CWRR: A Scheduling Algorithm for Maximizing the Performance of Quality of Service Network Router

Oladeji F.A.
Department of Computer Science University of Lagos
Lagos, Nigeria

Oyetunji M.O.
Department of ComputerScience and InformationTechnology
Bowen University Iwo, Nigeria

Okunoye O.B.
Department of Computer Science University of Lagos
Lagos Nigeria

ABSTRACT

This paper presents a novel approach for implementing quality of service as demanded by evolving applications in the Internet. For some decades now, research efforts have led to the extension of the TCP/IP in order to make the Internet a full-fledged quality of service network. Novel in the extension is the invention of the Integrated services and Differentiated services architectures. The Differentiated services architecture was widely accepted among researchers because of its scalability. In order to achieve some of the refinements to the current TCP/IP protocol by the IETF for DiffServ implementation in the Internet, new traffic management mechanisms such as differential packet buffering cum differential allocation of available link bandwidth are needed. This report studied some suggested scheduling algorithms in literature on how to incorporate a multi-queue paradigm and enforce service level agreement in the Internet. A new scheduling model that ensures maximum utilization of network bandwidth is used to assess experimental implementation of Differentiated services in a QoS-based router. The model, termed, carry-on Weighted Round Robin (cWRR) proved better than the original Weighted Round Robin (WRR) scheme in terms of low higher throughput and fairness to traffic sources in a multi-queue network core router paradigm.

General Terms

Differentiated Services, Quality of Services, and Weighed Round Robin

Keywords

DiffServ, QoS Router, Scheduler, TCP/IP, WRR

1.INTRODUCTION

A Router is an indispensable network device that receives traffic streams coming from various sources or other routers and forwards them to the next intermediate router or destination router. A key function of this device operating under a multi-queue paradigm is the scheduling approach through which it decides which packets and from which queue to transmit next. According to [1], the next generation Internet router can implement up to 64 different queues, each queue keeps packets of a particular group of application that expects same treatment from the network. In order to grant certain required quality to a network application, service level agreements between the traffic sources and the network needs to be enforced at the routers. Accordingly, the source specifies a particular request (traffic spec) while the network determines the estimated the sending profile of the source. Any violation may lead to unexpected treatment of the extra traffics from the source. The authors in [2] places emphasis on the importance of a router scheduling algorithm to be a procedure that influences three orthogonal traffic management functions: buffering of packets, ordering of packets for transmission and controlling of congestion by dropping packets if need be.

These key functions of a network router scheduling algorithm have led researchers to propose different algorithms for supporting many queues and satisfying various requirements of each traffic group in DiffServ. Most of the algorithms are criticized in terms of flow isolation, delays experienced by traffic packets at the router, fairness to traffic sources, simplicity of implementation, utilization and scalability among others. Isolation in the sense that queues that violates their sending rates should not carry their burdens to other cooperative traffic queues. Scheduling algorithms can be grouped into three: frame-based, time-stamped and the hybrid of the two. Frame-based shares the available resources among queues as quantum and cyclical services the queue as much as the quantum could allow [3, 4, 5, 6, 7]. The time-stamped counterpart attends to the traffic according to the time required by each packet for execution. While some frame-based scheduling algorithms such as in [ ] were noted for higher packet delays, some sorted algorithms like [8, 9, 10] are condemned because of their higher per packet processing complexities.

Weighted Round Robin (WRR) is a popular algorithm that many researchers had modified or extended for distributing resources at the routers because of its good traits in terms of fairness and minimum bandwidth reservation. As good as the algorithm, it has been criticized vehemently in terms of network efficiency. If a router can transmit as much as C bits per second and all that it was given to the device out of A bits that arrived for onward transmission is D, then the efficiency of the device over that time unit is D/C. The deficiency in WRR was traced to its logic in dealing with un-used quantum across queues in a service round [11, 12, 13].

The approach being introduced to the algorithm in this report is to utilize the un-used quanta in the same round. With this extension and series of simulations, a better performance of network router was achieved. The paper is presented as follows: the first section introduces router and its scheduling algorithms; the second section summarizes the previous work on extending WRR. The new logic that this paper used for the simulation is presented in section three while results and results discussion is presented in section four. Section five draws up the conclusion.

