

Performance Analysis of Active Queue Management Schemes for IP Network

Jahwan Koo¹, Seongjin Ahn², and Jinwook Chung¹

¹ School of Information and Communications Engineering, Sungkyunkwan Univ.
Chunchun-dong 300, Jangan-gu, Suwon, Kyounggi-do, Korea
{jhhoo, jwchung}@songgang.skku.ac.kr

² Department of Computer Education, Sungkyunkwan Univ.
Myeongnyun-dong 3-ga 53, Jongno-gu, Seoul, Korea
sjahn@comedu.skku.ac.kr

Abstract. Active Queue Management schemes have evolved over time and continue to do so. In this paper, we present a comprehensive survey of AQM schemes for IP network. Its purpose is to identify the basic approaches that have been proposed and classify them according to the design goals and performance issues of AQM schemes. The results from several performance evaluation such as the link utilization, the average delay, and the loss rates have been provided. In particular, simulation-based comparisons of AQM schemes help to understand how they differ from in terms of fairness, global synchronization, performance guarantee, complexity and scalability.

1 Introduction

A QoS-enabled network is composed of various functions for providing different types of service to different packets such as rate controller, classifier, scheduler, and admission control. The scheduler function of them determines the order in which packets are processed at a node and/or transmitted over a link. The order in which the packets are to be processed is determined by the congestion avoidance and packet drop policy (also called *Active Queue Management*) at the node. Although there are many papers related the AQM algorithms, there were few discussed together in a single paper.

In the next two sections, we present a comprehensive survey of all possible AQM schemes. Its purpose is to identify the basic approaches that have been proposed and classify them according to the design goals and performance issues of AQM schemes. Next, we provide a description of the basic algorithms for IP network, including drop tail, random early detection (RED) [1], BLUE [2], Random Exponential Marking (REM) [3], Proportional Integral (PI) [4], and Joint Buffer management and Scheduling (JoBS) [5]. Section 4 discusses the performance evaluation of the AQM schemes via simulation. The final section offers some concluding remarks.

2 Basic Algorithms

2.1 Drop Tail

The drop tail algorithm maintains exactly simple FIFO queues. There is no methods, configuration parameter, or state variables that are specific to drop tail queues. Although simple and easy to implement, drop tail has two well-known drawbacks, the lock-out and full queue phenomena.

2.2 RED

The RED algorithm [1] was presented with the objective to minimize packet loss and queuing delay, avoid global synchronization of sources, maintain high link utilization, and remove biases against bursty sources. To achieve these goals, RED utilizes two thresholds, min_{th} and max_{th} , and an exponentially-weighted moving average (EWMA) formula to estimate the average queue length, $Q_{avg} = (1 - W_q) * Q_{avg} + W_q * Q$, where Q is the current queue length and W_q is a weight parameter, $0 \leq W_q \leq 1$. The two thresholds are used to establish three zones. If the average queue length is below the lower threshold (min_{th}), the algorithm is in the normal operation zone and all packets are accepted. On the other hand, if it is above the higher threshold (max_{th}), RED is in the congestion control region and all incoming packets are dropped. If the average queue length is between both thresholds, RED is in the congestion avoidance region and the packets are discarded with a certain probability P_a . This probability is increased by two factors. A counter is incremented every time a packet arrives at the router and is queued, and reset whenever a packet is dropped. As the counter increases, the dropping probability also increases. In addition, the dropping probability also increases as the average queue length approaches the higher threshold. In implementing this, the algorithm computes an intermediate probability P_b , whose maximal value given by P_{max} is reached when the average queue length is equal to T_{high} . For a constant average queue length, all incoming packets have the same probability to get dropped. As a result, RED drops packets in proportion to the connections' share of the bandwidth.

