Congestion Control Algorithms in High Speed Telecommunication Networks.

Aun Haider, Harsha Sirisena, Krzysztof Pawlikowski †
and Michael J. Ferguson‡
Department of Electrical and Electronic Engineering
Department of Computer Science †,
University of Canterbury, Christchurch New Zealand
INRS-Telecommunications, Univ. of Quebec, Canada ‡
E-mail:{{hai22,sirisehr}@elec,{†krys,‡mike}@cosc}.canterbury.ac.nz

Abstract

Modern telecommunication and computer networks, including the Internet, are being designed for fast transmission of large amounts of data, for which Congestion Control Algorithms (CCAs) are very important. Without proper CCAs, congestion collapse of such networks is a real possibility. Random tele-traffic is a heterogeneous mixture of streams of data packets that have different quality-of-service requirements. By buffering submitted packets at gateway nodes we can regulate the rates at which data packets enter the network, although this may increase the overall packet delays to an unacceptable level. Therefore it is increasingly important to develop gateway mechanisms that are able to keep throughput of a network high, while maintaining sufficiently small average queue lengths.

Several algorithms proposed recently try to provide an efficient solution to the problem. In one of these, Active Queue Management (AQM) with Explicit Congestion Notification (ECN), packets generated by different data sources are marked at the network's gateways. In other algorithms, packets are dropped to avoid and control congestion at gateways. Thus, different senders of data can be required to reduce their traffic volume if needed. Communication with original data senders is maintained by returning marked acknowledgement packets. This paper presents a brief and breadth wise survey of major CCAs designed to operate at the gateway routers of multimedia telecommunication networks.

Keywords: Gateway, Router, Congestion Control.

1 Introduction

End-to-end congestion control in telecommunication and computer networks, including the current Internet, requires some form of feedback information from the congested link to the sources of data traffic, so that they can adjust their rates of sending data according to the available bandwidth in a given network. The feedback information about congestion can be explicit or implicit.

In the case of implicit feedback, the transport layer protocol of the network tries to maintain high throughput and low delay of data packets by estimating service time, changes in throughput, changes in end-to-end delay and packet drops. The Transport Control Protocol (TCP) of the current Internet employs such an implicit feedback through timeouts and duplicate acknowledgements for lost packets. Relying only on the implicit or indirect feedback at the end nodes is not sufficient to achieve high efficiency in networks.

Therefore we need more elaborate and explicit feedback mechanisms, such as Active Queue Management (AQM), to control or manage the congestion in networks. AQM employs a single Explicit Congestion Notification (ECN) bit in a packet header to feed back the congestion in the special high speed intermediate linking computers (also called gateways), to the end users or end nodes.

These intermediate computers or gateways consist of hardware and software components that link together different types of networks seamlessly. The limited space in their buffer memory necessitates proper management of incoming traffic packets. The technique used by gateways, for transferring any type of data from one host computer to another is called *routing*. These computers are often termed as gateway routers in literature [16]. The gateway will mark packets if end host computers support ECN, otherwise it will drop the packets during congestion.

Thus whole purpose of feedback from gateway routers is to avoid congestion in the first place and to control congestion in the second place, if such episode ever occurs. The algorithms which try to avoid and control congestion at gateway routers are subject of our study in this paper, and they are collectively termed as Congestion Control Algorithms (CCAs).

The rest of paper is organised as follows. Sections 2 to 13 deal with the descriptions of CCAs, and the Section 14 deals with their classification. Finally, we present our conclusions in Section 15.

2 Drop Tail Algorithm

Drop Tail (DT) is the simplest and most commonly used algorithm in the current Internet gateways, which drops packets from the tail of the full queue buffer. Its main advantages are simplicity, suitability to heterogeneity and its decentralized nature.

However this appproach has some serious disadvantages, such as lack of fairness, no protection against the misbehaving or non responsive flows (i.e., flows which do not reduce their sending rate after receiving the congestion signals from gateway routers) and no relative Quality of Service (QoS).