2. RELATED WORK

Weighted round robin (hereafter refers to as WRR) is a frame-based scheduling algorithm commonly used in a broad range of critical computing and communication systems like operating systems, CISCO router and Asynchronous Transfer Mode (ATM) network [14]. Compared to the sorted algorithms, the scheme seeks after differential allocation of available resources among competing traffic queues at per-packet complexity as low as O(1) [4]. In WRR, there is a pre-assigned variable per queue called the *weight,* which specifies the share rate of the resources due to each queue [5,6]. 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. The order in which the queues are visited is organized through an active list structure that keeps references of all queues that are backlogged. At each visitation, a WRR scheduler services a queue if its head-of-line packet size is less than its remaining fair share (or quantum).

Let us have a walkthrough of WRR logic. The status of the queues is given after three successive time intervals, where it is assumed that no further packets arrive during the period under consideration. Assuming four queues designated as Q0, Q1, Q2 and Q3 having bandwidth value of 2000bytes and sharing weight of 40%, 30% 20% and 10% respectively. All packets in all queues are assumed to be of length 300 bytes. In the first round, 2 packets are moved from Q0 to the output queue. This is because its weight allows for a removal of no more than 2000 * 40% = 800 bytes= 2 packets. For similar reasons, 2 packets are removed from Q1 to the output queue and 1 packet from Q2 and 0 packets from Q3 since the fair share of 200bytes is not enough to carry any packet. This means that a total of 1500 bytes have to be transmitted on the link whose capacity is 2000 bytes. Hence, there is an un-used capacity of 500bytes after the first round. This may be applicable in other rounds.

Despite its choice in some switching systems, a closer study of the discipline revealed some weaknesses in managing resources at its disposal. WRR introduces two types of resource wastage: one caused by service queue that becomes empty at the end of a round and the wastage that occurred when a backlogged queue is denied transmission because its packet size exceeds the remaining quantum.

These issues have attracted considerable research efforts in the open literature leading to the modification of the **algorithm in** different ways [4, 5, 6, 7, 11, 12, 13]. The issues are defined as follows:

2.1 Queues Becoming Empty at the End of a Round

In this case, after servicing a backlogged queue in the present round, it renders the queue empty. It is possible that the said queue did not consume its quantum in which case the remaining quantum is wasted. In the next service opportunity, depending on the policy on weight adjustment, the new quantum of empty queues are shared and added to the next calculated quantum of backlogged queues. This is assumed in [7]

2.2 Backlogged Queues Denied of its Remaining Quantum During a Round

The authors in [11] pointed out that WRR performance is degraded when confronted with situation of bursty traffic and this has been transferred to deficit weighted round robin. The flaw is well noted when the allocation is based on quantum calculated on size of available link bandwidth as done in [6, 7]. On such computation, WRR services a queue if its remaining quantum is greater than the packet size at the head of the queue. The unused quantum is credited in Deficit Weighted Round Robin (DWRR or simply DRR) to be used in the next round [15]. For optimal use of available resources in a service opportunity, the sum of resources used in a round should approach the available capacity if there are still backlogged queues i.e.

$$\sum_{i}^{n} S_i = C \tag{1}$$

where S_i is the amount of resources used by queue i, C is the available bandwidth and n the number of active queues. Thus, we can see that even though the system is overloaded, the available resources are not fully utilized because of resource provisioning power of WRR.

In order to extend or review the logic of weighted round robin, the authors in [11] took an approach by borrowing more credits against next weight reset. In their modification, instead of denial of transmission, the scheduler allows such queue to use more resources than expected. This may only work when the amount of resources available is not taken into consideration. If quota system is used, the cumulative bandwidth may exceed the link bandwidth. Such an illusion may not be feasible in reality

In accordance with bandwidth borrowing ahead of a round, [16] allows a queue, depending on its input rate status (over-limit, under-limit or at limit), to borrow in order not to miss its fair share. For example if packet size exceeds quantum but the status of the queue is at limit or under-limit, the packet is sent and the excess quantum used is deducted from the queue's share in the next round.