2.3 BLUE

The BLUE algorithm [2] uses different metrics to characterize the probability of dropping an arrival. This algorithm uses the current loss ratio and link utilization as input parameters. It maintains a single probability, P_m , which it uses to mark (or drop) packets when they are enqueued. If the queue is continually dropping packets due to queue overflow, BLUE increments P_m , thus increasing the rate at which it sends back congestion notification. Conversely, if the queue becomes empty or if the link is idle, BLUE decreases its marking probability. This effectively allows BLUE to "learn" the correct rate it needs to send back congestion notification. Besides the marking probability, BLUE uses two other parameters which control how quickly the marking probability changes over time.

The first is *freeze time*. This parameter determines the minimum time interval between two successive updates of P_m . This allows the changes in the marking probability to take effect before the value is updated again. The other parameters used, (d_1 and d_2), determine the amount by which P_m is incremented when the queue overflows or is decremented when the link is idle.

2.4 REM

The REM algorithm [3] is an AQM scheme that measures congestion not by a performance measure such as loss or delay, but by a quantity we call price. It treats the problem of marking (or dropping) arrivals as an optimization problem. The objective is to maximize a utility function subject to the constraint that the output link has a finite capacity. REM algorithm marks packets with a probability exponentially dependent on the cost of a link. The cost is directly proportional to the queue occupancy.

2.5 PI

The PI algorithm [4] uses a feedback-based model for TCP arrival rates to let the queue occupancy converge to a target value, but assumes a priori knowledge of the round-trip times and of the number of flows traversing the router. A feedback-based model consists of: 1) a *plant* which represents a combination of subsystems such as TCP sources, routers and TCP receivers, 2) the queue length at a router as a plant variable denoted by Q , 3) a desired queue length at a router (i.e., a *reference input*) denoted by Q_{ref} , 4) a *feedback signal* which is a sampled queue length used to obtain the error term, $Q_{ref} - Q$, 5) an *AQM controller* which controls the packet arrival rate to the router by generating a packet drop probability as a control signal. In [4], a simplified TCP flow dynamics model was developed. There, the open-loop transfer function (OLTF) of the plant was given by

$$P(s) = P_{TCP}(s) \cdot P_{QUEUE}(s) = \left(\frac{\frac{R_0 C^2}{2N^2}}{s + \frac{2N}{R_0^2 C}} \right) \cdot \left(\frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \right) \quad (1)$$

where N is a load factor (i.e., number of TCP connections), R_0 is the round-trip time, and C is the link capacity. Two main functions are used in the PI algorithm: one is the congestion indicator (to detect congestion) and the other is the congestion control function (to avoid and control congestion). The PI-controller has been designed based on (1) not only to improve responsiveness of the TCP flow dynamics but also to stabilize the router queue length around Q_{ref} . The latter can be achieved by means of integral (I)-control, while the former can be achieved by means of proportional (P)-control using the instantaneous queue length rather than using the EWMA queue length.

2.6 JoBS

The JoBS algorithm [5]) is capable of supporting a wide range of relative, as well as absolute, per-class guarantees for loss and delay, without assuming admission control or traffic policing.

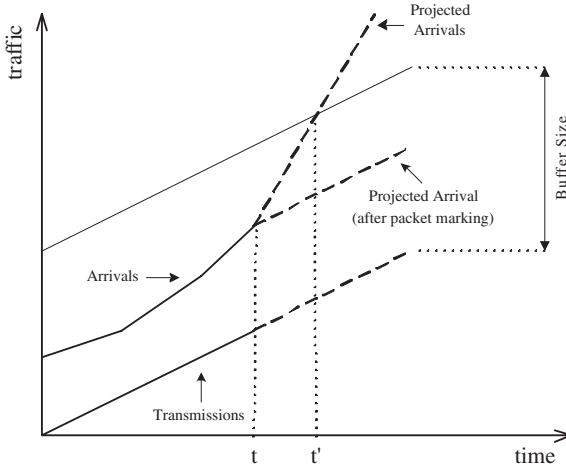


Fig. 1. The concepts of JoBS algorithm

The main idea of the JoBS algorithm is the following: At time t of a packet arrival, the router estimates the congestion window size and the round-trip time of the TCP flow. With these estimates, future traffic arrivals are projected, and impending buffer overflows are inferred, as illustrated in Figure 1. If a packet loss is projected, the algorithm reduces the congestion window size of the TCP source by marking packets with Explicit Congestion Notification (ECN). By reducing the congestion window size, the sending rate of the TCP source is reduced, and impending packet losses can be avoided. At any time t , the backlog at the router is equal to $R_{in}(t) - R_{out}(t)$. Hence, the JoBS makes an effort to meet the following requirement:

