International Journal of Soft Computing 11 (3): 160-164, 2016
ISSN: 1816-9503

An Initial Occupancy Based Router Policy Ensuring Better QOS for Real Time Data Transfers

Jyothish K. John and R.V. Siva Balan
Department of Computer Science, Noorul Islam University, Tamil Nadu, India

Abstract: Forwarding and delivery in internet is not guaranteed and it is a best-effort service. Real-time data transfers which have deadlines on various Quality of Service (QOS) parameters may suffer from severe problems of throughput and delay while getting transferred through internet. This can be overcome either by a scheduling mechanism or by a dropping policy which help the real-time data transfers achieve their QOS requirements without keeping a bias against non-real-time data flows. There are many active queue management policies and dedicated queuing mechanisms trying to bring fairness for flows which may also promote real-time data transfers indirectly but may not be sufficient to satisfy their deadlines of throughput and delay. Active queue management policies like RED based schemes try to bring general fairness by allocating bandwidth fairly to all the flows but even such schemes cannot provide a fair treatment to real-time flows which send small sized packets at regular intervals. The study proposes adropping policy which differentiate real-time data transfers and promote them to correct extend without denyingdeserving quality for non-real-time traffic. The experimental results prove that the new dropping policy, fair RED can promote real-time data transfers approximately to a correct extend without causing a bias against non-real-time transfers.

Key words: Real-time data transfers, quality of service, RED, initial occupancy, mean loss rate

INTRODUCTION

Data transfers in the internet can be classified into responsive and non-responsive. Responsive flows reduce their sending rate when congestion is sensed in the network. This happens when the network drops a few packets belonging to a flow. Hosts running their application using TCP have End-to-End congestion control mechanism. Non-Responsive applications on the other side do not have a congestion detection mechanism since they run over UDP protocol. Therefore, the growing rate of non-responsive flows in the internet reduces the congestion-adaptability of the internet. Applications concentrating on performance rather than reliability fall on the latter category. Real-Time data transfers which have high delivery deadlines either on throughput or delay can be justified for their non-responsiveness because most of them uses small packet sizes (Dimitriou and Tsaoussidis, 2008, 2010) but long non-real-time flows being non-responsive reduces the stability of internet. Real-time data transfers fall under the category of short non responsive flows which sends small packets in regular intervals. They also have deadlines on various QOS parameters. The general router policies designed for fair bandwidth sharing in internet majorly concentrates in restricting the non-responsive category. But, this will become a severe blow for the real-time data transfers which contribute only small rates. There should be a policy by which the scheduling or dropping module should identify real-time flows from all other traffic and promote them. The extent of favour may be proportional to the data rate they contribute to the internet. Those flows contributing less data rates should be able to achieve their QOS deadlines. However, the active queue management schemes like RED (Floyd and Jacobson, 1993) does not guarantee fair ness particularly for the real-time flows since they does not differentiate the flows. The policies like SDP (Lin and Morris, 1997) which promote the real-time flows are able to differentiate such flows correctly and promote them to a fair level without resulting on a bias against non-real-time flows as well. The dropping policy which is proposed in this study is inspired from RED; differentiate the flows like SDP but considering the initial occupancy of RED queue to frame admission control.

Related work: Integrated service and differentiated service are application level architectures for ensuring QOS for any type of traffic. The disadvantages of QOS architectures are additional protocol overhead, poor scalability and high infrastructural needs. Random early detection gateway for congestion avoidance Floyd and

Corresponding Author: Jyothish K. John, Department of Computer Science, Noorul Islam University, Tamil Nadu, India

Jacobson (1993) was proposed and is now recommended for deployment in the internet. RED allows a router to drop packets before its queue becomes saturated. Therefore congestion responsive flows will back-off early resulting in shorter average queue lengths which is good for interactive applications. Another advantage is packet drops will not occur in bursts. RED achieves this by dropping packets with a certain probability depending on the average queue length. RED does not categorize the flows into real-time and non-real-time, so RED policy does not explicitly promote real-time data transfers. The variants of RED, RIO (RED IN/OUT) (Clark and Fang, 1998) and Weighted RED (WRED) are other examples of active queue management schemes. The dropping policy is based on priority levels which are indicated within packets and in a congested scenario, lower priority packets are dropped. RIO and WRED need packets to be marked with priority levels.

