i)\ [i=0..p] = 0$	All	[A]

Increment pm_p if:

1.	$q_p > maxth_p$	Single	[S]
2.	$\text{sum}(q_i)\ [i=0..p] > maxth_p$	All	[A]

Mark packet in precedence p with pm_p if:

1.	$\text{sum}(q_i)\ [i=0..p] > maxth_p$	Threshold	[T]
2.	$\text{sum}(q_i)\ [i=p+1..N] = 0$	Idle	[I]

With multiplexing two different alternatives for the three main parts of the algorithm, we come up with eight different BIO algorithms: BIO-SST, BIO-SAT, BIO-AST, BIO-AAT, BIO-SSI, BIO-SAI, BIO-ASI, and BIO-AAI.

4. BIO COMPARISON

In this paper, BIO-AAI is compared with the RIO algorithm for ECN supported networks on the network shown in Figure 1 using ns simulation tool [16]. Performance of the RIO algorithm is also measured on similar network architecture. Basically, our configuration differs from the original RIO configuration in the number of active flows. This configuration represents many clients DS domains serviced by the ISP (Internet Service Provider) DS domain. The ISP domain is the simplest network with two edge routers and a core router. All links except link1 has 33Mb bandwidth. Since link1 is assigned 66Mb bandwidth, link2 becomes the bottleneck point. All client domains have 32 FTP/TCP flows active and they are connected to their respective domains at the other side of the ISP domain. Edge router 1, runs a token bucket marker for each client domain. Target rates and the round-trip times for each domain are chosen different for comparison. Source networks are numbered between 0 and 9. Target rate is 0.8Mb for even numbered networks, and 4.0Mb for odd numbered networks. RTT is selected as (n/2+1)*20ms where n is the network number.

The simulation is run for 500 seconds, all flows are started in the first 25 seconds and stopped in the last 25 seconds uniformly. The measurements are taken between 100-470 seconds of the simulation to exclude the learning phase of BIO.

Both RIO and BIO queues are assigned one bandwidth-delay product buffer space. With average RTT as delay, queue limit (qlim) is set to 260 1KB packets. RIO queue boundaries are selected as $maxth_{in}$=qlim, $minth_{in}$=$maxth_{out}$=qlim/2, $minth_{out}$=qlim/6. Floyd's proposal in [17] to set maxth as 3*minth is applied for out configuration. Since queue length is always below the $maxth_{out}$ ($minth_{in}$) parameter, we set $minth_{in}$ as half of the $maxth_{in}$ to provide more buffer space to the algorithms. BIO-AAI has only maxth parameter set to $maxth_{in}$ = qlim, and $maxth_{out}$=qlim/2 as in RIO.

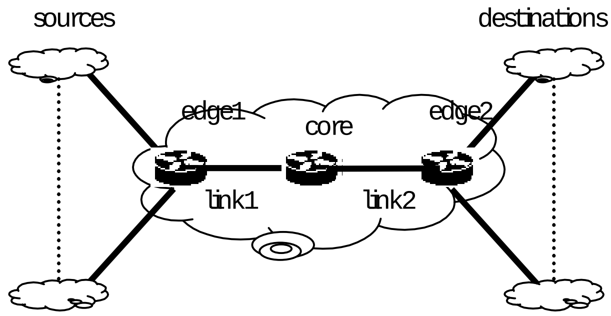

Figure 1. Simulation network configuration

Since the maxp and wq parameters of RIO are hard to tune for different network architectures and network conditions, many different alternatives are tried to select these two parameters appropriately (producing minimums and maximums for different concerns). BIO-AAI parameters are chosen as follows: IN (0.02, 0.01, 0.02, 0.01), OUT (0.001, 0.001, 0.003, 0.003) where the numbers represent (increment, decrement, increment freeze time, decrement freeze time) respectively.