$$\forall t : R^{in}(t) - R^{out}(t) \leq B_{lim}. \quad (2)$$

where B_{lim} is the size of the router's buffer, $R^{in}(t)$ is the total amount of traffic that has entered the router until time t , and $R^{out}(t)$ is the total amount of traffic that has left the router until time t . A unique feature of JoBS is that it considers scheduling and queue management (dropping) together in a single step.

3 Classification

AQM schemes can be classified in different ways. In [6], these were classified based on two dimensions : one is when decision on discarding packets is made and the other is what information is used to make packet discard decisions. In [7], a detailed classification of the various schemes has been proposed. It was based on the networking environment (ATM or IP networks), the type of congestion management mechanism implemented (congestion avoidance or congestion control and recovery), the number of thresholds used (none, global, or per-connection), decision information (global or per-connection), and the queue behavior (static or dynamic). In this paper, a wide variety of AQM schemes with

Table 1. Classification of AQM Schemes

Criteria	Congestion		Thresholds			State-Info	
	Control	Avoidance	None	Global	Per-Conn	Global	Per-Conn
DT	•		•			•	
RED [1]		•		•		•	
BLUE [2]		•		•		•	
REM [3]		•		•		•	
PI [4]		•			•		•
JoBS [5]		•			•		•

many different characteristics for IP network are considered, and hence a more extensive classification is proposed in Table 1. We refer to a survey article [7] for an exhaustive review of all possible AQM schemes. In summary, various kinds of AQM Schemes described section 2 have mainly focused on seven issues: 1) avoid congestion, 2) reduce the packet transfer delay, keeping the queue lengths at low levels, 3) avoid the TCP global synchronization problem, 4) achieve fairness among different traffic types, 5) deliver service guarantee (guaranteed or differentiated), 6) reduce the program complexity, and 7) increase the scalability. These issues are all inter-related.

Initial proposals for AQM for IP network [1], [2], and [3] were motivated by the need to improve TCP performance, without considering service differentiation. More recent research efforts [4] and [5] enhance these initial proposals in order to provide service differentiation.

4 Performance Analysis

We present an evaluation of our surveyed AQM schemes via simulation, using the *ns-2* network simulator [8].

