A Revised Logic for Weighted Round Robin Scheduler towards Improving Network Throughput in a QoS Internet

By

[1]F.A. Oladeji, [2]C.O. Uwadia, [3]D. G. Kourie and [4]M.C. Strauss

[1,2] Department of Computer Science, University of Lagos

[3,4] Fastar/Espresso Research Group, Department of Computer Science, University of Pretoria, South Africa

Abstract:

Attempts to support service differentiation and QoS on the Internet have brought about various traffic management models. One such class of models deals with traffic scheduling at the Internet routers. A network scheduler generally implements three orthogonal policies, namely: a scheduling policy, a queuing policy and a packet drop policy. While the three policies work together to prevent a network collapse in face of congestions, this paper focuses on weighted round robin (WRR) policy and seeks how its performance can be improved in the face of bursty traffic situations. WRR allocates available resources in proportion to the prescribed weight. The early implementation of the scheduler was faced with performance degradation when the algorithm meets with bursty input traffic. In such a situation, queues might not be served even though there is still bandwidth. This paper presents dynamic algorithm that caters for such lapses in this scheduler. Our scheme tagged carry-on WRR (cWRR) made sure the remaining bandwidth is used up by other backlogged queues in the same round. The new scheme was simulated in Java and our observation shows that the throughput achieved was improved when compared to ordinary WRR scheduling discipline.

1.0 Introduction

The Internet architecture has been subjected to modification in order to support vast emerging real-time services and applications. The coordinating body, the Internet Engineering Task Force (IETF), in its recommendations has proposed two architectures for achieving QoS in the ever-growing network: the Integrated Services (IntServ) architecture [4] and the Differentiated Services (DiffServ) architecture [2]. These two architectures were designed because the original Internet architecture could only provide best effort services where all traffic flows are given the same treatment at the routers. The emergence of voice, video and multimedia communication technology has warranted a new design for the Internet infrastructure. The IntServ architecture, with its associated Resource reSerVation Protocol provisioning models [5], has been criticized because of its ambition to provide differentiations based on fine-grained granularity of traffic flows. It attempts to maintain a queue and per-flow statistics for each traffic flow emanating from a source to a destination. The result would be a composition of millions of flows in the Internet.

requiring large overheads to keep each flow's statistics at IntServ routers.

Intserv's counterpart, DiffServ, seeks to aggregate flows into classes based on their specific service level agreements, providing per-hop treatment as specified by the traffic source or provider. This per-class treatment gives DiffServ the edge in implementing service differentiation and QoS at the routers.

Two issues are relevant in a QoS support network: how to provide service differentiation and service performance assurance. Since the Internet best effort model has limitations in addressing these issues, solicitations in the form of Request For Comments (RFCs) by the IETF have drawn the attention of network designers and analysts, who have proposed models and protocols for the Internet to render it a multi-service network [6].

To date, many service differentiation approaches have been postulated for various traffic management functions. By traffic management functions, we mean a set of network actions that monitor and control the flows of traffic in a network. Examples of such functions defined by ITU-AT and the IETF are traffic conditioning, admission control, packet buffering, packet scheduling, quality of service routing, network billing and a host of other menial functions defined in [7]. Service differentiation could be achieved by modifying some of the existing traffic management functions. For example, different metering rates can be used to mark different traffic types at the traffic admission meter as well as applying differential bandwidth allocation using differential weights or priority levels across the proposed multi-queues. This paper seeks how network router's performance can be improved through efficient traffic scheduling mechanism.

The other issue of relevance in a QoS support network is achieving performance assurance. How to satisfy the QoS requirements of competing flows, in a packet-switched network with decentralized routers, is a long-standing problem that has attracted considerable research in the Internet community [23]. Most of the new evolving real-time applications require minimum resources to operate effectively compared to original Internet best-effort bulky data. . Listed below are traffic groups and their quality expectations [8]:

- Those requesting a static amount of bandwidth that must be continuously available during connection lifetime (long duration flows). In this application group are videoconferencing, interactive audio (telephony), Audio/Video distribution (Television, Distance learning), audio/video retrieval (Video on Demand, audio library).
- Those requesting tightly constrained delay and delay variations and fixed transmission rate. In this group are voice, multimedia applications.
- Those requesting service guarantees (assured bandwidth) from a network. In this group are airline reservation systems, on-line transaction systems, process monitoring systems and network control messages.

The models that could guarantee the specific needs of a flow or class of flow are needed in the Internet. The major concern among researchers is how to manage the available resources, while adaptively and maximally satisfying growing traffic requirements.