A further analysis of WRR led to the design of DRR in [15] where unused credits accumulated from previous rounds are added to the next round quantum before the commencement of the round. Several variants of deficit round robin exist in literature such as DRR+, DRR++ whereby instead of transmitting the whole lump of packet share at time, an inner service round robin is introduced.

Authors in [7] also addressed the problem of unused transmission slot while proposing a Modified Weighted Deficit Round Robin (MWDRR) scheme for Passive Optical Network (PON) switching devices. The modified logic accumulates the current deficit counters at the end of a round and makes use of it at the beginning of the next round. The scheme then resets deficit counters of all queues to zero before refreshing the normal frame cycle. The proposed model is straight forward to implement and does not alter the normal sequence of servicing the queues.

In [17], it was suggested that if more credits are given to bursty queues in WRR, cell loss ratio and delay in ATM traffic queues can be reduced and performance improved. They then proposed a threshold model where queue developments are monitored against the set queue thresholds. Any time an arrival of new packet causes the observed queue length to exceed the threshold; a weight-up factor μ_i is added to the previous weight. A fixed scalar weight was used for the simulation.

[18] also redesigned WRR to suit delay-constrained queues. The authors condemned allocation of bandwidth that does not depend on actual queue load variations and sought to provide absolute delay constraint in short time scales. Thus, any time the delay-constrained class is temporarily overloaded, the WRR dynamically assigns a larger share of resources to its queue. Consequently, extra bandwidth will be to the detriment of other service queues which may even have traffics that have stayed longer than the traffic of the so-called delay sensitive queue.

A closely related modification is the work presented in [13] called carry-over round robin. In the article, unused quanta are used in the minor cycle which is another set of inner rounds. This introduces a little delay in the congested network.

CWRR Logic

The modification introduced into the existing WRR algorithm is that, at any time the available quantum of a service queue is less than the size of packet at the head of the queue, cWRR gives the remaining quantum to the next active queue rather than ignoring it as in WRR, or count it as deficit against the next round as in DRR. Each extra packet serviced is noted as bonus or improvement. Thus, the unused bandwidth is effectively utilized in the same round. The adjustment takes place in the same round and not postponed to the next round as most modifications had done.

A service queue in cWRR is considered active during a round if it has packets awaiting service and the scheduler is a work-conserving scheme in that it schedules as long as there are packets in the system. In order to implement the algorithm, an active list is maintained to hold the index of all active service queues (ActiveList) and an array variable is also defined to hold the quantum of the queues.

As a packet arrives to a passive service queue, its index is added to the tail of the ActiveList. A round in cWRR is defined as one round robin iteration during which the cWRR serves packets from all the service classes whose indices are present in the ActiveList until there is no more bandwidth to use. The queues' fair shares are refreshed at the beginning of the each round. The refreshing operation has to do with the active queues and not on each packet. Also, the comparison to determine whether a queue will be allowed to transmit is also done per queue and not per packet. The permission to continue to schedule if there is still bandwidth requires only one comparison at the end of normal round and is also one operation as in normal WRR. Thus, cWRR is also a variant of WRR scheme with O (1) per packet complexity.

3. cWRR IMPLEMENTATION IN DIFFSERV

A network scheduling has two main procedures; which are

i) Packet enqueue module (with the dropping process)

ii) Packet dequeue module

The enqueuing process is a scheduler activity in which incoming packets are accepted into the buffer while the dequeuing process is the act of taking packets out of the buffer. Enqueuing is a function of an active buffer management. The dequeuing function is the sole responsibility of a network scheduler. In the interim, the scheduling process in athat supports differentiated services is as follow:

Process at the Classifier module:

READ: reads a packet from input port buffer (those that are permitted by the conditioning routines)

CLASSIFY: determines its code point

ENQUEUE: enqueues packet on appropriate output port physical queue if existing or creates an active queue if the queue has being in passive state, or drops the packet if there is no approved buffer

Scheduler process at the output interface:

SELECT: selects the physical queue to transmit packets from,

DEQUEUE: removes the packet at head of this queue.

WRITE: writes the packet to the outgoing packets' buffer.

