IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 1 (Jul. - Aug. 2013), PP 107-113
www.iosrjournals.org

Assessing Buffering with Scheduling Schemes in a QoS Internet Router

Onadokun I.O[1], Oladeji F.A.[2]
[1] *(Centre for Information Technology and Management, Yaba College of Technology, Lagos, Nigeria.*
[2] *(Department of Computer Sciences, University of Lagos, Lagos. Nigeria.*

Abstract: *A key requirement for service differentiation as required in Internet of the future and QoS to work effectively is the extension of traffic management routines of the current TCP/IP protocol. Two among of such traffic functions are the introduction of differential packet buffering and multi-queue scheduling algorithms at the routers. Different propositions have been made to extend the Internet best-effort service model but they are yet to be incorporated into the protocol as they are still subjected to experimentations. This paper examines priority, round robin and weighted round robin scheduling algorithms that could be used in a multi-queue platform and simulate them with RIO-C penalty enforcement buffering scheme to determine which one could improve the network performance in terms of loss rate.*

From the simulation analyses, priority discipline ranked first with scheduler drop rate of 29.46% and a RED loss rate of 14.95%. Round Robin ranked second with 29.53% and 10.50% scheduler and RED loss rates respectively. Weighted round robin ranked third with 30.28% and 3.04% scheduler and RED loss rates respectively. With these results, it was observed that a network that desires feasible quality of service implementation could adopt admission control based on RIO-C with priority scheduling algorithm.

Keywords*: Loss rate, Priority scheduling, QoS, RIO-C, TCP/IP*

I. Introduction

New applications such as Multimedia applications, voice over IP (VoIP), e-business and other new services that are being routed on the Internet led to the increase of traffics in the network. Apart from the traffic load, these applications required differential treatment for their packets rather than one-size-fits-all of the best-effort TCP/IP service model. For more than a decade now, research efforts have been on the extension of the TCP/IP protocol to fully support and grant the quality required by newly evolving applications. One outcome of such extensions is the proposition of two frameworks for the Internet which are termed Integrated Services (IntServ) [1] and the Differentiated Services (DiffServ) [2] paradigms. The IntServ demands that each network flow is individually provided for at the routers along the path to their destinations which rendered it non-scalable. The DiffServ platform which received a general acceptance among researchers aims at grouping flows together and offers the desired service type to the group.

A key requirement for DiffServ paradigm to work effectively is the extension of network traffic management routines of the current TCP/IP protocol. Two among such traffic functions are the introduction of differential packet buffering and multi-queue scheduling algorithms at the routers. In [3], a scheduler is defined as “An algorithm that influences three orthogonal traffic management functions: the buffering of the packet streams, the ordering of the packets for transmission and the dropping of the packets to avoid network collapse”. These functions are made complex at a Quality-of-Service (QoS) based router if many queues with different service qualities are to be managed.

This paper presents the scenarios where different scheduling schemes were employed to service a buffer which supports differential buffering at a core router. The schedulers considered are the Priority (PRI), Round Robin (RR) and Weighted Round Robin (WRR) while Random Early Detection In-profile-Out-of-profile Coupled (RIO-C) buffering scheme is used as the active traffic admission manager.

II. Related Works

Congestion by definition is a situation whereby the inflow of traffic exceeds the outflow. According to [4], congestion is described as a situation where the total demand for a resource exceeds the available resource i.e.

$$\sum_{i=1}^{n} D_i > R \qquad (1)$$

Where D_i is the traffic demand by source i, R is the capacity of the network link and n is the number of active sources. When congestion sets in, queue is built. This is why the Internet protocol, TCP/IP, employed best-effort service model in which a router carries as much as it could from the buffered packets. Queue management is one of the critical traffic algorithms that can be used to achieve service differentiation. Two algorithms manage a queuing system: the buffering and the scheduling routines. These schemes are considered in the following subsections.