Network scheduling intends as a viable weapon to

sustain QoS in the Internet. It offers differentiation capability and a mechanism to predict the possibility of attaining desired quality metrics. For these two reasons, in recent years, efforts have been directed to ways of building QoS performance-oriented schedulers on the Internet. Existing QoS schedulers attend to traffic flows in different ways. A traffic flow is a stream of packets that traverses the same route from source to the destination and requires the same grade of service at each router or gateway in the path [31]. In this section we concentrate on WRR; one of the schedulers that are commonly used at the routers, some popular scheduling disciplines are described in section two.

WRR is an extension of the round robin scheduler [11, 12, 13, 18, 22, 27 29]. Round robin schemes guarantee a minimum resource to every backlogged queue not minding the arrival or departure times of those packets. It attains its fairness by assigning weights to the traffic queues and service in a round a number of multiplexed packets selected from each queue. According to [19], in a WRR scheduler, the maximum amount of service that a queue receives in each round is upper-bounded by its reserved allocation based on its weight. As such, no class can consume more service than what has been assigned. In that case, weight reset may not be applicable to the present service round. Scheduling takes place from the heads of queues and whether packets from a queue will be transmitted depends on the size of head of queue packet. As would be observed in our illustration in section three, using the weights to distribute resources in the WRR scheduler often leads to performance degradation in the face of existing resources and backlogged flows. This problem is encountered when a traffic class is prevented from transmission due to the fact that the packet rate at the head-of-queue exceeds the queue fair share. Here, a queue fair share of resources is defined as the proportion of available resource, calculated using weight that is reserved to a queue in WRR or other multi-queue handling scheduling environments. It is sometimes called the quantum in DRR schedulers [17, 23]. A queue is denied its share if it wants to take more than its fair share. In this paper, distinction is made from unused allocation caused when some service queues become empty and the one caused by a backlogged queue who is denied of transmission because of the burst input rate of its top packet. Some versions of WRR re-allocate resource before the commencement of next round to take care of the former unused space. The latter unused space is noticed after the round scheduling has started which is usually counted as a menial waste by the WRR and is neglected. The accumulation of such spaces may amount to something especially in the Internet where various packets of different sizes are multiplexed into a service queue. Thus, in the face of backlogged queue and menial size of resource, WRR's performance may not be optimum. Our model describes what can be done to avert this wastage.

This paper gives a report of observation of traffic scheduling with respect to WRR policy and discusses a Java simulation of a dynamic scheme that goes on scheduling from other queues until there is no resource wastage in a particular round.

The paper is broken into six sections. The next section discusses the efficacy of a scheduler in a Diffserv network and some popular schemes existing in literature. Section three analyses the logic of WRR while section four describes a new approach that we are envisaging to reduce resource

wastage inherent in WRR. In section five, Java simulation of the proposed dynamic WRR scheduling policy is presented with its performance improvement. In section six, the conclusion is drawn.

2.0 Related Work

In this section, we explore some important schemes that have been proposed for Qos networks especially the one related to our new model. Their efficiency is assessed by considering one or combination of fairness, bounded delay and computational complexity. We assume that as packets arrive, they are classified into different queues as usually done by diffserv routers; and the queues are attended to in a particular order. Generalized Processor Sharing is a generic scheduler that is fair with low complexity O(1) and a guarantee on packet delay. GPS [20] maintains a queue for each flow emanating from a source end and serves them bit by bit in a round-robin fashion. The scheme has a worst-case delay bound of L_{max}/r with O(1) complexity as proved in [28]. L_{max} is the maximum packet size in the network and r is the total link bandwidth. Though GPS is fair to all flows, it is difficult to implement in the real world where unit of transmission is a packet. Researchers usually use GPS as a reference scheduler to assess the fairness and delay bounds of their schedulers. Some time-stamped schedulers tries to achieve GPS delay bound but at higher complexity. In such schedulers, packets are first sorted into order of calculated finish times [8, 26]. WFQ is a popular example. A packet-based WFQ approximates GPS by servicing queues based on their earliest finished times. Thus WFQ could achieve GPS-relative delay bound of O(1) but at O(NLog N) complexity. WRR is a fairness-oriented scheme that does not use packet arrival time but reserves a portion of bandwidth to every queue that is backlogged in a round. Thus, WRR has no sorting overhead while serving the queues and as result, it is noted for O(1) complexity. The next section illustrates the logic used by WRR. Deficit Weighted Round Robin was derived from WRR to take care of the variable nature of input traffic which WRR cannot handle properly. Both WRR and DWRR are confronted with the problem described in section one. While WRR was designed for fixed packet size network, DWRR can tolerate both fixed and variable packet types. Two problems are paramount with WRR: i) it misbehaves during bursty input rate and ii) its packet may experience high delay depending number of flows that are backlogged at the service time.

