

DESIGN AND IMPLEMENTATION OF LOW LATENCY WEIGHTED ROUND ROBIN (LL-WRR) SCHEDULING FOR HIGH SPEED NETWORKS

Zuber Patel and Upena Dalal

Department of Electronics Engg., National Institute of Technology, Surat, India

ABSTRACT

Today's wireless broadband networks are required to provide QoS guarantee as well as fairness to different kinds of traffic. Recent wireless standards (such as LTE and WiMAX) have special provisions at MAC layer for differentiating and scheduling data traffic for achieving QoS. The main focus of this paper is concerned with high speed packet queuing/scheduling at central node such as base station (BS) or router to handle network traffic. This paper proposes novel packet queuing scheme termed as Low Latency Weighted Round Robin (LL-WRR) which is simple and effective amendment to weighted round robin (WRR) for achieving low latency and improved fairness. Proposed LL-WRR queue scheduling scheme is implemented in NS-2 considering IEEE 802.16 network [1] with real time video and Constant Bit Rate (CBR) audio traffic connections. Simulation results show improvement obtained in latency and fairness using LL-WRR. The proposed scheme introduces extra complexity of computing coefficient but its overall impact is very small.

KEYWORDS

Scheduling, WRR, LL-WRR, fairness, latency

1. INTRODUCTION

The phenomenal growth in real time services such as interactive voice & video poses challenge in meeting end to end QoS [2] requirement. Unlike non-real time data services, these real time applications have stringent performance requirements. Keeping this in mind, many schemes have been proposed by researchers for efficient packet queuing and scheduling. The primary job of queuing and scheduling is to treat different traffic classes with variable degree of priority to provide performance guarantee for range of different traffic types and profiles. They determine the order in which packets from different service classes are processed and served, hence it dictates resource allocation to different connections.

The scenario considered here consists of packet queues of various connections (or sessions) waiting for transmission through a single output port of network node. Scheduler component of network node schedules packets based on some policy so as to achieve requirements of each connection such as minimum reserved transmission rate (MRTR), latency, jitter and fairness. It is desirable to have low complexity in the implementation of scheduler to provide QoS in high speed converged networks. A queue scheduling scheme may not possess all of the above desirable scheduling properties instead offers subset of them. For example, weighted fair queuing

(WFQ) [3] and its variant worst-case fair weighted fair queuing (WF2Q) [4] are having good delay and fairness properties but have high implementation complexity. Self clocked fair queuing (SCFQ) [5] uses the virtual finish time of the packet that is currently being transmitted as the system virtual time. As a result, the complexity of computing the system virtual time [6] of SCFQ is $O(1)$ but delay increases linearly with increase in number of sessions. RPS based schemes (such as FFQ [7], SPFQ and MD-SCFQ) offer low complexity at the expense of degradation in fairness. Many schemes such as WFQ and SPFQ fail to provide stable latencies for real time traffic. This issue is addressed in [8] to guarantee low and stable latency for real time flows.

This paper proposes modification to WRR named as LL-WRR that improves *latency* and *fairness* of real time services with very small impact on complexity and data rate. LL-WRR is a simple and lightweight method of controlling *Round Robin length* by changing integer weights of connections while keeping the simplicity of WRR. The Round Robin length signifies the sum of packets to be scheduled from all queues in single round. To obtain integer weight, fractional weight of each connection is multiplied by constant integer in classical WRR. Thus, the sum of these integer weights i.e. Round Robin length is fixed and does not scale well with variation in number of connections. In LL-WRR scheme, instead of multiplying constant integer, a *co-efficient* γ is multiplied to fractional weights. The coefficient γ is function of number of connections present in network and it is made to decrease as number of connections increases. This eventually reduces Round Robin length and latency remains low. The co-efficient γ (hence Round Robin length) is computed only at the beginning of each WRR cycle rather than every packet arrival or departure. This keeps complexity of LL-WRR low.

The rest of the paper is organized as follows. Brief idea of conventional WRR, list based interleaved WRR and Multiclass WRR is reviewed in Section 2. Section 3 presents proposed LL-WRR scheme with the analysis on latency, rate and fairness properties. Section 4 discusses simulation results obtained in NS-2. Finally, the conclusion remarks are given in Section 5.