2.1 Differential Buffering

A buffer is the memory space in a network node used for temporary storage of packets before it is forwarded on a link. Buffer management scheme is the algorithm that determines which items are allowed into the buffer especially when the size is finite e.g. an M/M/1/K system [5]. For instance, when a switching device like a router is busy, the spaces for keeping incoming packets are buffers. The simplest buffering scheme is the drop-tail in which all other incoming packets are dropped when the buffer is full. While deliberating on how to implement an active queue management (AQM) scheme to support differentiated services in [6], it was suggested that buffers should not be allowed to get filled up before admission control is applied. To top the story, implementing QoS in a multi-queue platform calls for differential buffering of traffic packets. In such platform, there suppose to be an agreement with the sources of traffic and the network. While the traffic source specifies its desired quality of service from the network, (service level), the network determines the amount of traffic the source should inject into the network (service profile). This is termed Service Level Agreement (SLA) between the traffic source and the protocol driving the network [2].

In order to implement DiffServ, packets are expected to be buffered into different service treatment queues at the routers along the path to the destination. Researchers have proposed various buffering mechanisms to achieve this requirement but yet to be adopted in the Internet. Among such schemes are the use of RED In-profile Out-of- profile (RIO) [7, 8], weighted RED [9] and drop tail algorithms. Random Early Detection (RED), according to [7], attempts to avoid global synchronisation whereby, useful packets are dropped when buffer is full. During incipient congestion, it sets two important thresholds, the maximum (max_{th}) and the minimum (min_{th}) thresholds respectively. The RED algorithm proceeds in this order when a packet arrives:

- Calculate the average queue size.
- Queuing up arriving packets only if the average queue size is below the min_{th} threshold.
- Depending on set packet drop probability (max_p), the packet is either dropped or enqueued if the average queue size is between the min_{th} and max_{th} threshold.
- The packet is automatically dropped if the average queue size is greater than the max_{th} threshold.

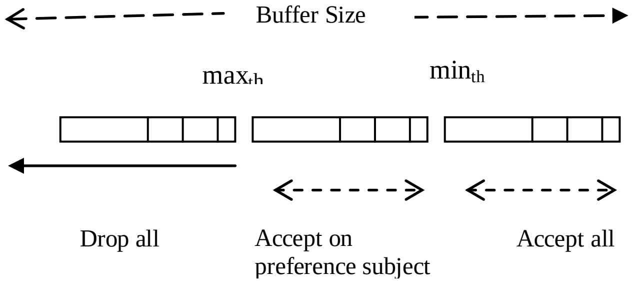

Fig. 1 RED logic for buffer management

As shown Fig. 1, all traffics that arrive when the minimum threshold is not reached is globally accepted while in between the maximum and minimum thresholds, traffic acceptance into the buffer is subjected to a certain probability test. This algorithm was incorporated to the TCP/IP in 1993 and the detail is in [7].

An attempt to accommodate the RED algorithm for DiffServ platform has led to the extension of the algorithm to RED In-profile Out-of-profile simply called RIO scheme in [8]. In RIO, packets marked by the edge routine with the same code points are meant to receive the same treatment from the network but a marginal penalty exist among these packets when its source exceeds the agreed profile. Those packets after the agreed committed information rate are considered out of profile even though they have the same code points with packets within the committed rate. Such out-of–profile packets are buffered differently. These internal queues for the packets of the same code points (physical queue) are called virtual queues. In ns 2, a physical queue can consist of more than two virtual queues called precedence levels [8]. Literature also revealed two approaches to the use of RIO buffering scheme called RIO-Coupled (RIO-C) and RIO-Decoupled (RIO-D) [10, 11].

In RIO-C, the probability of dropping an out-of-profile packet in a physical queue is based on the average queue lengths of all virtual queues while the probability of dropping an in-profile packet is based solely on the weighted average length of its virtual queue. RIO-C derives its name from the coupled relationship of the average queue calculation. In the case of RIO-D, the probability of dropping an out-profile packet in a physical queue is based on the size of its virtual queue. RIO-D calculates the average queue for packets of each virtual queue independently [10]. For example, the average queue length for green, yellow, and red packets will be calculated using the number of green, yellow and red packets in the respective queues. The strictness or wildness of RED depends on the parameter settings for each queue.

Other differential services buffering scheme is the use of Weighted RED (WRED). WRED is an extension to RED with congestion avoidance capabilities where different queues may have different buffer occupation thresholds before random dropping starts and different dropping probabilities based on a single queue length [12]. The scheduling policy used to pick packets from a queue influences the strength of the buffer management routine.