The simulation experiments were carried out using four queues and four scheduling disciplines: strict priority (PRI), simple round robin (RR), weighted round robin (WRR) and carry-on weighted round robin (cWRR). Since the router system is better studied in a steady state, the experiments were allowed to run for 120 seconds and statistics taken after 30 seconds. Subsequent statistics were recorded at 30 seconds interval. Out of about 900,000 packets that were generated, the first 120,000 packets were ignored such that the analysis could commence when the system is assumed to be stabled. Packets of equal size are generated.

The study aims at achieving a network close to the one shown in Fig. 1 whereby network sources in DiffServ domain send packets to an ingress router which implements two routing algorithms in series; edge and core routines. At the edge level, traffic conditioning functions like metering, shaping and remarking are carried out to enforce traffic profiles compliancy. Traffics that scale through the conditioning tests are forwarded to the core routine where the forwarding takes place.

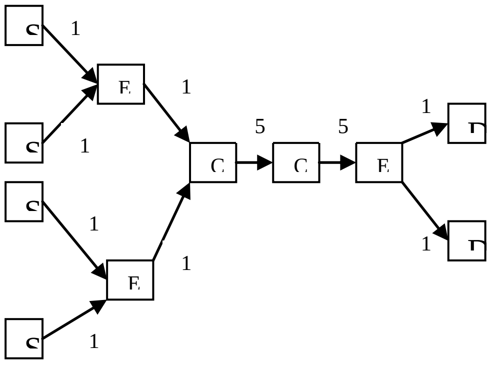

Fig 1: Simulation Topology

The links from the sources to the edge routers (Edge1 and Edge2) and from destinations to the edge router 3 are configured to have bandwidths of 10Mb with link delays of 5ms. From core1 router to core2, core2 to edge router 3, the setting of the bandwidth is 5Mb and delay of 5ms. The parameters are set to allow burstiness in traffic and to study the effect of congestions at the core routers. The sources S_0 and S_1 send to destination D_1 through edge1 while S_2 and S_3 send to destination D_2 through edge2. According to the prescribed DiffServ network domain, edge router measures traffic streams, ensures compliancy and classifies packets using an agreed code called Differentiated Service Code Point (DSCP). Violated traffics downgraded to a higher drop precedence virtual queues.

With the above parameter settings, queues are built at both the edge and the core facilities since the arrival rate to them exceeds the available bandwidth of the forwarding engine. In the experiments, what happened to each packet passing through the core router was traced. Packet streams from core 2 to edge3 (c2e3) was traced and analyzed The one-way average packet delays, for the scheduling algorithms and for the queue designated to hold real-time traffics like the voice, were computed. Also, the achieved throughputs over some time intervals, and the fairness indices of the schedulers were computed. The simulation was run for 120 seconds and took queuing statistics after every 30 seconds. Thousands of packets of different types were generated within the simulation time. Four service queues (physical) were simulated and are tagged Q10, Q20, Q25 and Q30. The study used RIO (Random Early Detection In-profile Out-of-profile) buffering approach. With RIO, violating packets from each physical queue are downgraded into respective virtual queues and are tagged Q11, Q21, Q26 and Q31. These queue naming conventions are by assumption.

4. RESULTS AND DISCUSSION OF RESULTS

The table 1 below shows sample packet statistics recorded during the simulation. Relevant fields are: packet status event (+ for enqueue, - for dequeue, d for drop and r for receive), time of event, type of traffic (tcp for TCP traffic, paroo for Pareto on/off application such as web, expoo for exponentially generated on/off application like the voice and finally the cbr for constant bit rate application) and the packet's unique identification number. A packet may be enqueued at a router, scheduled or dropped.

Table 1. Sample packet traces In ns 2

Packet Status	Time (s)	Traffic Type	Packet Size	Packet ID
-	73.388	pareto	520	162453
+	73.389	pareto	520	162457
-	73.389	pareto	520	162457
r	73.389	Tcp	520	162442
+	73.389	pareto	520	162460
-	73.393	pareto	520	162460
r	73.875	exp	520	163541
+	73.875	exp	520	163555
-	73.875	exp	520	163555
r	73.875	exp	520	163542

The performance of the core router is based on the following parameters:
One-way average delay: This is obtained by finding the difference between arrival time and departure time of each packet according to the method described in [14, 19]. This delay was calculated for the scheduling algorithms WRR and cWRR. Also, the one-way packet delay experienced by packets of the queue designated as real-time was calculated. i.e. physical queue Q10. The baseline is that the lower the delay the better the performance.