2. RELATED WORK

Generally speaking, packet scheduling algorithms can be divided into two categories (1) Timestamp based (2) Round Robin based. Timestamp-based algorithms have provably good delay and fairness properties, but generally need to sort packet deadlines, and therefore suffer from complexity logarithmic in the number of flows N . Generalized Processor Sharing (GPS) [3] (also called Fluid Fair Queuing) is considered the ideal time stamped scheduling discipline that achieves perfect fairness and isolation among competing flows. However, the fluid model assumed by GPS is not amenable to a practical implementation. However, GPS acts as a reference for other scheduling disciplines in terms of delay and fairness. Practical timestamp schedulers try to emulate the operation of GPS by computing a timestamp for each packet. Packets are transmitted according to their timestamps. WFQ and WF2Q are examples of time stamped schedulers. WFQ is packet-by-packet equivalent of GPS. WFQ exhibits some short-term unfairness which is addressed by the WF2Q.

Round-robin-based algorithms achieve $O(1)$ complexity by eliminating time stamping and sorting. The simplicity of these algorithms can be useful for traffic scheduling in very high speed networks. They support fair allocation of bandwidth, but unable to provide good delay bounds. Most basic round robin type scheduler for differentiated services network is WRR. WRR assures fraction of output link bandwidth to each service queue by assigning appropriate weight. The deficit round robin (DRR) is modification of WRR which takes into account packet size for scheduling. In following sections, we shall discuss and analyze few WRR based methods namely conventional WRR, List based interleaved WRR and Multiclass WRR.

2.1. CONVENTIONAL WRR

The WRR is simple Round Robin based scheduling algorithm used in packet-switched networks with static weight assigned to connections' queues. It cycles through queues transmitting amount of packets from each queue as per its weight (Figure 1) thus guaranteeing each connection a fraction of output link bandwidth. It also ensures that lower priority queues never starved for long time for buffer space and output link bandwidth. It has processing complexity of $O(1)$ which make it feasible to high speed interfaces in both core and at the edge of network. The primary limitation of WRR queuing is that it provides correct percentage of bandwidth to each service class only if all of the packets in all queues are of same size or when mean packet size is known in advance.

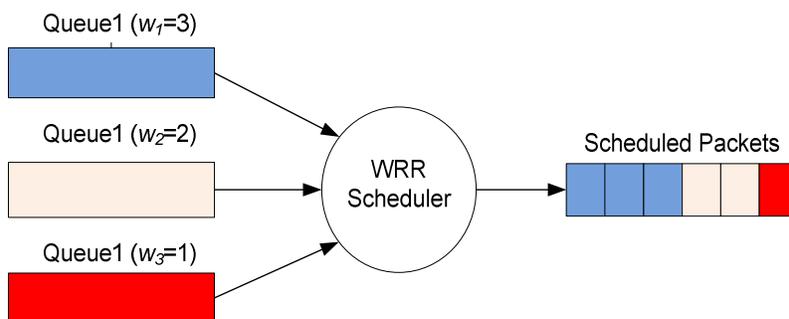


Figure 1. Operation of Conventional WRR

WRR scheduling is based on assigning fraction weight ϕ_i to each service queue such that sum of weight of all service queues is equal to one.

$$\sum_{i=1}^N \phi_i = 1 \quad (1)$$

Since weight is *fraction* and we want to determine number of integer packets to be served from each queue, the fraction weight is multiplied by proper constant integer M . The product is rounded off to nearest larger integer to obtain *integer weight* w_i . This integer weight value of each queue specifies number of packets to be serviced from that queue. The total sum of these counter values is referred to as round robin length. The integer weight of i^{th} queue is

$$w_i = \lceil \phi_i * M \rceil \quad (2)$$

The sum of existing N active connections is defined as round robin length W and is given by

$$W = \sum_{i=1}^N w_i = M \quad (3)$$

Assume that the rate of outgoing link is r , and the rate offered to i^{th} connection is

$$r_i = \frac{w_i}{W} r \quad (4)$$

Let us understand the effect of increasing number of connections on the rate. As number of connection N increases, the equality of Equation (1) tells that individual weight of connection ϕ_i decreases and this reduces w_i . Since sum of all weight remains constant, W remains unchanged and hence as per Equation (4) rate of that connection decreases.

The latency θ_i of any connection i defined in [9] and is adopted in our analysis. For particular scheduling algorithm, parameters such as transmission rate of output link, allocated rates and number of connections may influence latency. We determine worse-case latency for connection i for conventional WRR scheduler. Assume that there are N connection queues being backlogged and scheduler is currently serving w_i^{th} packet from i^{th} connection. Since cycle length is W , there could be as many as $W-w_i$ packets to be served from other $N-1$ queues before $(w_i+1)^{\text{th}}$ packet from connection queue i is served. Therefore worst-case latency for i^{th} connection is

$$\theta_{i,WRR} = (W - w_i) \frac{L_i}{r} = W(1 - \phi_i) \frac{L_i}{r} \quad (5)$$

where L_i is the maximum length of packet that belongs to i^{th} connection. This worst-case latency increases as ϕ_i decreases with increase in number of connections. Hence it has inefficient latency tuning characteristics. To compute *total latency* experienced by a packet, queuing latency should be added to Equation (5).