From our survey of some QOs-based WRR schedulers, we have come across some proposals that describe how to dynamically adjust the service weight so that burstiness of some queues can be accommodated. In [11, 30], the weight adjustment is generally based on observed queue length. The authors in [13, 14] are more interested in adjusting the weight of delay-constrained queues. How to minimize delay jitter usually caused by WRR was the interest of the authors in [29]. In [18], the proposed class-based scheduler centred on how the excess bandwidth caused when some queues are empty can be borrowed by queues of the same priority.

One article that also addressed the problem of unused transmission slot was [9] while proposing MWDRR scheme for Passive Optical Network switching devices. The authors accumulated the current deficit counters at the end of a round and send pending packets in order of queues whose

head-of-queue is larger than the summed deficit counter. The scheme then reset deficit counters. Implementing MWDRR would amount to destroying the quality of DRR to keep deficit such that queues denied of their quota may be able to send in the next round. Since, it made sure that the sum of deficits is used up in the same round. Our own model is straight forward to implement and does not alter the normal sequence of servicing the queue. The next section describes how WRR algorithm works.

3.0 Logic of WRR Scheduler

Weighted Round Robin, simply WRR is a family of round-robin schedulers noted with a complexity as low as O(1) [21]. The scheme seeks after fair allocation of available resources among classes of competing flows. In WRR, there is a pre-assigned variable per queue called the *weight*, that specifies the fair share of the resources due to each class. Various proposals have used different values to indicate a weight. In [13,], the weight used is an arbitrary scalar quantity representing the number of cells that can be transmitted when the queue is visited. As used in [11, 12, 29], the weight relates to the percentage of link bandwidth distributed to a service class. In wireless system that uses the WRR technique [19], the weight represents the fractional time, called time-frame that a down link may upload or download. Once the weight is set, and queues are backlogged, a circular scan is made to all the queues in a round.

In WRR scheme, active queues are serviced in round robin fashion and the number of packets to schedule depends on the service class's weight. A queue is allowed to send packet if the length of the packet at the top of the queue is not greater than the bandwidth reserved for that queue. The scheduling goes on until there is no backlog queue, and in that case, the weight may be reviewed or left the same for the next scheduling run. WRR misbehaves when variable-size packets are in some queues. In such a circumstance, a queue is denied of transmission if its head of packet size is greater than its guaranteed rate. This may persist for rounds until there is a weight reset.

Let us have a walkthrough of a WRR scenario shown in Fig. 1. In the figure, the status of the queues is given after 3 successive time intervals, where it is assumed that no further packets arrive during the period under consideration. There are 4 queues designated Q0, Q1, Q2 and Q4, having bandwidth share 800, 600, 400 and 200bytes respectively and being weighted as 40%, 30% 20% and 10% respectively with respect to the output channel's 2000 bytes/sec capacity. 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 and 2 packets constitute 600 bytes, while 3 packets constituting 900 bytes would be forbidden. For similar reasons, 2 packets are removed from Q1 to the output queue, as well as 1 packet from Q2 and 0 packets from Q3. Q3 packet was not scheduled because its first packet size of 300 bytes is more than its share of 200bytes. This means that a total of 1500 bytes have to be transmitted on the 2000 bytes per second output channel with 500bytes unused after the first round. In the second round, the selection of packets from Q0, Q1 and Q2 is repeated as before, except that now there is only 1 packet in Q1 to select. The same penalty applies to Q3. At the end of round 2, throughput becomes 1200bytes, and a waste of 800bytes. Finally, in the third round, only 1 packet is available for selection from queues Q0 and Q2 respectively making a 600 bytes throughput.