2.2 Differential Scheduling Schemes

Packet scheduling is the process of choosing which packets stored in a buffer should be transmitted over a specified link. The choice must be taken in a very small period of time in relation to the packet transmission time. Given higher priority to certain queue is at the expense of other queues which may resort to starvation and longer waiting time. Among available or suggested scheduling disciplines in literature, this paper considered the use of priority, round robin and weighted round robin (WRR).

In priority algorithm, priority level is assigned to each service queue. When the scheduler is ready to select the next queue to be serviced among the backlogged queues, the queue with the assigned highest priority is selected. The disadvantage of this discipline is that other service queues may be starved if the higher priority queues keep having backlogs. In the case of round robin, queues do not have priorities but each queue has a fair access to the limited network resources and prevents a bursty queue from consuming more than its fair shares of bandwidth. According to [13], queues are serviced one packet at a time in cyclical order. Empty queues are skipped. In WRR, there is a pre-assigned variable per queue called the *weight,* which specifies the share rate of the resources due to each service queue [14, 15, 16]. Once the weight is set, and some queues are backlogged, a circular scan is made to all the queues to select approved number of packets for multiplexing by the transmission link. At each visitation, a WRR scheduler services a queue if its head-of-line packet size is less than its remaining fair share. Weighted round robin is commonly used in a broad range of critical computing and communication systems like operating systems, CISCO router and Asynchronous Transfer Mode (ATM) network.

III. Simulation Scheduling Scenarios with RIO-C Buffering Scheme

Buffering mechanisms are best assessed with levels of strictness of its algorithm and its packet loss rate. In this report, algorithms of RIO-C buffering scheme is simulated with the three scheduling disciplines discussed above using network simulator 2. The topology used is as shown in Fig. 2. Two UDP sources (S1 and S2) are configured to send traffics to the same destination D through edge E1, core C1 and edge E2.

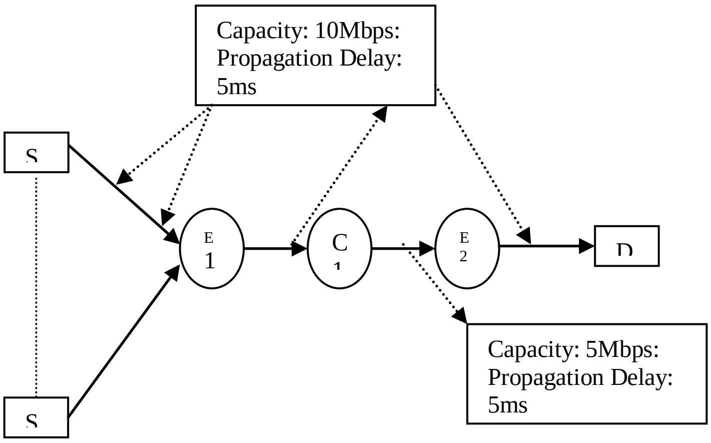

Fig.2. simulation topology

According to DiffServ architecture in [2], edge router conditions and classifies packets using associated differentiated service code point and service level agreement (traffic profile) while core router only buffers and schedules packet based on the setting of the edge router. In the above settings, queues are built at both edge E1 and core C1 facilities because packets arrival rate to them exceeds the available bandwidth [4]. The network link of 10Mb bandwidth with a link propagation delay of 5ms is connected from each source to edge 1. From the core router C1 to the edge E2, the bandwidth is set to 5Mb so as to allow burstiness in traffic and to study the effect of congestions at the core router C1. The UDP packet which requires no acknowledgement is generated using constant bit rate traffic generator within ns 2.