Table 2: One-way Average Packet Delay from core 2 to edge 3 (in seconds)

Object Type	WRR	cWRR
For Scheduler	0.00538	0.00514
Real-time Source Only	0.00063	0.00061

Comparing WRR and cWRR, cWRR has the lower average delay. For the real-time traffic queue, the average delay experienced by cWRR real-time packets is less than that of other schedulers.

Throughput: Comparing WRR and cWRR in terms of throughput, Fig. 2 and Fig. 3 show the histogram of scheduled traffics using WRR and cWRR algorithm respectively at four time intervals of 30 secs each. The difference of throughput achieved was computed and plotted into histogram shown in Fig. 4.

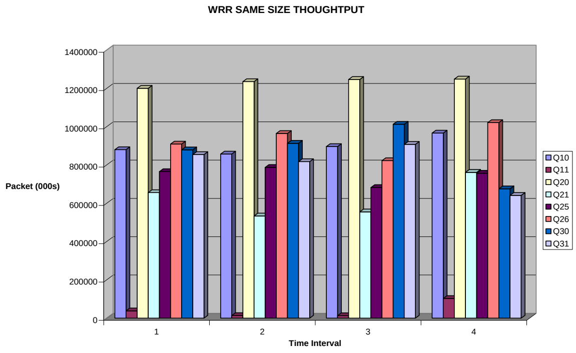

Fig 2: Throughput Analysis of WRR Discipline

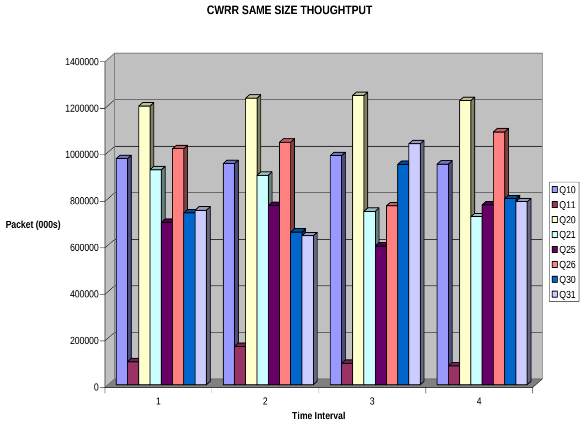

Fig 3: Throughput Analysis of WRR Discipline

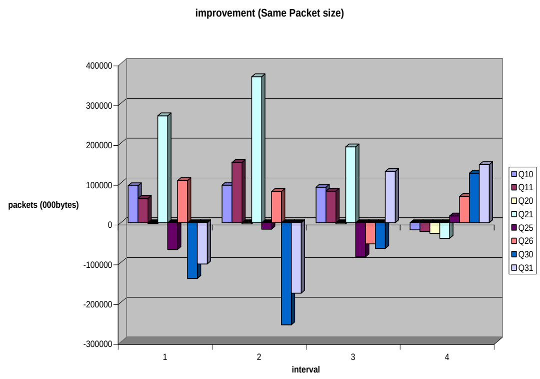

Fig 4: Throughput Improvements of cWRR over WRR Disciplines

Fairness Analysis**:** In order to compare the fairness indices of the schedulers, Jain index was used. This index finds the square of the sum of service received by all the queues and divide by the product of sum of squares of the bytes scheduled based on the number of active queues. Based on this performance metric, cWRR is fairer than WRR with 95.2% and 94.5% respectively.

With the above statistics, Table 3 give the summary of the scheduling polices with respect to their positional ranking.