The QoS is a new the idea in the traditional "best effort" Internet as given in [4], in which we have some guarantees of transmission rates, error rates and other characteristics in advance. QoS is of particular concern for the continuous transmission of high-bandwidth video and multimedia information. Transmitting this kind of content is difficult in the present Internet with DT.

Generally DT is used as a baseline case for assessing the performance of all the newly proposed gateway algorithms.

3 DECbit Algorithm

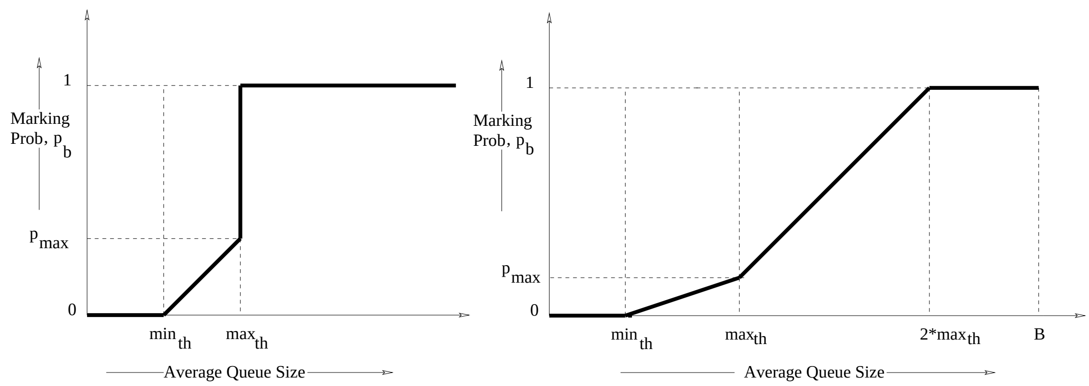
The earliest example of congestion detection at gateways is provided by the DECbit congestion avoidance scheme [21]. In this scheme the congested gateway uses a *congestion-indication* bit in packet headers to provide feedback about congestion. When the average queue length exceeds one, the gateway sets *congestion-indication* bit in the header of arriving packet.

The sources use the window based flow control mechanism. They update their windows of data packets once every two round trip times. If at least half of the packets in the last window had the *congestion-indication* bit set, then the window size is decreased exponentially, otherwise it is increased linearly.

The main disadvantages of this scheme are averaging queue size for fairly short periods of time and no difference between congestion detection and indication. The solutions of these problems were attempted by RED algorithm, discussed in Section 4.

4 Random Early Detection Algorithm

In [2], the Random Early Detection Algorithm (RED) had been proposed to be used in the implementation of AQM (explained in Section 1). For each packet arrival the average queue size, $\widehat{q_n}$, is calculated using the Exponential Weighted Moving Average (EWMA) as in [18]. The average queue size so computed is compared with the minimum threshold (min_{th}) and the maximum threshold (max_{th}) to determine the next action.



(a) (b)

Figure 1: Marking probability in case (a) is RED and (b) is gentle RED.

The basic RED algorithm can be summarized as follows: If the $\widehat{q_n} \leq min_{th}$, then no incoming packets are marked or dropped. If $min_{th} \leq \widehat{q_n} \leq max_{th}$, then the arrriving packet is marked/dropped with probability p_b, which is given by: $p_b \leftarrow max_p(\widehat{q_n} -$

$min_{th})/(max_{th} - min_{th})$. Finally, if we have $\widehat{q_n} > max_{th}$ then all incoming packets are marked/dropped.

To make the inter-packet drop uniform instead of geometric [22] suggests to use, $p_a \leftarrow p_b/(1 - count \cdot p_b)$ as the marking/dropping probability, where $count$ indicates the number of packets forwarded since last mark/drop. A graph showing the marking/dropping probability p_b versus average queue length $\widehat{q_n}$ of the RED algorithm is presented in Figure 1 (a).