The proportional fairness $\eta_{PF}=1$ since a connections i (j) can lead the other connection j (i) at the most by w_i (w_j) packets. To measure *worst-case fairness*, a metric called Worst case Fair Index (WFI) is defined in [4] to characterize fair queuing servers. A server is said to guarantee a WFI of C_i for connection i , if for any time t the delay of a packet arriving at t is bounded

$$d_i < a_i + \frac{Q_i(t)}{r_i} + C_i \quad (6)$$

where $Q_i(t)$ is the queue size of connection i at the packet arrival time t and C_i is called worst case fair index for connection i . Suppose a new packet of connection i arrives at time t when the server has just crossed i , and suppose the backlog of connection i at time t (denoted by $Q_i(t)$) is multiple of w_i . Then, this packet departs after a maximum of time of $\lceil Q_i/r_i + (W + w_i + 1)L_i/r \rceil$. Thus WFI of WRR scheduler is given by

$$\begin{aligned} C_{i,WRR} &= d_i - a_i - \frac{Q_i(t)}{r_i} \\ &= (W - w_i + 1) \frac{L_i}{r} \end{aligned} \quad (7)$$

As the number of connections increases, w_i decreases and hence WFI increases. So increase in number of connections on scheduler degrades WFI.

2.2. LIST BASED INTERLEAVED WRR

In list based WRR scheme [10], instead of serving w_i packets from i^{th} connection in single visit, the service is distributed evenly over the entire Round Robin cycle. Scheduler visits queues of connections as per the “*service list*” maintained by it. The number of times connection i appears in the service list is proportional to its weight w_i , but these appearance are not necessarily consecutive as in conventional WRR. The service list is updated only at the time of new connection establishment or connection termination.

In order to form service list, we create $M (= \max_{i=1 \text{ to } N} (w_i))$ slots in service list and each slot contains entries of indices of connections. A connection i will have w_i entries in service list evenly distributed across all slots. The total length of service list is $W = \sum_{i=1}^N w_i$ is called Round Robin length. Scheduler parses this service list and determines queues to be serviced.

Consider any two connections i and j with $w_j \geq w_i$. In list based WRR scheduling, the connection j can lead i by at the most $w_j - w_i + 1$ packets in any partial Round Robin cycle. Let us assume that $S_i(t, t+\tau)$ and $S_j(t, t+\tau)$ are services offered to connections i and j respectively by server during interval $(t, t+\tau)$. Then as per definition given by [5] the proportional fairness η_{PF} is the difference in the normalized services offered to i and j . Because of cyclic nature of scheduling, the maximum normalized service by which j can lead i , is the same as maximum normalized service by which i can lead j . In other words,

$$\left| \frac{S_i[t, t+\tau]}{w_i} - \frac{S_j[t, t+\tau]}{w_j} \right| \leq \frac{w_j - w_i + 1}{w_j} \quad (8)$$

Hence, for this list based scheme the proportional fairness is

$$\eta_{PF} = \frac{w_j - w_i + 1}{w_j} \quad (9)$$

Since the values of w_i and w_j are normally larger than 1, η_{PF} of this scheme is smaller than conventional WRR and hence it has better proportional fairness. Its latency value $(W - w_i) = W(1 - \phi_i)$ suggests that this scheme also lacks efficient latency tuning.

2.3. MULTICLASS WRR

This scheme offers scheduling properties similar to WFQ based schemes. Multiclass WRR [10] has efficient tuning characteristics and it is worst-case fair. To get the initial grasp of Multiclass WRR, consider M classes from \mathcal{C}_1 to \mathcal{C}_M containing N_1 to N_M connections respectively and all connections have unity weight. Also, let W_1 to W_M represents maximum length of Round Robin cycle of class \mathcal{C}_1 to \mathcal{C}_M respectively in increasing order of size. Multiclass WRR works on *minicycle* which is set to W_1 visits. The scheduler operates by embedding smaller Round Robin cycles within a minicycle. In every minicycle, all connections of class \mathcal{C}_1 are always visited. After this, the connections in subsequent classes are visited from the leftover visits from previous classes. That is to say, connections in any class \mathcal{C}_m are visited from leftover visits from classes $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{m-1}$ in a minicycle. If the fraction of the output link bandwidth assigned to connection i is ϕ_i then from equality $\sum_{i=1}^M \phi_i \leq 1$ following condition holds:

$$\sum_{m=1}^M \frac{N_m}{W_m} \leq 1 \quad (10)$$

The operation of Multiclass WRR is shown pictorially in Figure 2 Notice that the minicycle may or may not end at the boundary of class. Besides, it may require one or more visits of minicycle to serve all connections of any class after \mathcal{C}_1 . In the scenario of Figure 2, the first minicycle terminates after visiting the $(N_1 + \beta_1)^{th}$ connection. During the second round minicycle when the server crosses the class \mathcal{C}_1 boundary, it jumps to visit the $(N_1 + \beta_1 + 1)^{th}$ connection. The second minicycle ends at the end of \mathcal{C}_2 . The third minicycle ends when the last connection in class \mathcal{C}_2 is visited.

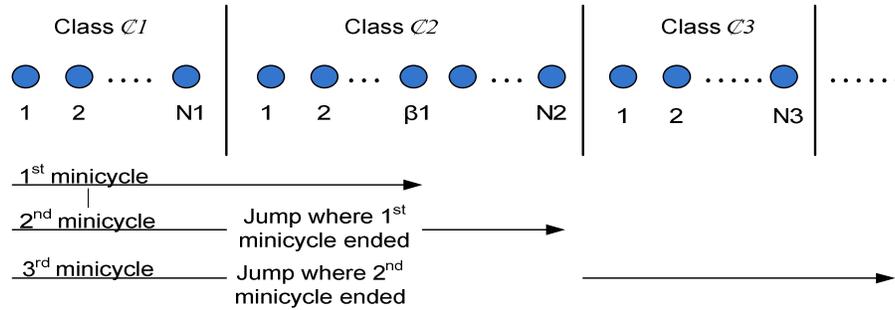


Figure 2. Operation of Multiclass WRR with minicycles

It can be shown that Multiclass WRR has near optimal proportional fairness over any interval during which both connections i and j are continuously back-logged. In addition, it has efficient latency tuning characteristics and it is worst-case fair.

Many other modifications to WRR have been suggested in research literature. Improvement in average delay and throughput with large service round is discussed in [11] whereas [12] suggests a *revenue based* criterion is to adjust weight to allocate resources in optimum way that maximizes total revenue. In other work [13], negative deficit weighted round robin (N-DWRR) is proposed which schedules packets based on credit, negative credit and packet size to improve the bandwidth utilization rate without increasing the total latency.

3. PROPOSED LOW LATENCY WRR (LL-WRR) SCHEME

Proposed LL-WRR scheme is simple extension to WRR which represents tradeoff between *rate* and *worst-case latency*. We first discuss the basic concept of proposed scheme. In real networks, packets of a connection experience queuing latency and scheduling (processing) latency. In section 2.1, we have seen that WRR scheduling latency increases with number of connections linearly. The queuing latency, however, increases nonlinearly with traffic load [14]. Assuming the same arrival rate of packets on each connection, the traffic load is directly proportional to the number of connections. Assuming M/M/1 queuing with Poisson's arrival at rate λ and service rate μ and traffic load on i^{th} queue is ρ_i , the average queuing latency T_i of i^{th} connection is given by little's formula

$$T_i = \frac{\rho_i}{\mu(1-\rho_i)} \tag{11}$$

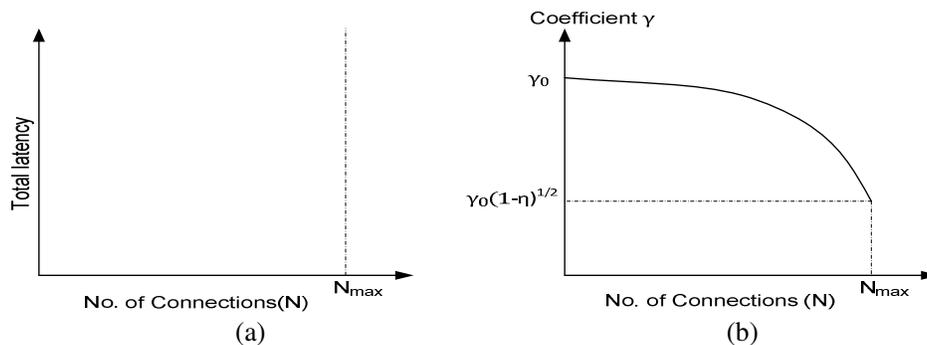


Figure 3. (a) Total latency variation (b) Variation of Coefficient γ of LL-WRR

The *total latency* varies as shown Figure 3(a) with number of connections N . It rises slowly initially and then increases rapidly as N becomes very large. In our proposal, to improve worst-case latency, a coefficient γ is multiplied to weight ϕ_i of each connection and is made to vary with N . Since latency is proportional to γ , we can keep latency low by decreasing γ with increase in N . To counteract rise in total latency (Figure 3(a)), we reduce co-efficient slowly initially for smaller value of N and then decrease rapidly for large value of N . In other words, co-efficient must vary as shown in Figure 3(b). The function that satisfy this characteristics is given as