Token bucket policer is used for traffic conditioning. The meter monitors the sending rate of each traffic source and determines whether a packet is in-profile or out-of-profile. Three experimental scenarios were conducted using the same topological setup; RIO-C with priority scheduling algorithm, RIO-C with round robin algorithm, and RIO-C with the weighted round robin algorithm. Traffics from edge E1 to core C1 were grouped into two physical queues, each having two virtual queues (or precedence levels). The buffer configuration is set in ns 2 as:

```
$qE1C1 configQ 0 0 20 40 0.02
$qE1C2 configQ 0 1 10 20 0.10
```

This script configured the RIO-C physical queue 0, virtual queue 0 (in-profile) to 20, 40 and 0.02 minimum packets, maximum packets and maximum probability thresholds (RED max$_p$) respectively. While traffics in physical queue 0 and virtual queue 1 (out-of-profile) used 10, 20 and 0.10 minimum packets, maximum packets and maximum probability respectively. For the schedulers, the order in which the queues are visited is organized through an active list structure that keeps references of all queues that are backlogged.

3.1 Performance Metrics Used

As said earlier, this study aimed at measuring the strictness of the RIO-C buffering scheme with the loss rate of the schedulers. This can be determined using the packet loss rate due to RED algorithm and loss rate due to the scheduling algorithm. Loss rate is the ratio in percentage of packets that were dropped (or lost) by RIO-C-based router during an interval of time to the total number of packets that arrived to the router [17]. Such packets are referred to as being lost on transit. i.e.

$$\text{Loss Rate \%} = \frac{N_L *100}{N_A} \qquad (2)$$

where N_L and N_A are the numbers of lost packets and total arrived packets respectively.

IV. Results and Discussion of Results

After running the simulations, using the experimental setup describe in Figure 2 in previous section, events of what happened between the core C1 link to edge E2 link were traced at some time intervals into respective files. On analysing these files, the following results were obtained for packets drop (loss) rate statistics.

Table 1 to Table 3 show the number of packets that were dropped by the schedulers. Plotting this into column chart gave the graphical analyses in Fig. 3 to Fig. 5. As shown in Fig. 3, compliant queues Q10 and Q20 witnessed low drops (24 and 77 packets respectively) compared to non-compliant queues Q11 and Q21 (with 1272 and 4639 packets respectively) in the first interval. This condition persists in the other intervals. This means the scheduler maintains compliancy while servicing the queues. Fig. 4 and Fig. 5 also witnessed the same policy while using the round robin and weighted round robin respectively.

Table 1: Packet drop Analysis- Priority scheduler

Queues	1st Interval	2nd Interval	3rd Interval	4th Interval
Q10	24	24	24	24
Q11	1272	2586	3936	5303
Q20	77	77	77	77
Q21	4639	9148	13677	18160

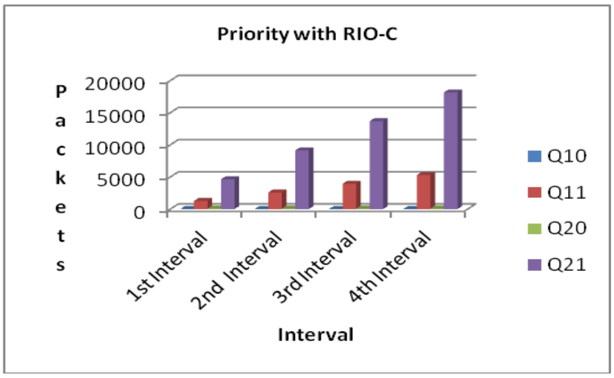

Fig. 3: lost packets using priority scheme

Table 2: Packet drop Analysis- Round Robin scheduler

Queues	1st Interval	2nd Interval	3rd Interval	4th Interval
Q10	28	28	28	28
Q11	2916	5869	8821	11804
Q20	15	15	15	15
Q21	2921	5885	8835	11780