The main disadvantage of RED is that its performance is very sensitive to the parameters settings. A badly configured RED will not do better than DT.

5 Variations of RED Algorithm

Some important variations of basic RED algorithm are briefly described below.[1]

5.1 Gentle RED Algorithm

In the original version of the RED algorithm described in Section (4) all of the incoming packets are marked or dropped if $\widehat{q_n} > max_{th}$. This can lead to oscillatory behavior as shown by [8]. The marking probability curve of the gentle variation of RED with maximum buffer size B is given by Fig 1 (b). This algorithm is much more robust to the undesired oscillations in queue size and to the setting of parameters as compared to original RED.

5.2 Flow RED Algorithm

The Flow RED (FRED) variation was reported in [14], in which authors argue that RED is unfair towards different types of traffic. FRED uses the per active flow accounting to impose on each flow a loss rate that is dependent upon the flow's use of the buffer. The idea behind FRED is to keep state based on the instantaneous queue occupancy of a given flow. If a flow continually occupies a large amount of the queue's buffer space, then it is detected and limited to a smaller amount of the buffer space. Thus fairness between flows is maintained. One of limitations of FRED, is the higher queue sampling frequency.

5.3 RED with Preferential Dropping Algorithm

The RED with Preferential Dropping (RED-PD) [15], is an identification-based approach which uses preferential dropping to control the high bandwidth non responsive flows. This approach has two main steps. The first step is to identify the non responsive high bandwidth flows and the second step is to reduce their bandwidth. This algorithm draws heavily from the core stateless fair queueing and the flow random early detection mechanisms, discussed in Sections 10 and 5.2 respectively. RED-PD uses packet drop history to identify and control the non responsive flows. Its main limitation is that, it cannot control a large number of non responsive flows properly.

[1]Other variations, such as Balanced RED [1], Stabilised RED [27], RED with In/Out Bit [3], Weighted RED [12], etc. are not being discussed, due to lack of space.

5.4 Adaptive RED Algorithm

The Adaptive RED (ARED) configures its parameters based on the traffic load. An on-line algorithm is given in [7]. According to it, if the average queue size q_n is in between min_{th} and max_{th}, then the max_p is mutliplicatively scaled up by factor α or scaled down by factor β depending on current status of traffic load, with $\alpha = 3$ and $\beta = 2$. Recently another version of this algorithm was reported by [9]. In this version max_P is increased additively and decreased multiplicatively, over time scales larger then a typical round trip time, to keep the average queue length within a target range, which is half way between min_{th} and max_{th}.

Main advantage of ARED is that it works automatically for setting of its parameters in response to the changing load. Its limitation is that, it is not clear that which is best and optimum policy of parameters change.

6 Proportional Integral Controller Algorithm

In order to overcome the limitations of response speed, stability, coupling between queue length and loss probability of RED, the Proportional Integral controller (PI) was designed as given in [11]. It can be implemented at a RED router as: $p := a * (q - q_{ref} - b * (q_{old} - qref) + p_{old}$; $p_{old} := p$; $q_{old} := q$; where q is the present queue length, q_{ref} is a desired queue length, and a, b are constants.

7 CHOKe Algorithm

In the CHOKe algorithm [19], whenever a new packet arrives at the congested gateway router, a packet is drawn at random from the FIFO buffer, and compared with the arriving packet. If both belong to the same flow, then both are dropped, else the randomly chosen packet is kept intact and the new incoming packet is admitted into the buffer with a probability that depends on the level of congestion. This probability is computed exactly the same as in RED. It is truly a simple and stateless algorithm which does not require any special data structure. However this algorithm is not likely to perform well when the number of flows is large compared to the buffer space,

8 BLUE Algorithms

The basic idea behind the RED queue management system is to detect the incipient congestion earlier and to feed back the congestion notification to the end hosts, allowing them to reduce their sending rates accordingly. The RED queue length gives very little information about the number of competing connections in a shared link.