Lin and Morris (1997) proposes a separate queue management scheme called FRED (Flow Random early Detection, modified version of RED) suitable for fragile and adaptive flows. FRED maintains better fairness by admitting the equal share of flows. Hence, it controls misbehaving flows from consuming more bandwidth. But, the policy is also not focussed towards promoting real-time data transfers. Floyd and Fall (1993) propose a router policy to restrict the unresponsive flows that does not reduce the sending rate even when packets are dropped. Such flows are termed as non-TCP friendly flows and such flows are identified from the drop history of RED. Mahajan *et al.* (2001) propose a mechanism to control the dropping of responsive flows. The technique have two parts: Identifying the responsive flows and Mechanism to prevent the dropping for responsive flows. Identifying the flows is performed by random sampling from the RED drop history. Feng proposes another Active Queue Management Algorithm, BLUE. A problem with queue management (like RED) algorithms is highlighted in the paper as they use queue lengths as the indicator of the severity of congestion. Instead BLUE uses packet loss and link idle events to identify congestion.

Pan *et al.* (2000) propose a stateless active queue management scheme-CHOKe which deals with an alternate queue management scheme inspired from RED. The queue for incoming packet is FIFO which is having a RED like minimum and maximum thresholds. But RED tries to maintain fairness only after the queue length become greater than minimum threshold and by this time misbehaving flows may have occupied the queue. Therefore fairness in RED is only granted after queue length is greater than minimum threshold. CHOKe scheme only differs in policy between minimum threshold and maximum threshold when compared to RED. CHOKe algorithm try to bring better fairness. It assumes that the statistics of misbehaving flows are present in the occupancy before attaining minimum threshold.

Mamatas and Tsaoussidis (2009) propose a new service differentiation policy for real-time traffic. They suggest a new scheduling policy for non-congestive packets. Non-congestive packets are packets that do not cause congestion in network (real-time packets). They are analysed by their small packet sizes. The router captures the size of the packet and learns whether it is real time traffic, if so it is serviced faster. The limit of favour is controlled by a configurable threshold. They experiments the impact of non-congestive queuing, NCQ (Mamatas and Tsaoussidis, 2007) for sensor traffic. The authors propose LIBS (2007) (Less Impact Better Service) philosophy. The packets can be classified into congestive and non-congestive based on packet sizes. Packet sizes serves as an easier identification for real-time data transfers in internet.

Policies considering initial occupancy: Size oriented Dropping Policies (SDP) use RED based queuing scheme. But RED does not take in account the initial occupancy of the queue. It does not analyse the type of packets occupying the area before minimum threshold. The admission control mechanism of RED is based only on queue length. Knowing the type of packets occupied the initial portion of the queue helps to frame a fair admission control policy when queue length falls between minimum and maximum thresholds. However SDP proposes an easy and successful way differentiate real time packets from others based upon its size. The proposed policy modifies the RED based scheme with its admission control framed fairly according to the initial occupancy of the queue. In the admission control phase of RED, real-time packets are differentiated based upon their sizes like SDP and prevented from getting dropped to a fair extend. This will help the real time flows which are small in packet size get the correct share of bandwidth (Fig. 1).

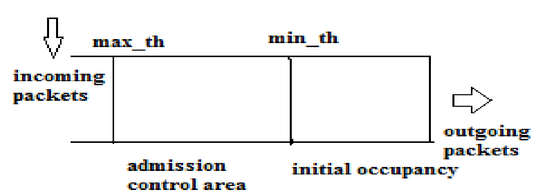

Fig. 1: Admission control area and initial occupancy of RED queue

MATERIALS AND METHODS

Initial occupancy based router queue management: We propose a new algorithm fair RED that consider initial occupancy of queue and frame admission control based on this information. The q_len is the average length of RED queue based on which admission control is determined. The min_th and max_th are the minimum and maximum thresholds for the RED queue. pkt_size is the size of the packet in bytes. size_th threshold is set for distinguishing real time and non-real time packets. When q_len reaches beyond min_th, real_share and fair_share for the flow corresponding to the incoming packet is computed. The real_share has to be calculated from the initial occupancy as the ratio number of bytes occupied by real-time packets to the total initial occupancy. If admission control area of RED is large, sampling of initial occupancy can be employed.