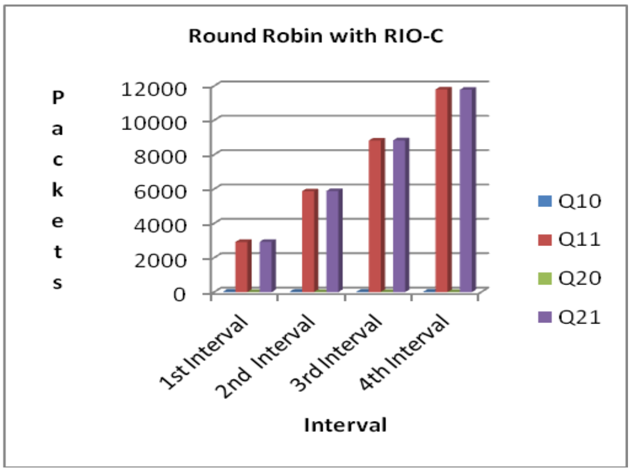

Fig. 4: lost packets using round robin scheme

Table 3: Packet drop Analysis-Weighted Round Robin scheduler

Queues	1st Interval	2nd Interval	3rd Interval	4th Interval
Q10	119	119	119	119
Q11	5871	11894	17924	23951
Q20	0	0	0	0
Q21	65	93	146	152

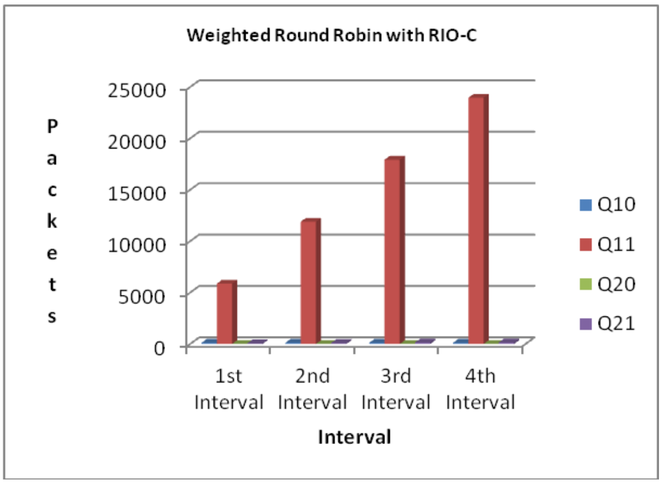

Fig. 5 lost packets using weighted round robin scheme

In terms of scheduler drop rate

Comparing the performance of these schedulers using the loss rate, Table 4 shows the ranking of the schedulers and this is also depicted in Fig. 6. Note that the lower the loss rate by the scheduler, the better the scheduler and that the higher the loss rates by the RED algorithm, the stricter the algorithm [11]. In Table 4, the priority scheduling scheme ranked first with loss rate of 29.45%. The round robin scheduling scheme ranked second with loss rate 29.53% while weighted round robin ranked third with 30.28%.

Table 4: Performance ranking of the schedulers based on loss rate

Schedulers	Drop Rate (%)	Rank
PRI	29.45905062	1st
RR	29.53781145	2nd
WRR	30.28166373	3rd

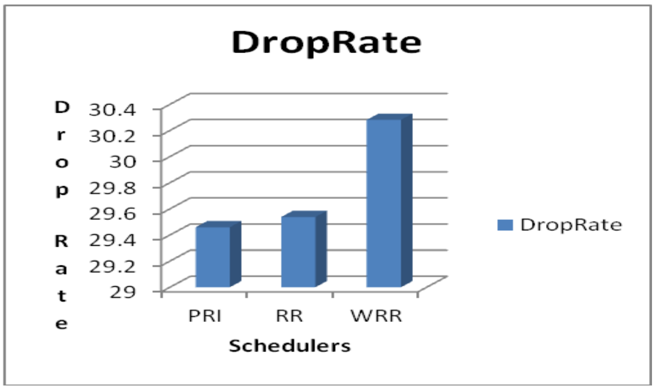

Figure 6

In terms of the strictness of the buffer scheme

Statistics of packets that were dropped due to RED algorithm were analysed using the out-of-profile service queues. In ns 2, the packets that were probabilistically dropped are kept as e-drop packets. This record was analysed using the loss rate metric as well. The result is as shown in Table 5.

Table 5 Performance ranking based on strictness of the buffer scheme

Scheduler	E-drop loss Rate (%)	Rank
PRI	14.95	1st
RR	10.50	2nd
WRR	3.04	3rd

In Table 5, priority with RIO-C ranked first implying that if the Internet of the future generation seeks after strict traffic policy/admission control, priority scheduling with RIO-C would be of better choice. Priority in itself is noted for jeopardizing other queues from receiving service but when policy-based buffer scheme is enforced, other service queues will have the advantage of receiving service. It is a matter of setting strict service profiles for applications of higher priority level.