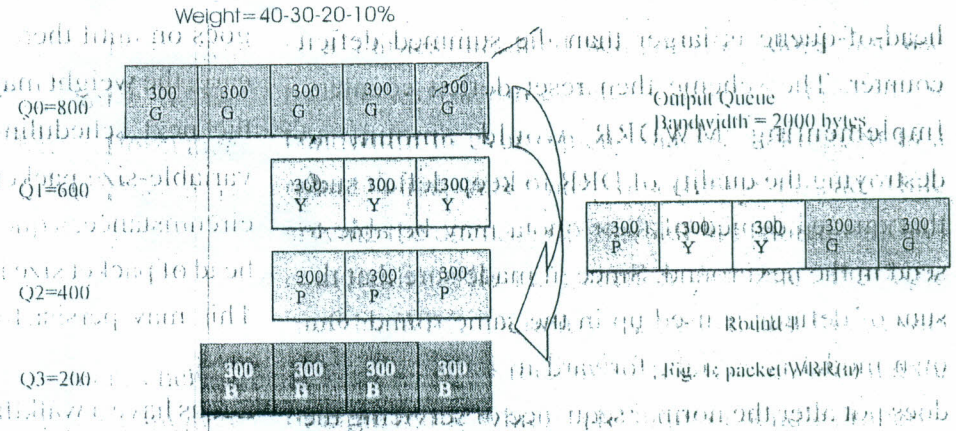

Fig. 1c packet WRR(a)

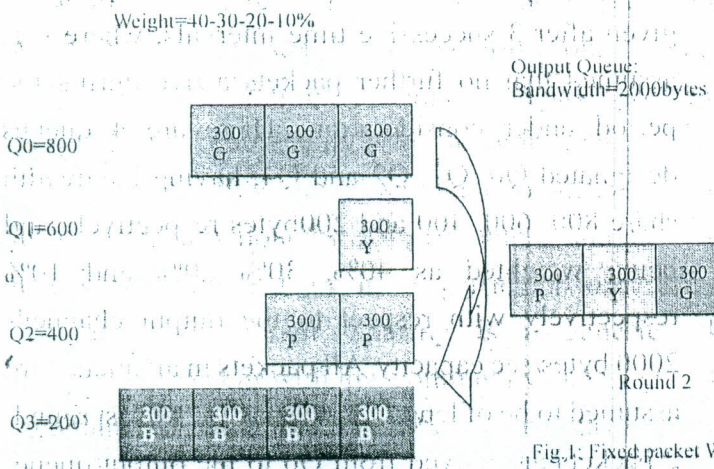

Fig.1: Fixed packet WRR(b)

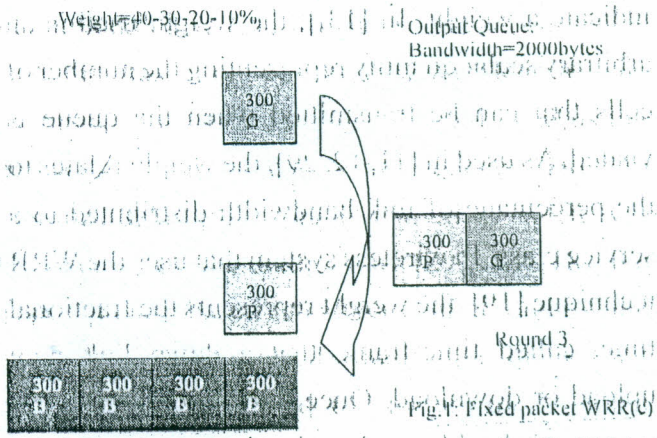

Fig.1: Fixed packet WRR(c)

When all the other queues are empty, Q3 packet may be given permission to use the entire channel's bandwidth since it does affect the fairness or delay index of other competing queues. Its penalty may persist until there is a weight reset.

WRR can operate in a static or dynamic manner in terms of weight changes or remaining unused resource. We are more interested in the latter in this article; the former shall be looked into in our subsequent articles. There are two ways a WRR can cause bandwidth wastage in expense of full backlogged flows. The first one occurs when at

backlogged queue becomes empty. In such a case, a dynamic WRR would adjust the bandwidth allocations while computing the fair share for the next round. The second means of bandwidth wastage, although minimal, may occur during a service round. In a service round, the maximum a queue can transmit is upper-bounded by the amount of resource that has been reserved for it. As such, no queue can consume more service than what has been assigned. This is the usual algorithm of WRR stated in [23,24]. It is our contribution in this paper that the unused resource created during a round is allowed to be used up by higher-bandwidth classes in the same round without affecting the normal service share of lower classes. The next section describes our model. Our approach seeks after near-100% utilization of the available resources in every round.

4.0 cDWRR Analysis

In this section, we describe the revised model, called carry-on WRR (cWRR). The revision introduced into the existing WRR algorithm is that, after servicing all the active queues, the scheduler checks if the total transmitted bytes are less than the indicated channel's bandwidth. If there is excess bandwidth, it goes over the round again operating as normal (as per the rules for WRR) to pick packets for dispatch to the output channel from the backlogged queues, starting with the first. Each extra picking is noted as bonus and abbreviated as b in the result tables presented in the next section. In each case, the scheduler keeps watch of whether the bandwidth of the outgoing channel has reached maximum capacity or whether more packets may be added to it.