BLUE and Stochastic Fair Blue Algorithms (SFB) were designed to overcome these problems, by using packet loss and link idle events for protecting TCP flows against non-responsive flows. SFB is highly scalable and enforces fairness using an extremely small amount of state information and a small amount of buffer space. It is a FIFO queueing algorithm that identifies and limits the non responsive flows based on an accounting similar to BLUE.

In [28], authors show by simulation that both BLUE and SFB perform much better than the RED.

9 Random Exponential Marking Algorithm

The Random Exponential Marking Algorithm (REM) given in [24] is a new technique for congestion control, whose main aim is to achieve a high utilization of link capacity, scalability, negligible loss and delay.

REM algorithm maintains a so called variable *price* $p_l(.)$, which is a congestion measure and is updated as follows:

$$p_l(k+1) = [p_l(k) + \gamma(\alpha_l(b - l(k) - b_l^*) + x_l(k) - c_l(k))]^+$$

where $\gamma > 0$ and $\alpha_l > 0$ are small constants and $[z]^+ = max\{z, 0\}$. Here, $b_l(k)$ is the aggregate buffer occupancy, b_l^* is a target queue length, $x_l(k)$ is the aggregate input rate to queue and $c_l(k)$ is the available bandwidth. The constant α_l trades off between utilization and queueing delay during the transient period. The constant γ controls the reponsiveness of REM to changes in network conditions. At the equilibrium point we have $\alpha_l(b_l(k) - b_l^*) + x_l(k) - c_l(k) = 0$, and $p_l(k+1) = [p_l(k) + \gamma(b_l(k+1) - (1 - \alpha_l)b_l(K) - \alpha_l b^*)]^+$ as an update rule. If a packet traverses links $l = 1, 2, ..., L$ that have prices $p_l(k)$ at a sampling instant k, then the marking probability $m_l(k)$ at queue l is $m_l(k) = 1 - \phi^{-p_l(k)}$, where ϕ is a constant. The end-to-end marking probability for a packet is $1 - \prod_{l=1}^{L}(1 - m_l(k)) = 1 - \phi^{-\sum_l p_l(k)}$, which can be approximated as $(log_e\phi)\sum_l p_l(k)$.

The main limitations of this scheme are: firstly, it gives no incentive to cooperative sources and secondly, a properly calculated and fixed value of ϕ must be known globally.

10 Fair Queueing Algorithms

The Fair Queueing Algorithms (FQ) [5], and Stochastic Fair Queueing Algorithms (SFQ) [17], are mainly used in the multimedia integrated services networks for their fairness and delay boundedness.

The frame based class of FQ is called Weighted Round Robin (WRR) [20], which is a router queue scheduling method in which queues are serviced in round robin fashion in proportion to a weight assigned for each flow or queue. Each queue is visited once per round. The Deficit Round Robin (DRR) [25] is a modified version of WRR. It takes into account the lengths of the data packets being served. These algorithms are not used in the Internet. They lie at one end of classification continuum in Section 14. Opposite to FQ lies another algorithm known as Class Based Queueing (CBQ), which is described in [23].

11 Core Stateless Fair Queueing Algorithm

The Core Stateless Fair Queueing Algorithm (CSFQ) is a highly scalable approach for enforcing the fairness between different flows without keeping any state in the core of the network [26]. It relies on the per flow accounting and marking at the edge of the network, in conjunction with the probabilistic dropping mechanism in the core network.

A key impediment to the deployment of CSFQ is that it would require an extra field in the header of every packet, and modification of all routers in the network.

12 Virtual Queue Algorithm

The Virtual Queue Algorithm (VQ) is a radical technique, reported by Gibben and Kelly [10]. In this scheme, the link maintains a virtual queue with the same arrival rate as the real queue. However, the capacity of the virtual queue is smaller then the capacity of a real queue. When the virtual queue drops a packet, then all packets already enqueued in the real queue as well as all of the new incoming packets are marked until the virtual queue becomes empty again. The fixed size FIFO virtual queue seems to be a weakness of this algorithm.