In summary, from the simulation experiments and results, it can be concluded that priority disciple with RIO-C can be used as scheduling and buffering scheme for the next-generation Internet.

V. Conclusion

This paper followed up on the report in [11] where RIO-C and RIO-D were compared to determine which one would strictly enforce traffic service level agreement proposed for differentiated services paradigm. In this present report, it is realised that the impact of the schedulers also need to be considered with the buffer algorithm. This led to the simulation using the suggested buffering scheme RIO-C with three popular scheduling schemes in literature: priority, round robin and weighted round robin. The results of the simulations were diverse but if logically scrutinized, it turned out that if admission control is handled by the RIO-C algorithm, then priority scheme can be extended to support service differentiation in the next generation Internet routers.

Since these algorithms simulated are applicable to wired network, the future extension of this work is to see how the choice of better algorithms could be adapted to wireless systems.

References

[1] R.Braden, D.Clark and S.Shenker *Integrated Services in the Internet Architecture: an Overview*, (IETF Draft, RFC.1633) 1994 Available:from:http:// en.scientificcommons.org/42772402.

[2] S.Blake, D. Black, M. Carlson, Z. Wang. And W. Weiss W. *An Architecture for Differentiated Services*, (IETF Draft, RFC 2474, 1998).

[3] D. Stiliadis, A. Varma *'Efficient Fair Queuing Algorithms for Packet Switched Networks'*, IEEE/ACM Transaction on Networking Vol. 6, 1998.

[4] R. Jain '*Congestion Control in Computer Networks: Issues and Trends'*, IEEE Network Magazine, pp. 24-30, 1990.

[5] Kleinrock K ‘Queuing systems (volume II computer applications’, .1986) http://www.cs.ucla.edu/~lk/full_bibliography.html

[6] R. Braden and D. Clark *Recommendations on Active Queue Management and Avoidance in the Internet'*, Internet Draft, RFC 2309, 1998

[7] S. Floyd and V. Jacbson Random Early Detection Gateway for Congestion Avoidance, *IEEE/ACM Transactions on Network, 1(4)*, 397-413, 1993

[8] D. Clark and W. Fang Explicit Allocation of Best Effort Packet Delivery Service, *ACM Transactions on Networking, Vol. 6(4)*, 362-373, 1998

[9] I. Teerawat and H. Ekram Introduction to Network Simulator ns 2 (Springer Science Business Media, LLC, 233 Spring Street, New York, NY, 2009)

[10] K. Fall and K. Varadhan, *ns Notes and Documentation* http://www.isi.edu/ nsnam/ ns/ns-documentation.html

[11] F.A Oladeji, I.O Onadokun and M.O Oyetunji Evaluating Buffering Schemes for Next-Generation Internet QoS Router, *International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181*, www.ijert.org Vol. 2 Issue 6, June 2013

[12] Cisco Technical Specification, Distributed Weighted Random Early Detection, *Release 11.1(17)* URL:http:// www.cisco.com/univercdlccltdldoc/product/ software/ios Ill/cc 111/wred.pdf.

[13] C. Semeria *Supporting Differentiated Service Classes: Queue Scheduling Disciplines*, (White paper, Juniper Networks Inc., 2001) http://users.jyu.fi/~timoh/kurssit/verkot/scheduling.pdf

[14] L. Luciano, M Enzo and S. Giovanni Tradeoffs between low complexity, low latency, and fairness with Deficit Round Robin Schedulers, *IEEE/ACM proceeding on Networking, Vol. 12(4), pp. 375-385*, 2004

[15] Y. Hyun-Ho, K. Hakyong, O. Changhiwan. and K Kiseon A Queue Length-based Scheduling Scheme in ATM Networks, IEEE 1999, pp. 234-237, 1999.

[16] W. Heng-Yi, C. Min-kuan and C. Chia-Cung. *The Switch-Board Sub-carrier Allocation Policies in Multi-Service OFDM Systems*, IEEE 2006 pp. 1328-1332, 2006.

[17] I. Widjaja *communicationn networks – fundamental concepts and key architecture* (McGraw-Hill Inc. 2000)