The ideal_share is the count of real-time flows in the initial occupancy/total no of flows in the initial occupancy. The algorithm computes the share of real time packets in the initial occupancy in bytes as real_share and tries to balance it with the ideal share in the admission control phase of RED. In all the computations real-time packets are identified by their packet sizes. The fairRED algorithm is given bellow. The extract_length(queue) is a function for finding the queue length. RED uses a moving average to calculate the queue length. The enqueue(packet) is called when a packet has to be queued. prob_drop(packet) is a function which is called when probabilistic dropping has to be performed. The probability is calculated in the same way as in RED. The algorithm only differ in the admission control policy of RED. The size_th is set in bytes to differentiate real-time and non-real time packets. Inital occupancy of the queue is found by the function initial_occupancy_check() which is also explained in Alogrithm A.

Alogrithm A; inital occupancy of the queue:

```
When packet arrives,
q_len= extract_length(queue)                    //Analyse queue length
        if ( q_len< min_ th)
enqueue(packet)//(like RED)
if ( q_len> max _th)
prob_drop(packet)//Drop the packet (like
    RED)
if  (min_th<q_len<max_th)
{
//Check the size of packet
to analyze whether it is real- time
or non-real time
pkt_size=pack_size(packet);
if (non -real packet)
        //Probabilistic_drop(like RED)
prob_drop(packet)
if (real-time packet)
{
    //Check the initial occupancy of the queue and find real_share and
ideal_ share.
initial_occupancy_check( );

if (real_share<fair_share)
{
enqueue( packet)  //do not drop
//update real share
    size_real=size_real+pkt_size;
    else
    prob_drop(packet)//like RED
    }
}
}
```

The initial occupancy of the queue is calculated using the function initial_occupancy_check()when the qlen is between min_th and max_th (Algorithm B).

Alogrithm B; randomly select n packets from initial occupancy region of queue(n is the sample size):

```
for(n=0; n<sample_size;n++)
 {
    pkt_size=packet_size(packet)
    if(pkt_size<size_th)
        {
    count_real++
        size_real=size_real+pkt_size
        }
else
    Keep a data structure to store the count of packets corresponding to a
non-real flow.
fair_share = count_real / count_real +
        count_non_real
real_share= size_real / size_real +
        size_non_real
}
```

RESULTS AND DISCUSSION

A dumbbell topology has been created using network simulator. Total 100 senders and 100 receivers are modeled and threeintermediate routers implemented with the dropping policies. There were 90 nodes each sending FTP data (non-real-time data transfer) of 1000 bytes (rate: 2000 b/sec), 5 VoIP nodes send data of 120 bytes (small packets at regular intervals, rate: 1200 b/sec) and 5 nodes are considered as sensor node sending 40 bytes (tiny packets rate: 130 B/sec). The data rate is for medium data flow scenario. Data rates of VoIP and Sensors are increased and decreased to model other 2 different scenarios (high and low) that is data rate for VoIP is increased and decreased while the others is set to medium to model high data rate and low data rate scenario for VoIP. Similarly data rate is varied to model high and low data rate scenario for sensor traffic. Simulation is carried out for 30 sec in 3 rounds with dropping policies DropTail, RED, fair RED (proposed policy). The trace is analyzed in each case and throughput, mean loss rate of smallpackets (VoIP and Sensor flow) are monitored. Here instead of sampling the initial occupancy to frame the correct admission control policy, every packet is analyzed to find weather it is real-time or non-real-time. In the experiments, packet size for determining real time packet (size_th) is set as 130 bytes. Loss rate is decreased for small flows when the routers implement proposed dropping policy. For RED and fairRED, the parameters are set as (minimum Threshold) min_th = 2500 B and (maximum Threshold) max_th = 7000 B, respectively. The matrices used to analyze the results of simulation are given.