13 Adaptive Virtual Queue Algorithm

The latest algorithm as presented in [13] is the Adaptive Virtual Queue Algorithm (AVQ). If C is a capacity of the link and γ is the desired utilization at the link, then the router maintains a virtual queue whose capacity $\widetilde{C} \leq C$ and whose buffer size is the same as the buffer size of the real queue. At each packet arrrival, the virtual queue capacity is updated acccording to the differential equation $\dot{\widetilde{C}} = \alpha(\gamma C - \lambda)$, where γ is the arrival rate at the link and α is a damping factor. Both of these parameters determine the stability of the AVQ algorithm. The authors also compare their algorithm with other well known techniques such as RED, REM, PI, and VQ schemes. The adaptation of virtual queue does not correctly follow the changing traffic pattern at gateway, and it is also FIFO.

14 Comparison and Classification Of Major Gateway Algorithms

The above algorithms can be classified on the per flow basis for packets arrivals / departures or the preferential treatment in dropping as done in [15]. The former is known as a scheduling based approach and latter is known as preferential based approach. A classification of discussed CCAs is presented in Figure 2 and their main features are given in Table 1.

15 Conclusions

This paper briefly surveys gateway congestion control algorithms, noting their strengths and weaknesses. It seems that at present no single algorithm can solve all of the problems of congestion control on computer networks and the Internet. More research is

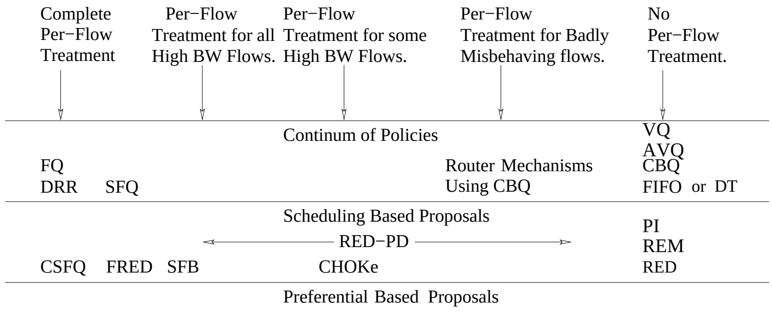


Figure 2: Classification of different gateway congestion control algorithms on the basis of per flow treatment.

needed in this direction. It is also mentioned that almost all of the surveyed papers have not employed any statistical techniques to verify their simulation results. Such techniques can be provided by [6].

References

[1] F. Anjum and L. Tassiulas. Balanced-RED: An Algorithm to Achieve Fairness in the Internet. *Proceedings of IEEE INFOCOM'99*, Mar 1999.

[2] B. Braden, D. Clark, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet. *Network Working Group, RFC2309*, Apr 1998.

[3] D. Clark and W. Fang. Explicit Allocation of Best-Effort Packet Delivery Service. *IEEE/ACM Transactions on Networking*, 6(4):362–373, Aug 1998.

[4] E. Crawley, R. Nair, et al. A Framework for QoS-based Routing in the Internet. *Network Working Group, RFC2386*, Aug 1998.

[5] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 1–12, Sep 1989. Austin,Texas.

[6] Greg Ewing, Krys Pawlikowski, and Don McNickle. *Akaroa 2 User's Manual*. University Of Canterbury, Christchurch, New Zealand, Aug 2001.

[7] W. Feng, D. Kandulur, D. Saha, and K. Shin. A Self Configuring RED Gateway. *Proceedings of IEEE INFOCOM '99*, Mar 1999.

[8] V. Firoiu and M. Borden. A Study of Active Queue Management for Congestion Control. *Proceedings of IEEE INFOCOM'2000*, March 2000.

[9] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. *Under Submission*, Aug 2001.