A service queue in cWRR is considered active during a round if it has packets awaiting service. An active list is maintained to hold the index of all active service classes (ActiveList) and an array variable is also used to hold the number of packets to be picked in case a service class is authorized to send more than one packets. As packets arrive to each service class, 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. To assess the performance of cWRR, it is noted that the performance features of WRR are still maintained in cWRR. The computation of the queue fair share occurs at the beginning of the round, it is done for the number of queues and it is not for all 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 complexity O(1). The initiative to transmit more packets from higher-bandwidth classes may reduce the delay bounds of some packets and improve the strength of WRR scheme if this higher-bandwidth class is one of the delay-constrained classes. Thus, cWRR may be used where higher-bandwidth class consists of delay-sensitive traffic classes.

5.0 Simulation of the carry-on WRR

In this simulation just as in Fig 1.1above, four service classes are maintained and the simulation commences by generating a random number of packets that arrive at each queue (Poisson in nature). The weight per queue is fixed as 40-30-20-10% or simply 4-3-2-1. The simulation was carried out for the cWRR and WRR for comparative purposes. We presented the result of

two simulations for illustrative purpose. Table 1 shows the case where arrived packets are of the same size; while variable packet sizes are recorded in Table 3. The input pattern shown in the simulation were analysed in the case that no review has taken place to the original WRR. The implementation tool is Java Distribution Kit 1.5.0_10. In order to implement the WRR and cWRR algorithms in Java, a specified number of packets are generated into an array and are randomly mapped into four queues. Java graphics and colors facilities are used to represent a packet. A set of events buttons can be chosen to set up the simulation parameters, such as: the queue weights; the link bandwidth in bytes; the packet type (fixed or variable) and the number of packet to be generated. Once these are specified, a click on *Start button* executes the algorithm and graphically displays the progress of the queues. The system clock ticks are noted as the simulation commences to indicate the relative time taken by each of the algorithms. In each round, the number of bytes transmitted in each class (throughput), as well as the time taken to schedule in a round was recorded. The tables below show the results from the simulation. In Table 1, the records of implementing scenario when packet sizes are fixed and varied is presented. The output queue from the simulation is translated using letters in the output row in the table. For example, when a blue packet from Q3 is scheduled for transmission, in output row in the WRR and table, it is represented as a B. In this same way, G stands for Green packet, P for pink packet and Y for yellow packet. From the output queue row, we derive the amount of bytes transmitted in each round from the queues using packet size indicated column 5 (Packet Size). Also in the output row, a slash(/) demarcates the end of a round. A "b" in the round columns indicates bonus that a particular queue is allowed to send because of the opportunity offered by the carry-on scheduler (cWRR).

From table 1, the lower part is the observed records of cWRR. Here, Q0 sends 3 packets in round 1 (2 packets as a result of quantum and 1 packet bonus as a result of bandwidth leftovers). Q1 and Q2 received their normal share by sending 2 and 1 packet respectively. Q3 is denied because its quantum does not afford it. This makes a 20% increase in throughput (cWRR=1800, WRR=1500 as extracted to Table 2 (%Thruput Increase column). In Round 2, Q2 gains a bonus of 300 bytes (1 packet transmission) because Q0, Q1 after being served based on their normal quantum become empty and Q2 is the next active queue. Q3 also benefited because it became the only active queue because there is still excess bandwidth. The summary of the throughput difference is shown in Table 2 and depicted in Fig. 2.

In the 1th session two, variable packet sizes arrived into the queues and result is analyzed as seen in the lower part of Table 2. The wastage column represents the difference between the given channel's bandwidth and the throughput after a round of scheduling. We also derive the percentage wastage by taking the difference over the bandwidth provided and multiply by 100. From the same Table 2, we compute the percentage increase in throughput between the two scheme by subtracting the throughput obtained in WRR in a round from the one obtained in cWRR, the result is taken over WRR and multiplied by 100.

From our observations in Table 2, there were cases of 40% increase in throughput under the %Thruput column. This shows that cWRR proves better than WRR. This is best viewed in Fig. 2. Fig 3 also shows

that cWRR outperform WRR even 100% in some cases. We also observed a reduction in resource wastage when using cWRR compared to WRR. On average, there was a drop from 80% to 12% when variable packets were used to test the simulatio. This is shown in the average row in column 2.