$$\gamma = \gamma_0 \sqrt{1 - \eta(N/N_{\max})^2} \quad (12)$$

where γ_0 is appropriate integer constant that decides Round Robin length for a given number of connections N . The constant fraction η (having value less than 1) decides the value of γ when number of connections N reaches to maximum value i.e. N_{\max} . The integer weight of i^{th} connection w_i is obtained by ceiling the value of product $\gamma\phi_i$. The corresponding cycle length W is sum of w_i . The scheduler latency of our scheme is obtained by substituting these values of w_i and W in Equation (5) and then total latency $\theta_{i,LL-WRR}$ can be obtained by adding queuing latency of Equation (11) to scheduling latency.

$$\theta_{i,LL-WRR} = \left[\left(\sum_{i=1}^N \lceil \gamma\phi_i \rceil \right) - \lceil \gamma\phi_i \rceil + \frac{\rho_i}{\mu(1-\rho_i)} \right] \quad (13)$$

Similarly, rate of i^{th} connection is obtained from Equation (4) by substituting new values of w_i and W

$$r_i = \frac{\lceil \gamma\phi_i \rceil}{W} r \quad (14)$$

From Equation (13) we can surmise that as number of connections increases, the total latency tends to remain very low. This is because as number of connections increases, γ reduces rapidly and the term $\sum_{i=1}^N \lceil \gamma\phi_i \rceil$ reduces. Thus increase in $\theta_{i,LL-WRR}$ due to reduction in $\lceil \gamma\phi_i \rceil$ and increase in traffic load ρ_i is compensated by decrease in $\sum_{i=1}^N \lceil \gamma\phi_i \rceil$. The Equation (14) suggests that data rate performance is negligibly affected at low value of N . But as N increases to large value, even if W decreases, significant reduction in $\lceil \gamma\phi_i \rceil$ reduces ratio and hence the data rate r_i . This demonstrates the trade-off between latency and data rate. This scheme also improves *worst-case fairness*. From Equation (7), we may write WFI for LL-WRR as

$$C_i = (W - w_i + 1) \frac{L}{r} = \left(\sum_{i=1}^N \lceil \gamma\phi_i \rceil - \lceil \gamma\phi_i \rceil + 1 \right) \frac{L}{r} \quad (15)$$

As the value of γ decreases faster at large N , the term $W-w_i$ reduces and WFI index (C_i) decreases and hence improves worst case fairness. As far as complexity is concerned, our scheme slightly increases complexity of scheduler as update in γ is required but this update is done once in a Round Robin cycle. Thus it is possible to achieve low latency and improved fairness with little increase in complexity of implementation.

3.1. ARCHITECTURE OF LL-WRR

The overall LL-WRR scheduling architecture is shown in Figure 4 and algorithm is shown in pseudocode 1. Each connection's queue is assigned integer weight w_i whose value is obtained by multiplying its fractional weight ϕ_i with γ and then ceiling the product. Scheduler visits each connection one after the other and removes w_i packets. Once it serves all the queues i.e. when a Round Robin cycle is completed, it determines number of connections that are present and then updates value of γ . Based on updated value, integer weights of all connections are recalculated and process is repeated.

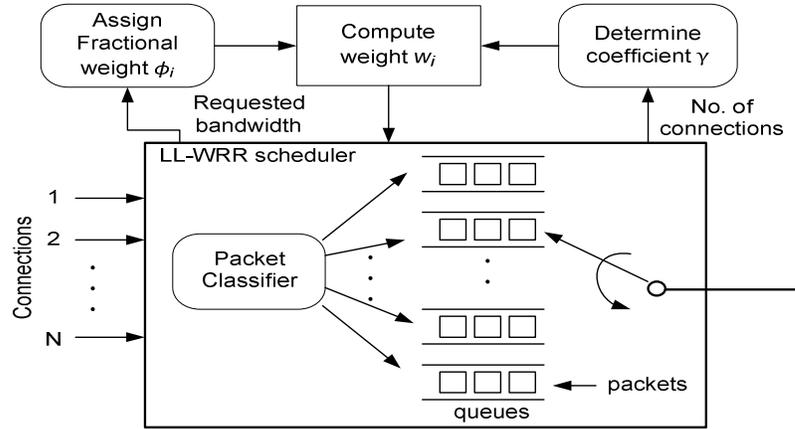


Figure 4. Architecture of LL-WRR scheduling

Pseudocode 1: Algorithmic steps of LL-WRR scheduling

Notations:

- N : Number of connections
- N_{max} : Maximum number of connections supported
- ϕ_i : fractional weight of connection i
- w_i : integer weight of connection i
- γ : coefficient
- W : sum of w_i or Round Robin length
- Q_i : size of queue of i^{th} connection
- $PktSize_i$: Size of packet for i^{th} connection

Initialization:

Assign fractional weight Φ_i to each service queue
 $N = 0$;

For all i do

$$w_i = \left\lceil \phi_i \left(\gamma_0 \sqrt{1 - \eta \left[\frac{N}{N_{max}} \right]^2} \right) \right\rceil$$

End for

Enqueuing:

$Q_i = Q_i + PktSize_i$;
If ($Q_i > Q_{i,max}$) drop the packet;

Dequeuing:

```

While (True)
  If (Nonempty queue exist) {
    For all  $i$  do
      if ( $w_i > 0$  and  $Q_i > 0$ )
        Transmit packet to output link from queue  $i$ ;
         $w_i = w_i - 1$ ;
         $Q_i = Q_i - PktSize_i$ ; /*update queue size*/
      End if
    End for /* round is over */
    Update  $N$  and  $\phi_i$ ;
    Compute  $\gamma$ ;
    For all  $i$  do
       $w_i = w_i * \gamma$ ;
    End for
  End if
End while

```

4. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed LL-WRR scheduling algorithm in the context of IEEE 802.16 MAC layer [1]. More specifically, the effect of number of connections (with varying subscriber stations) on the scheduling algorithms is studied. The LL-WRR algorithm is developed, configured and simulated in NS-2 [15]. Simulation script is written in OTcl for defining wireless network scenario with single 802.16 base station and multiple subscriber stations (SSs) where SSs are *mobile nodes* with average mobility of 5 m/s.

4.1. SIMULATION RESULTS

The objective of simulation experiments is to evaluate the performance of proposed LL-WRR algorithm and compare it with conventional WRR and deficit round robin (DRR) scheduling algorithms under CBR audio and MPEG4 video traffic. The experiments are conducted with these two types of traffic (flows) generated by each SS. Plots of *data rate*, *latency* and *fairness* under varying no. of connections (CBR plus video connections) are obtained for LL-WRR.

The data rate plot for each traffic type (Figure 5 and 6) is obtained as a function of number of connections N in network. As seen from plots, data rate performance WRR is slightly better than our scheme and DRR under moderate to high CBR traffic. For MPEG4 traffic, DRR achieves highest data rate whereas data rate of proposed LL-WRR is almost similar to WRR.

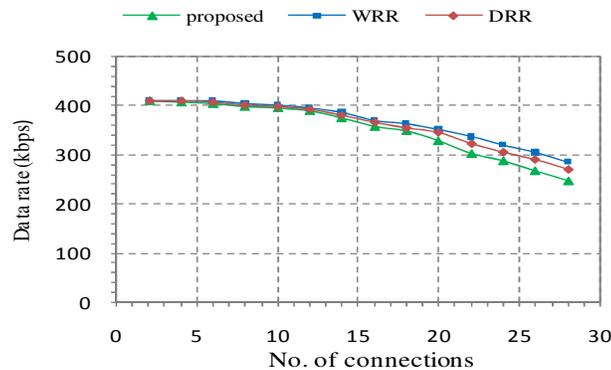


Figure 5. Data rate of CBR audio traffic.

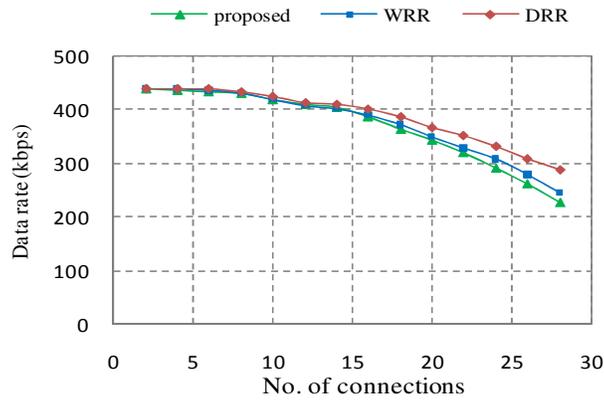


Figure 6. Data rate of MPEG4 video traffic

Proposed scheme exhibits much better latency characteristics than conventional WRR and DRR under CBR traffic and slightly better than DRR under video traffic as evident from plots of Figure 7 and 8. The conventional WRR does not guarantee bounded delay and hence its delay performance is worst than others.