Table 1. Measurement from RIO and BIO simulations

wq	maxp	Goodput on bottleneck link	Total Loss Rate (out, forced)	Avg Delay
0.0001	0.10	32.9030701248242	0.11889507	126.1374
0.0002	0.10	32.9746413061960	0.12048474	124.1927
0.0006	0.20	32.9999999983952	0.08771180	127.0812
0.0001	1.00	32.9999999983952	0.01198577	126.4260
0.0800	1.00	32.9999999983952	0.01476733	119.6525
BIO		32.9998851042515	0.00001755	60.9447

Based on the results obtained from about 300 simulations, throughput, loss rate and delay are measured for the selected set of parameters. Since the only bottleneck in the network is link2 and no other traffic class exists on the network, the number of packets in queue for each packet arrival is measured as the queuing, and overall, delay indication.

The results in Table 1 indicate that different set of RIO parameters satisfies different concerns. Based on the delay measurement we can easily conclude that for all parameters RIO cannot control the queue successfully, since queue length is always around the maximum threshold for out packets. Due to the average length calculation logic of the RIO algorithm, queue length does not exceed the $maxth_{out}$ limit much, for TCP traffic.

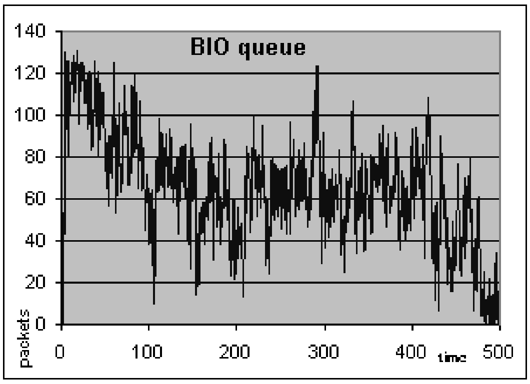

Figure 2. Queue length over time for BIO

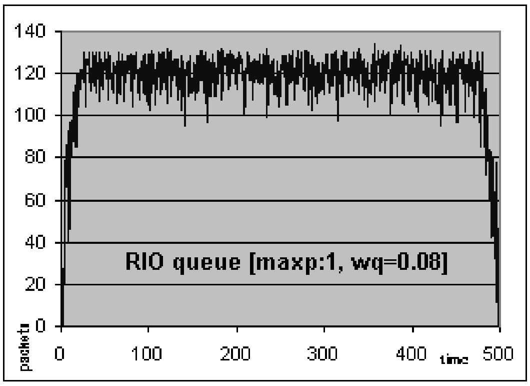

Figure 3. Queue length over time for RIO

Results also prove that BIO can control the congestion successfully, hence loss rate and the delay are quite smaller than any RIO configuration. BIO achieves this even if it is not fine-tuned using hundreds of simulations as done for RIO.

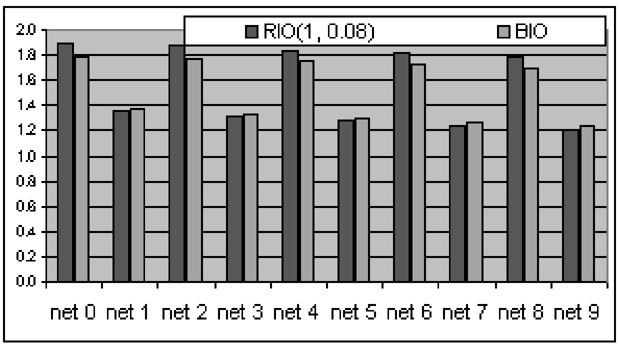

Figure 4 Measured rates for serviced customer networks

The only area, RIO is good, is the throughput measurement. Among the selected simulations, BIO is slightly worse than some of the RIO configurations. However, this small increase in throughput of RIO is achieved by acting as a drop tail queue as seen in Figure 3. BIO can still compare to RIO with respect throughput even its loss rate is quite low and acts as an AQM as shown in Figure 2.