Table 3: Performance Ranking

Evaluating Metrics	Object type	WRR	cWRR
Average Packet Delay	Scheduler	4th	3rd
	Real-time	2nd	1st
Throughput	Scheduler	2nd	1st
Fairness	Scheduler	3rd	2nd

5.CONCLUSION

In this research study, quality of service traffic management mechanisms with respect to differential scheduling and buffering of traffics in DiffServ domain network were presented. A probable solution to one of the weaknesses of weighted round robin scheme in terms of the unused quantum was proposed. Simulations and a comparative study with the original logic and some frame-based scheduling algorithms revealed a better performance of the modified version. The good features of the original weighted round robin algorithm are retained in the modified version. The testbed was based on the setting platforms of next generation Internet (Differentiated Services).

6. REFERENCES

[1] S. Blake, D. Black, M. Carlson, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services”, IETF Draft, RFC 2474, 1998.

[2] D. Stiliadis and A. Varma , “Efficient Fair Queuing Algorithms for Packet Switched Networks”, IEEE/ACM Transaction on Networking Vol. 6 April 1998.

[3] C. Semeria, “Supporting Differentiated Service Classes: Queue Scheduling Disciplines”, White paper, Juniper Networks Inc., 2001.

[4] L. Luciano, M. Enzo, and S. Giovanni, “Trade-offs between low complexity, low latency, and fairness with Deficit Round Robin Schedulers”, IEEE/ACM on Networking, Vol. 12(4), Aug. 2004.

[5] W. Heng-Yi, C. Min-Kuan, and C. Chia-Cung, “The Switch-Board Sub-carrier Allocation Policies in Multi-Service OFDM Systems”, IEEE 2006 pg. 1328-1332

[6] S. Hideyuki, Y. Makiko, F. Ruixne, and S. Hiroshi, “An improvement of WRR cell in scheduling in ATM Networks”, IEEE 1997 pg. 1119-1123

[7] L. Dong-yeal and O. Seung, “A new DBA Scheme to Improve Bandwidth Utilization in EPONs”, ICACT2006, ISBN 89-5519-129-4, Feb 20-22, 2006, pg. 1063-1067

[8] S. Arunabha, M. Ibraz, S. Ravikanth, and B. Subir, “Fair Queuing with Round Robin: A new Packet Scheduling Algorithm for Routers”, Proceedings of the Seventh International Symposium on Computers and Communications (ISCC ’02), ISDN:1530 134602 IEEE, 2002, pg. 101-106.

[9] P. Goyal, and V. Harrick, “Generalized Guarantee Rate Scheduling Algorithms: A framework”, IEEE/ACM Transaction on Networking, Vol. 5(4) Aug. 1997.

[10] S. Golestani, “A Self-clocked fair Queuing Scheme for Broadband Applications”, Proceedings of IEEE INFOCOMM 1994, pg. 634-646

[11] H. Yoshihiro, T. Shuji, and I. Yutaka, “Variably Weighted Round Robin Queuing for Core IP Routers”. IEEE/ACM, 2002, pg. 159-166

[12] S. Hideyuki, Y. Makiko, F. Ruixne, and S. Hiroshi “An improvement of WRR cell in scheduling in ATM Networks”, IEEE 1997 pg. 1119-1123

[13] D. Saha, S. Mukhejee, and S. Tripathi, (1996) “Carry-over Round Robin: A Single Cell Scheduling Mechanism for ATM Networks” IEEE Journal 0743-166x/96, 1996, pg. 630-637.

[14] K. Mezger and D. Petr, “Bounded Delay for WRR”, Technical Report, Dept. of Electrical Engineering and Computer Sciences, University of Kansas 1995

[15] M. Shreedar and G. Varghese, “Efficient Far Queuing using Deficit Round Robin”, IEEE/ACM Transactions on Networking, Vol. 4(3), Jun. 1996.

[16] G. Mamais, M. Markaki, G. Politis and I. Vernieris I., “Efficient Buffer Management and Scheduling in a Combined IntServ and DiffServ Architecture- A Performance Study”, Technical Report, University of Virgina, Mar. 2004.

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

[18] M. K. Kim and H. S. Park, “Safeguarding self-similar traffic in packet-switching System with High Utilization”, IEEE/ACM Trans on Networking, Vol. 5(4) Aug. 2004.

[19] Yuming Jiang, “Network Calculus and Queuing Theory: Two Sides of One Coin”, Proceeding, VALUETOOLS Conference, Pisa, Italy Oct 20-22, 2009.