Throughput: Throughput is used to measure the overall performance of the network in terms of effective bandwidth utilization. Throughput = Data received/ Transmission Timewhere Datais the number of bytes delivered from a sender to the corresponding receiver during the connection (Transmission Time).

Mean throughput: Mean throughput is the average of the Throughput values of the individual flows.

Loss rate: Loss rate is the number of lost packets divided by the total number of transmitted packets. Packets can be lost due to buffer overflows or proactive dropping.

$$\text{Loss rate} = \frac{\text{No. of lost packets}}{\text{No. of transmitted packets}}$$

Mean loss rate: Mean loss rate is the average of all individual loss rates in the network. Mean Loss rate = / n. DropTail form a bias against certain flows by admitting the flows which send burst of packets and which send large packet sizes. The experiment results prove that mean loss rate, percentage of packets dropped for real-time packets belonging to both VoIP and Sensor traffic are reduced with fairRED when compared with RED. This is because fairRED frames its admission control policy based upon the initial occupancy of the queue. There after the correct rate of real-time traffic is admitted and prevented from dropping. Therefore fairRED strives to admit equal rate of bytes from various flows. However the policy does not form a bias against the FTP traffic. The throughput of the small flows are also increased with fairRED when compared with DropTail and RED (Fig. 2-6).