Table 1: Comparison of major CCAs in the Current Internet

No.	Algorithm	Main strengths	Main weaknesses
1.	DT	simplicity; no State information needed	lacks QoS; no fairness; global synchronization problems; biased for bursty traffic
2.	DECbit	simple; distributed; optimized; low over head; congestion feedback by marking packets; dynamic; provides good fairness	use simple averaging; biased against bursty traffic
3.	RED & Variants	simple; fair; QoS; EWMA; AQM; unbiased for bursty traffic	sensitive to parameters settings
4.	PI	simple; fast; robust; AQM; less queue oscillations	estimation and setting of constants
5.	CHOKe	simple; stateless and easy to implement	fairness and scalability problems
6.	BLUE & SFB	low packet loss rate and less buffer needed	not scalabe
7.	REM	low packet loss; high link utilization; scalable; and low delay	based on global parameter; lacks QoS
8.	FQ & DRR	bound on delay	expensive to implement
9.	SFQ	reduced look up cost.	complicated; incomplete fairness; more queues
10.	CBQ	better management of gateway resources	modified ethernet; no traffic control
11.	CSFQ	fairness	extra field in packet header
12.	VQ	high link utilization.	fixed & DT type of VQ
13.	AVQ	adapive to traffic changes	DT used in VQ

[10] R.J. Gibben and F.P. Kelly. Resource pricing and evolution of congestion control. *Automatica*, 1999.

[11] C. Hollot, V. Misra, D. Towsley, and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. *IEEE INFOCOM*, 2001.

[12] Cisco Systems Inc. Quality Of Service. *http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.htm*, 1998.

[13] Srisankar Kunniyur. Analysis and Design of an Adaptive Virtual Queue Algorithm for Active Queue Management. *ACM SIGCOMM*, 2001.

[14] D. Lin and R. Morris. Dynamics of Random Early Detection. *Proceedings of SIGCOMM*, Sep 1997.

[15] Ratul Mahajan and Sally Floyd. Controlling High Bandwidth Flows at the Congested Router. *ICSI Tech Report TR-01-001*, Apr 2001.

[16] A. Mankin and K. Ramakrishnan. Gateway Congestion Control Survey. *Network Working Group, RFC1254*, Aug 1991.

[17] P. E. McKenney. Stochastic Fairness Queueing. *Proc. IEEE INFOCOM*, 2:733–740, June 1990.

[18] Young P. *Recursive Estimation and Time-Series Analysis*. Sringer-Verlag, 1984.

[19] Rong Pan, Balaji Prabhakar, and Konstantinos Psounis. CHOKe, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. *IEEE INFOCOM*, Mar 2000.

[20] N. Pekergin. Stochastic Bounds on Delays of Fair Queueing Algorithms. Technical Report PRISM, UVSQ 10, Universit'e de Versailles-St-Quentin, 1998.

[21] K.K. Ramakrishnan and Raj Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks. *ACM Transactions on Computer Systems*, 8(2):158–181, May 1990.

[22] Floyd S. and Jacobson V. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug 1993.

[23] Floyd S. and Jacobson V. Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386, Aug 1995.

[24] Steven H. Low Sanjeewa Athuraliya, Victor H. Li and Qinghe Yin. REM: Active Queue Management. *IEEE Network*, 2001.

[25] M. Shreedhar and George Varghese. Efficient Fair Queueing Using Deficit Round Robin. *Proceedings of ACM SIGCOMM'95*, 25:231–243, Aug 1995.

[26] Ion Stoica, Scott Shenker, and Hui Zhang. Core Stateless Fair Queueing: Achieving Approximatley Fair Bandwidth Allocations in High Speed Networks. *ACM SIGCOMM*, Sep 1998.

[27] T.V. Lakshman Teunis J. Ott and Larry H. Wong. SRED: Stabilized RED. *Proceedings of IEEE INFOCOM '99*, Mar 1999.

[28] Debanjan Saha Wu-chang Feng, Dilip D. Kandlur and Kang G. Shin. BLUE: A New Class of Active Queue Management Algorithms. *Technical Report CSE-TR-387-99, University of Michigan*, April 1999.