In Figure 4, ratio of the measured throughput to the target rate for each serviced customer network is given. Here the term network is used as in context in Section 4. TCP favors flows with low target rates since it takes longer to achieve its old throughput upon a packet loss for flows with higher throughput. Therefore, odd numbered networks get more share from the access bandwidth. Since BIO's drop rate is around zero, BIO minimizes this well known behavior of TCP.

5. CONCLUSION

RIO is used to support AF service for DS networks, but its inherent parameter selection and adaptiveness problems cause to fail to reacting to different network situations. Due to this problem, RED/RIO algorithms should be re-tuned for network or traffic changes as shown in [9]. An adaptive alternative, BIO, was already proposed against RIO. However, the proposed algorithm failed to achieve high link utilization due to its marking aggressiveness.

In this paper, several modifications to the BIO algorithm are mentioned and a promising variation is compared with RIO using simulations. It is shown that the proposed BIO outperforms RIO in loss rate and delay.

Simulations indicate that BIO controls the queue successfully and achieves very low loss rates on ECN supported networks. While the link utilization is slightly lower for BIO algorithm against RIO, this is mainly because of the drop-tail like behavior of the RIO algorithm.

To eliminate the well-known parameter selection problem of RED/RIO, hundreds of simulations are run for RIO with different parameter sets and the parameters giving the best results are employed. Since BIO is an adaptive algorithm it beats RIO even without fine-tuning. It also reacts to any change in the network or traffic conditions without any interaction.

Currently, we are working on the behavior of the other BIO variations for different traffic patterns, network architectures and, lack of ECN support.

6. REFERENCES

[1] Jacobson, V., 1988, Congestion Avoidance and Control, Proceedings of SIGCOMM '88, Stanford, CA, ACM

[2] Nagle, J., 1984, Congestion Control in IP/TCP Internetworks, Request For Comments: 896

[3] Braden, B., et. al., 1998, Recommendations on Queue Management and Congestion Avoidance in the Internet, Request For Comments: 2309

[4] Blake, S., et.al., 1998, An architecture for Differentiated Services, Request for Comments: 2475
[5] Floyd, S., Jacobson, V., 1993, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking
[6] Feng, W., Kandlur, D., Saha, D., Shin, K., 2000, BLUE: A new class of active queue management, http://www.eecs.umich.edu/~wuchang/blue/
[7] Doran, S., 1998, EBONE Interface Graphs, http://adm.ebone.net/~smd/red-1.html
[8] May. M., Bolot, J., Diot, C., Lyles, B., 1999, Reasons not to deploy RED, INRIA Sophia-Antipolis, ENSIM, Sprintlabs, http://www-sop.inria.fr/rodeo/personnel/mmay/may-red.html
[9] Feng, W,. Kandlur, D., Shin, K., 1999, A Self-configuring RED Gateway, INFOCOM '99
[10] Braden, R., Clark, D., Shenker, S., 1994, Integrated Services in the Internet Architecture: an Overview, Request for Comments: 1633
[11] Jacobson, V., Nichols, K., Poduri, K., 1999, An Expedited Forwarding PHB, Request for Comments: 2598
[12] Heinanen, J., Baker, F., Weiss, W., Wroclawski, J., 1999, Assured Forwarding PHB Group, Request for Comments: 2597
[13] Clark, D., Fang, W., Explicit Allocation of Best Effort Packet Delivery Service, 1998, IEEE/ACM Transactions on Networking 6(4), 362-373
[14] Floyd, S., Gummadi, R., Shenker. S., Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management
[15] Okuroglu, B., Oktug, S., BIO: An Alternative to RIO, SPIE ITCom, Colorado, 19-24 August, 2001
[16] NS: Network Simulator, http://www.isi.edu/nsnam
[17] Floyd, S., RED: Discussions of Setting Parameters, November 1997, http://www.icir.org/floyd/REDparameters.txt