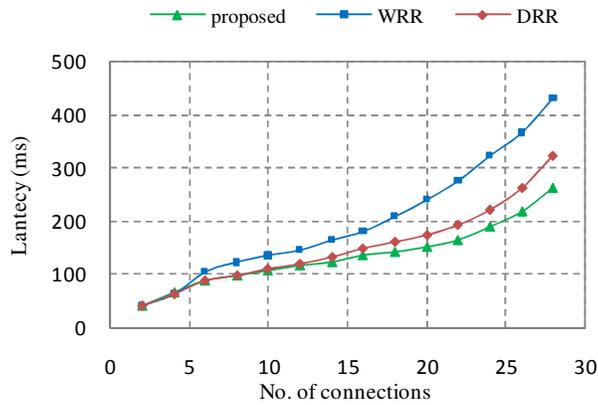


Figure 7. Latency of CBR audio traffic

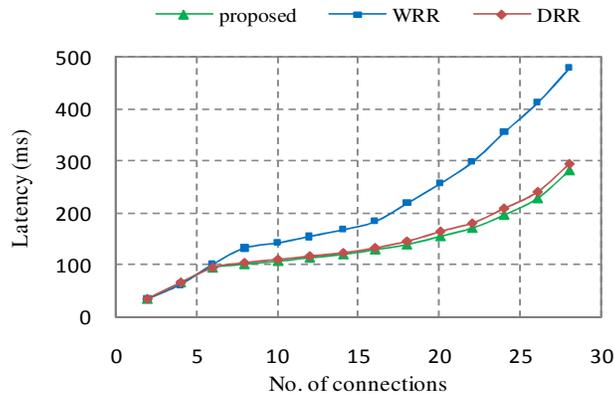


Figure 8. Latency of MPEG4 video traffic

Besides low latency, LL-WRR scheme also offers improvement in fairness as compared to conventional WRR. Figure 9 demonstrates as number connections increases LL-WRR offers much better fairness than conventional WRR. The fairness performance of conventional WRR is inferior to both DRR and LL-WRR. Since DRR takes into account packet length for scheduling, it possesses good fairness property and LL-WRR fairness closely follows DRR.

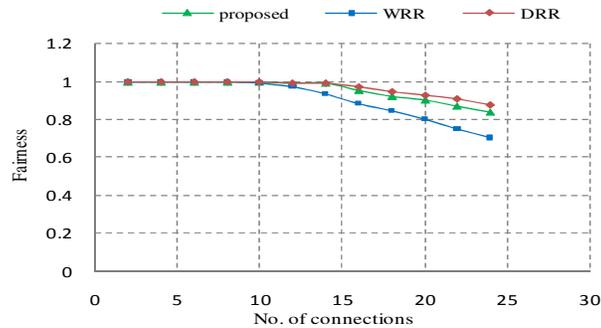


Figure 9. Fairness plot

The mobility of wireless node is an important factor to analyze for understanding its impact on latency. Latency varies with the variation in speed of node; initially increases rapidly up to 9 m/s and the more or less remains steady as shown in Figure 10 and 11. All schemes have this kind of behaviour under both CBR and video traffic. For CBR traffic, both WRR and DRR exhibit similar but larger latency than LL-WRR whereas for video traffic DRR has lower latency than WRR. When speed is increased from 1 m/s to 9 m/s, delay of LL-WRR scheme is increased from 38ms to approximately 150ms under CBR as well as video traffic. Then, it rises very slowly and tends to remain constant.

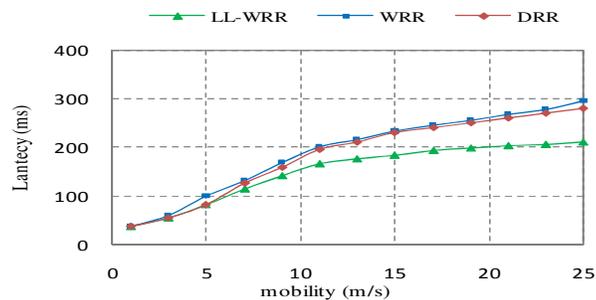


Figure 10. Latency for CBR with mobility (No. of nodes=10)