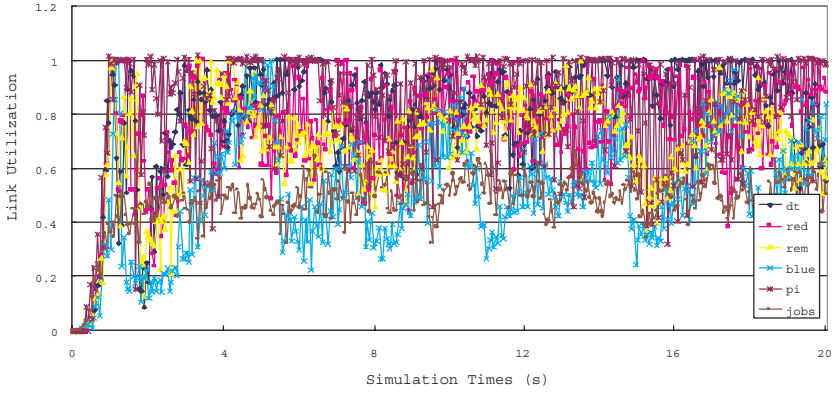
4.1 Simulation Setup

We consider a bottleneck link with 45 Mbps bandwidth, 10 ms propagation delay, and 100,000 bytes queue size. The rest of the links (edge links) are all 100 Mbps with 200 packet queues. Their propagation delay is randomly in the range 5-20 ms. Each source is connected to the corresponding sink at the other side of the network, i.e., source S_i is connected to sink R_i . There are 3 TCP source/sinks and one UDP source/sink connected to each edge node. Each TCP source is an FTP application on top of NewReno TCP. The FTP packet size is 1,000 bytes. Each UDP source is an Exponential On-Off source with peak rate 1 Mbps, average rate 200 Kbps, average Off duration 80 ms, and packet size 100 bytes. The experiment lasts for 20 seconds of simulated time, and ECN is available in the entire network. We compare the performance of six different algorithms at the core router governing the bottleneck link.

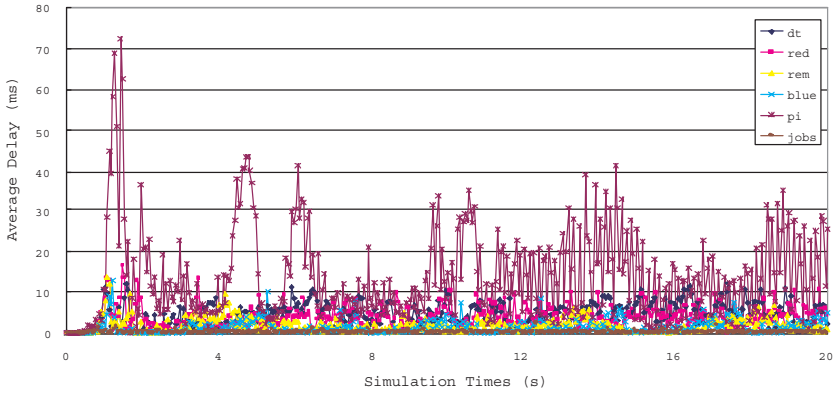
- **Drop-Tail.** We use Drop-Tail to have an estimate of the loss rates encountered without AQM. With Drop-Tail, incoming packets are discarded only when the queue is full.
- **RED.** We use RED with a minimum threshold $min_{th} = 20,000$ bytes, and a maximum threshold $max_{th} = 80,000$ bytes. The parameter max_p is set to 1, and the weight used in the computation of the average queue size is set to $W_q = 0.002$.
- **BLUE.** We use the minimum time interval $freeze\ time = 100ms$. Other parameters $d1$ and $d2$ are set to 0.02 and 0.002, respectively.
- **REM.** The parameter values of REM are set to $\phi = 1.001$, $\alpha = 0.1$, $\gamma = 0.001$, and $b^* = 20$. Their behavior according to the probability determined by the link algorithm is described in Section 2.
- **PI.** We configure the PI algorithm with approximate RTTs and a tight upper bound on the round-trip times $R+ = 180$ ms, with a sampling frequency of 160 Hz, and get $a = 1.643e - 4$ and $b = 1.628e - 4$. The target queue length Q_{ref} is set to 70,000 bytes.
- **JoBS.** We use parameter settings of $K = 10$ and $\alpha = 0.9$ which are proper values in [5].