In Fig 3, the weakness of WRR when it meets queues with variable packets is well noticed. It takes WRR eight rounds to service while cWRR because of its permission to allow more transmission consumes four rounds. Thus, we can say based on the evidence above, that using our proposed and designed cWRR would go along way for a better service differentiation and better maximum resource usage when compared to models that employed ordinary WRR.

Table 1:Run session Scheduling Scenario: Fixed Packet, Weight:4-3-2-1; Bandwidth=2000; Number of packet=15

	Class Colour	Class fair share	No. of Arrived Packet	Packet Size(bytes)	Round 1 O/p (bytes)	Round 2 O/p (bytes)	Round 3 O/p (bytes)	Round 4 O/p (bytes)	Time Spent
WRR	Q0(Green)	800	5	300	600	600	300	0	75 secs
	Q1(Yellow)	600	3	300	600	300	0	0	
	Q2(Pink)	400	3	300	300	300	300	0	
	Q3(Blue)	200	4	300	0/23secs	0/19secs	0/11 secs	1200/22secs	
	O/p Queue		GGYYP/GGYP/GP/BBBB						
cWRR	Q0(Green)	800	5	300	600+300b	600	0		61 secs
	Q1(Yellow)	600	3	300	600	300	0		
	Q2(Pink)	400	3	300	300	300+300b	0		
	Q3(Blue)	200	4	300	0/29secs	0+300b/21secs	900/11 secs		
	O/p Queue		GGYYPG/GGYPPB/BBB						

Table 2: Summary of Results Obtained from two Simulation Runs

Round	WRR Thruput	WRR Wastage	WRR %Wastage	cWRR Thrupt	cWRR Wastage	cWRR %Wastage	%Thruput Increase
1	1500	500	25	1800	200	10	20
2	1200	800	40	1800	200	10	50
3	600	1400	70	900	Empty		50
4	1200	Empty					40
Average			45			10	40

SESSION 2: Variable Packet Size, Bandwidth 2000 bytes, 15 Packets Generated, W=4-3-2-1

	WRR			cWRR			
1	900	1100	55	1800	200	10	100
2	300	1700	85	1700	300	15	466.67
3	300	1700	85	400	Empty		
4	300	1700	85				
5	300	1700	85				
6	300	1700	85				
7	300	1700	85				
8	1200	Empty					
Average			80.71			12.5	283.33

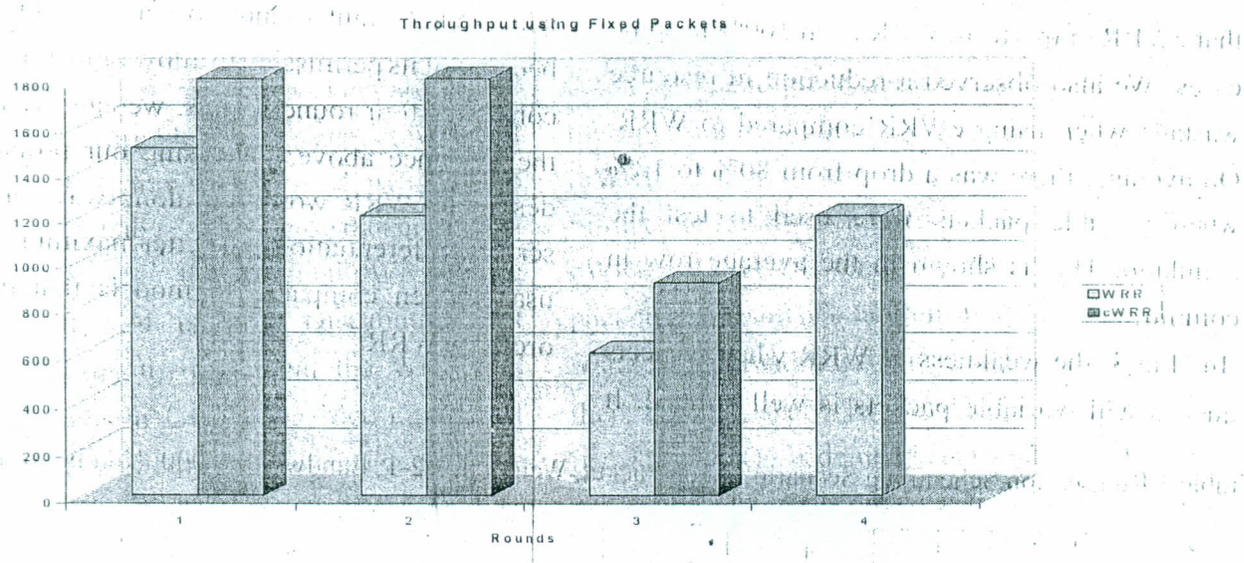

Fig 1.2: Throughput using Fixed Packet size(in bytes)