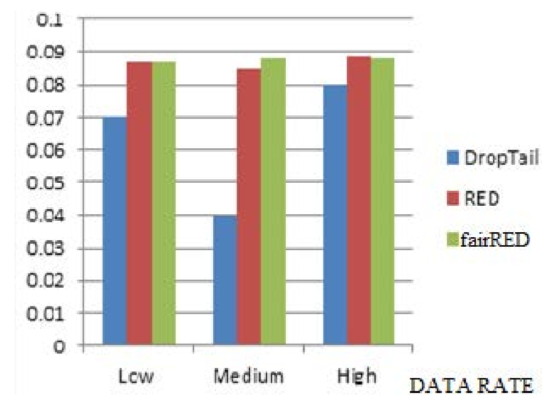

Fig. 2: Mean loss rate for FTP traffic

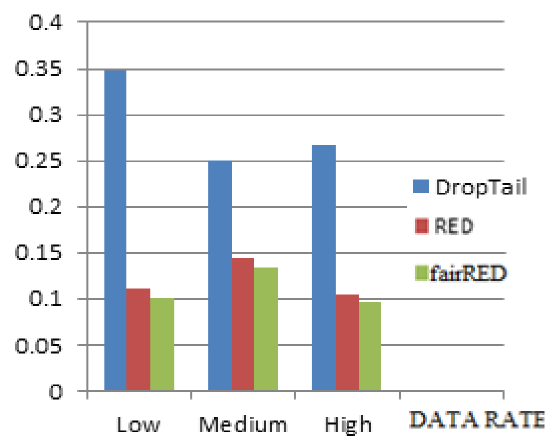

Fig. 3: Mean loss rate for VoIP traffic

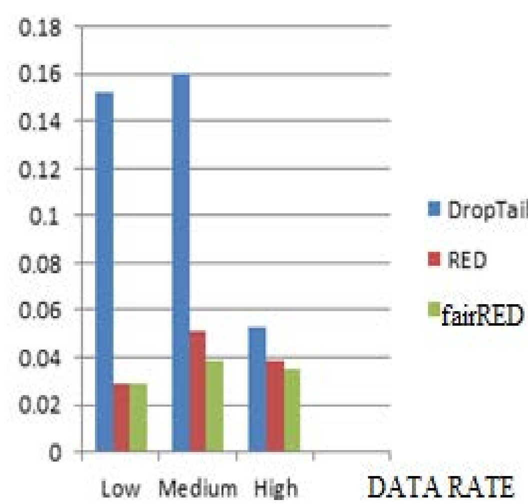

Fig. 4: Mean loss rate for sensor traffic

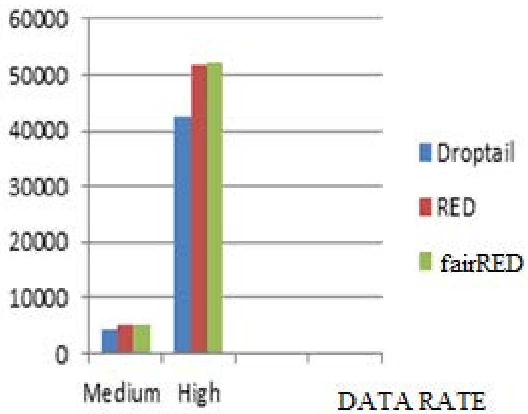

Fig. 5: Throughput for VoIP traffic

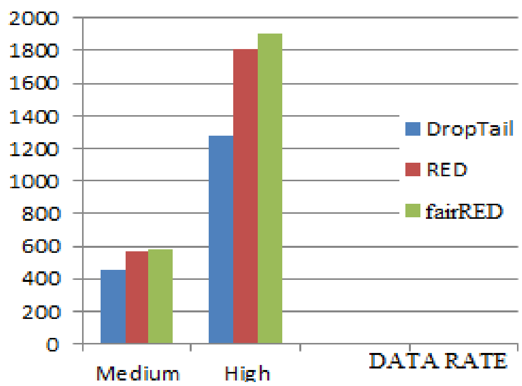

Fig. 6: Throughput for sensor traffic

CONCLUSION

The dropping policy proposed in the study is RED based but modified to provide quality of service even to flows that send packets in small rates at regular intervals. An easy traffic differentiation within the router is achieved by analyzing the packet sizes as in SDP. Simulation results also prove that fairRED employees a fair admission control mechanism when compared to DropTailand RED. Loss rate is decreased for small flows and their throughput is increased when the routers implement fairRED dropping policy. The policy considers the initial occupancy for deriving actual share and tunes admission control based on it. Throughout the experiments for RED and fairRED, the parameters, min_Th and max_Th and max_Th was maintained as 2500 B and 7000B.This is the criteria deciding the admission control area of RED and a fairRED. By expanding area of admission control, the experiment has to be repeated to find the impact of these parameters on the policy. The initial occupancy is analyzed in the experiments by checking all the packets which practically consumes processing time especially when large admission control area and more number of flows are involved. Here sampling of the packets in the initial occupancy can be performed which has to be experimented further.

REFERENCES

Clark, D.D. and W. Fang, 1998. Explicit allocation of best-effort packet delivery service. Networking, IEEE/ACM Trans., 6: 362-373.

Dimitriou, S. and V. Tsaoussidis, 2008. A new service differentiation scheme: Size based treatment. Proceedings of the International Conference on Telecommunications ICT 2008, June 16-19, 2008, IEEE, Saint Petersburg, Russia, ISBN: 978-1-4244-2035-3, pp: 1-5.

Dimitriou, S. and V. Tsaoussidis, 2010. Promoting effective service differentiation with size-oriented queue management. Comput. Networks, 54: 3360-3372.

Floyd, S. and V. Jacobson, 1993. Random early detection gateways for congestion avoidance. IEEE/ACM Transa. Networking, 1: 397-413.

Lin, D. and R. Morris, 1997. Dynamics of random early detection. ACM. SIGCOMM. Comput. Commun. Rev., 27: 127-137.

Mahajan, R., S. Floyd and D. Wetherall, 2001. Controlling high-bandwidth flows at the congested router. Proceedings of the Ninth International Conference on Network Protocols, November 11-14, 2001, IEEE, Seattle, Washington, USA., ISBN: 0-7695-1429-4, pp: 192-201.

Mamatas, L. and V. Tsaoussidis, 2009. Differentiating services with Noncongestive Queuing (NCQ). IEEE. Trans. Comput., 58: 591-604.

Pan, R., B. Prabhakar and K. Psounis, 2000. CHOKe-a stateless active queue management scheme for approximating fair bandwidth allocation. Proceedings of the 19th Annual Joint Conference of the Computer and Communications Societies, March 26-30, 2000, Tel Aviv, Israel.