4.2 Simulation Results

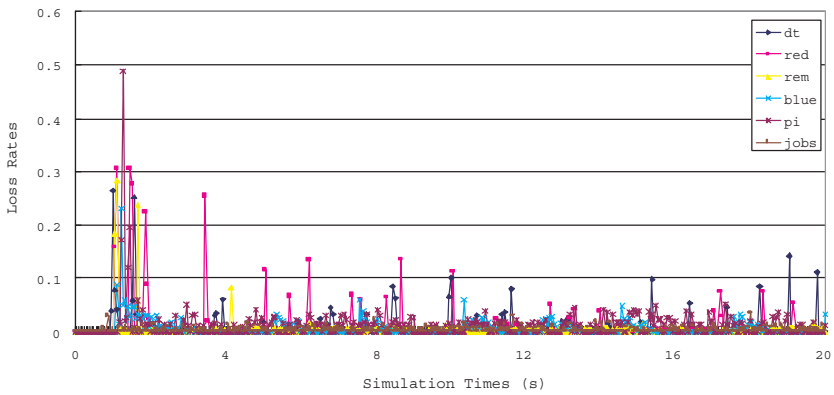
For each algorithm, we monitor the link utilization, the average delay, and the loss rates at the core router, and present our results in Figure 2 (a), (b), and (c), respectively. Not surprisingly, the Drop-Tail is almost always full, which explains the relatively high loss rates. One of the biggest problems with TCP's congestion control algorithm over drop-tail queues is that the sources reduce their transmission rates only after detecting packet loss due to queue overflow. Figure 2 (c) tells us that, without AQM, one can expect loss rates in the order of 12%. RED manages to stabilize the queue length around $max_{th} = 80,000$ bytes. However, it is unable to detect incipient congestion caused by short-term traffic load changes. As a result, AQM parameter configuration has been a main design



(a) Link Utilization



(b) Average Delay



(c) Loss Rates

Fig. 2. Simulation Results

issue. For these experiment, this is the ideal setting of max_p since it minimizes both the queuing delay and packet loss rates for RED. While min_{th} and max_{th} are chosen so that traffic is dropped with a probability of one only if the queue is full, other parameters are the default RED parameters in *ns-2*, and are therefore expected to cover a large range of operating conditions. RED is instructed to use ECN when needed. For the BLUE experiments, $d1$ is set significantly larger than $d2$. This is because link utilization can occur when congestion management is either too conservative or too aggressive, but packet loss occurs only when congestion management is too conservative. Also, Figure 2 (b) shows that a crudely configured PI algorithm drops almost as much traffic as Drop-Tail and is very sensitive to the parameter selection. Note that such a crude parameter tuning is to account for the uncertainty on estimates of the RTTs at router configuration time. Conversely, the JoBS algorithm completely avoids packet losses and lowers the queue size at all times. However, its utilization is relatively low in our simulation.

5 Conclusions

In this paper, the issues for IP networking environment have been discussed and a detailed classification of all possible AQM schemes has been provided. In addition to describing the schemes themselves, the results from several performance evaluation such as the link utilization, the average delay, and the loss rates have been presented. The evaluation has two objectives. First, we compare the performance of our surveyed AQM schemes in a range of traffic conditions. Second, we compare each algorithm's design goals such as the link utilization, the average delay, and the loss rates.

References

1. S. Floyd and V. Jacobson. "Random early detection for congestion avoidance," IEEE/ACM Transactions on Networking, 1(4):397-413, July 1993.
2. W. Feng, D. Kandlur, D. Saha, and K. Shin. "Blue: A new class of active queue management algorithms," Technical Report CSE-TR-387-99, University of Michigan, April 1999.
3. S. Athuraliya, V.H. Li, S.H. Low and Q. Yin. "REM: Active queue management," IEEE Network, Vol. 15, Issue. 3, pp 48-53, May 2001.
4. C.V. Hollot, V. Misra, D. Towsley and W. Gong. "On designing improved controllers for AQM routers supporting TCP flows," In proceedings of IEEE INFOCOM 2001, volume 3, pp 1726-1734, Anchorage, AK, April 2001.
5. J. Liebeherr and N. Christin. "Buffer management and scheduling for enhanced differentiated services," Technical Report CS-2000-24, University of Virginia, 2000.
6. R. Guerin and V. Peris. "Quality-of-service in packet networks: basic mechanisms and directions," Computer Networks, 31:169-189, 1999.
7. M. Labrador and S. Banerjee. "Packet dropping policies for ATM and IP networks," IEEE Communications Surveys, Vol. 2, No. 3, pp. 2-14, Third Quarter 1999.
8. ns-2 network simulator. <http://www.isi.edu/nsnam/ns/>.