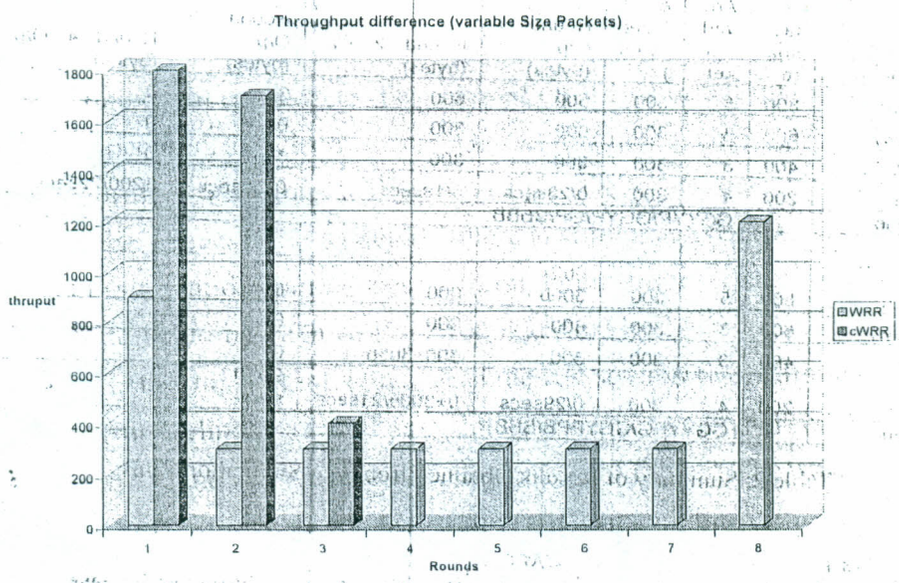

Fig 1.3: Throughput using Variable Packet size(in bytes)

6.0 CONCLUSION

In most of the specifications and designs for achieving differentiated services in the Internet, priority weighted round robin and deficit weighted round robin schemes are largely considered to be the core router scheduling basically because of their implementation simplicity at both hardware and software level. A statically configured packet-based WRR scheduler may not work efficiently where the queue packets are not of the same size, and is even worsened when there is input rate burstiness. The approach to solve these problems led to introduction of DWRR. DWRR is found useful in supporting both responsive and unresponsive flow in a work-conserving router. This paper confirms the fact that both WRR and DWRR may cause performance degradation because of their static nature and inflexible rules of

suspending transmission from backlogged queues if its fair share is less than the head of queue packet length. We designed and tested an adjusted scheme called carry-on WRR and carry-on DWRR. Our simulated results show better throughput performance as well as a reduction of resource wastage compared to normal WRR and DWRR.

As a first step in our future work, the simulation scenarios tested above will be expanded. Simulations based on open source simulation tool, ns will be developed, to examine the impact of cWRR and cDWRR across a much wider spectrum of scenarios. Our future work would demand that the weight which determines the fair distribution of available resources be dynamically regulated based on traffic statistics along the route.

REFERENCES

1 Arunabha S., Ibraz M., Ravikanth S. and Subir B. "*Fair Queuing with Round Robin: A new Packet Scheduling Algorithm for Routers*", ISCC '02, 2002.

2 Blake S. Black D. Carlson M, Wang Z. and Weiss W. "*An Architecture for Differentiated Services*", Dec. 1998, IETF RFC 2474

3 Bogdan C., Jason N. and Wong C. "*Group Round Robin: Improving the Fairness and Complexity of Packet Scheduling*", ANCS '05, Oct 2005, Princeton, New Jersey, USA, pp. 29-38

4 Braden R. Clark D. and Shenker S. (1994) "*Integrated Services in the Internet Architecture: an Overview*", Jul 1994, RFC 1633

5 Braden R., Zhang L. Berson S. and Jamin S. "*Resource ReSerVation Protocol (RSVP) Version 1, Functional Specification*", Sept. 1997, RFC 2205

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

7 Cuthbert L.G. and Sapanel J.C. "*ATM: The broadband Telecommunication Solution*, IEE. London 1993

8 Demers A., Keshav S. and Shenkar S. "*Analysis and Simulation of a Fair Queuing Algorithm*. Proc. ACM SIGCOMM 1989

9 Dong-yeal L. and Sung-hyeub O. " A new DBA Scheme to Improve Bandwidth Utilization in EPONs", ICACT2006, Feb 20-22, 2006, pp. 1063-1067

10 Floyd S., Jacobson V. "*Random Early Detection gateways for Congestion Avoidance*", IEEE/ACM Transactions on Networking, 1(4) Aug, 1993

11 Heng-Yi W., Min-kuan C. and Chia-Cung C. "*The Switch-Based Subcarrier Allocation Policies In Multi-Service OFDM Systems*", IEEE 2006 pp.1328-1332

12 Hideyuki S., Makiko Y. Ruixne F., and Hiroshi S. "*An Improvement of WRR cell in scheduling in ATM Networks*", IEEE 1997 pp. 1119-1123

13 Hyun-Ho Y., Hakyong K., Changhwan O. and Kiseon K. "*A Queue Length-based Scheduling Scheme in ATM Networks*", IEEE 1999, pp. 234-237

14 Kim M. and Park H. "Scheduling self-similar traffic in packet-switching systems with high utilization", IEE Proc. Commn. Vol. 151(5) Oct 2004

15 Liebeherr J. and Christin N. "*Buffer Management and Scheduling for Enhanced DiffServ*", Technical Report, University of Virgina, Aug. 2000

16 Luciano L., Enzo M. and Giovanni S. "*Tradeoffs Between Low Complexity, Low Latency, And Fairness With Deficit Round Robin Schedulers*", IEEE/ACM on Networking, Vol. 12(4), August 2004

17 MacGregor M. and Shi W. "*Deficit for Bursty-critical flows: DRR++*", IEEE 2000

18 Manais G., Markaki G., Politis G. and Venieris I. "Efficient Buffer Management and Scheduling in a Combined IntServ and DiffServ Architecture: A Performance Study", IEEE 1999, pp. 236-

19 Mezger K. and Petr D. "Bounded Delay for WRR", Technical Report, Dept. of Electrical Engineering and Computer Sciences, University of Kansas 1995

20 Parekh A. and Gallager R. "A generalized processor sharing approach to flow control in integrated services network: The single node", IEEE/ACM Trans Networking, Vol. 1(3) June 1993, pp 344-357

21 Peng-Kong K., Kee-Chaing C. and Brahim B. "*Multicode-DRR: A Packet-Scheduling Algorithm for Delay Guaranteed in Multicode-CDMA Network*", IEEE Trans. On Wireless Communications Vol. 4(6) Nov. 2005

22 Salil S. and Harish S. " Fair, Efficient and Low-Latency Packet Scheduling Using Nested Deficit Round Robin", IEEE 2002 pp.6-10

23 Semeria C. "*Supporting Differentiated Service Classes: Queue Scheduling Disciplines*", White paper, Juniper Networks Inc., 2001

24 Shreedar M. and Varghese G. "*Efficient Far Queuing using Deficit Round Robin*", IEEE/ACM Transactions on Networking, vol. 4(3), Jun 1996

25 Sriram R. and Joseph P. "*Stratified Round Robin: A Low Complexity Packet Scheduler with Bandwidth Fairness and Bounded Delay*", Proc. Of ACM SIGCOMM '03 Karlsruhe Germany, Aug 2003

26 Sungwon Ha, Kang-Won Lee and Vaduvur Bharghavan: "*Performance Evaluation of Scheduling Algorithms in an Integrated Packet Services Network Environment*", IEEE 2000

27 Stiliadis D., Varma A. "*Efficient Fair Queuing Algorithms for Packet Switched Networks*", IEEE/ACM Transaction on

Networking vol. 6 April, 1998

28 Xu J. And Lipton J. "On fundamental Tradeoffs between Delay Bounds and Computational Complexity in Packet Scheduling Algorithms", ACM Proc.of SIGGCOM '02, Pittsburgh, USA, Aug. 19-23 2002

29 Yao-Tzun W. Tzung-Pao L. and Kuo-Chang G. "An improved Scheduling Algorithm for Weighted Round Robin Cell Multiplexing in an ATM Switch", IEEE 1994 pp. 1032-1036

30 Yoshihiro H., Shuji T. and Yutaka I. "*Variably Weighted Round Robin Queuing for Core IP Routers*", IEEE 2002 pp. 159-166

31 Yunkai Zhou and Harish Sethu " *On Achieving Fairness in the Joint Allocation of Processing and Bandwidth Resources*", International Workshop on Quality of Service, June 2-4 2003, Monterey,CA, USA.

32 Zhang S., and Chen-Ing Y. " *On Service Differentiation in Mobile Adhoc Networks*", Journal of Zhejiang University Science 2004 5(9) pp. 1087-1094