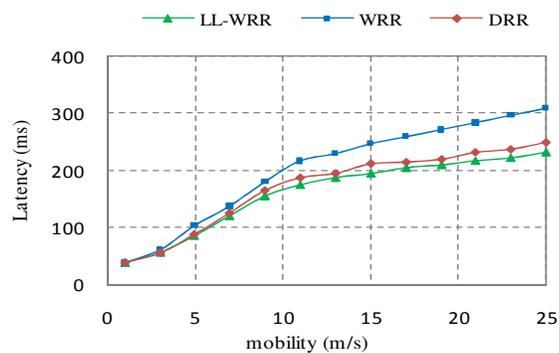


Figure 11. Latency for video with mobility (No. of nodes=10)

5. CONCLUSION

The work of this paper presents simple but efficient scheme named LL-WRR to improve conventional WRR in order to achieve low worst-case latency and improved fairness without much sacrificing rate. However, the computation of coefficient γ introduces additional complexity in proposed scheme but its overall impact will be very small, since it is computed only at the beginning of WRR cycle and not at every packet arrival and departure. The simulation results show that proposed scheme exhibits very low latency than conventional WRR for both CBR audio and MPEG4 video traffic. As compared to DRR, LL-WRR offers less latency for CBR audio traffic. Our scheme also offers better fairness than WRR and remains very close to DRR.

REFERENCES

- [1] IEEE 802.16e-2005, "Local and Metropolitan Networks — Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," 2006.
- [2] X. Gao, G. Wu and T. Miki, "End-to-end QoS Provisioning in Mobile Heterogeneous Networks," IEEE Wireless Communication, Vol.11, No.3, pp.24-34, June 2004.
- [3] Abhay K. Parekh and R.G.Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks," IEEE/ACM Transaction on Networking Vol.2, No.2, April 1994.
- [4] C.R. Bennett and H. Zhang, "WF2Q: Worst-case fair weighted fair queueing," IEEE INFOCOM'96, pp.120-128, March 1996.
- [5] S.J. Golestani, "A self-clocked fair queueing scheme for broadband applications," Proceedings of IEEE INFOCOM '94, Vol.2, pp.636-646, June 1994.
- [6] H.M. Alnuweiria and H.Tayyar, "Analysis of virtual-time complexity in weighted fair queueing," Journal of Computer Communications (Elsevier), Vol. 28, No.7, pp.802-810, May, 2005.
- [7] D. Stiliadis and A. Varma, "Efficient fair queueing algorithms for packet-switched networks," IEEE/ACM Transaction on Networking, Vol. 6, No.2, pp.175-185, April 1998.
- [8] L.H.Chen, Wu E.H.K, M.I. Hsieh, J T Horng and G H Chen, "Credit-based low latency packet scheduling algorithm for real-time applications," In proc. of IEEE Int. Conf. on Communication, Networks and Satellite (ComNetSat 2012), pp.15-19, July 2012.
- [9] D. Stiliadis and A. Varma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," IEEE/ACM Transaction on Networking, Vol. 6, No.5, pp.611-624, Oct. 1998.
- [10] H.M. Chaskar and U. Madhaw, "Fair Scheduling with Tunable Latency: A Round Robin Approach," IEEE /ACM Transaction on networking, Vol. 11, No.4, pp. 592-601, Aug. 2003.
- [11] W. Mardini and Mai M. Abu Alfool, "Modified WRR Scheduling Algorithm for WiMAX Networks", Journal of Network Protocols and Algorithms, Vol. 3, No.2, pp.24-53 July 2011.
- [12] A. Sayenko, T. Hamalainen, J. Joutsensalo and P. Raatikainen, "Adaptive scheduling using the revenue-based weighted round robin," Proceedings of IEEE ICON '04, Vol.2, pp.743-749, Nov. 2004.
- [13] R. Ouni, J. Bhar and K. Torki, "A New Scheduling Protocol Design Based On Deficit Weighted Round Robin For QoS Support In IP Networks," Journal of Circuits Systems and Computers, World Scientific Publishing Company, Vol. 22, No. 3, (2013) 1350012.
- [14] D. Bertsekas and R. Gallager, "Data Networks," 2nd Ed., Prentice Hall, Englewood Cliffs, New Jersey 07632.
- [15] The Network Simulator-NS-2, <http://www.isi.edu/nsnam/ns>.

AUTHORS

Zuber M. Patel is assistant professor at S.V. National Institute of Technology (SVNIT), in Surat, India. He received Bachelor's degree in Electronics from SVRCET, Surat in 1998 and obtained M.Tech (Microelectronics) from Indian Institute of Technology (IIT) Bombay, Mumbai in 2009. His current research areas are VLSI and Embedded design of Wireless transceiver, Digital VLSI Design and FPGA. Presently he is pursuing PhD from SVNIT, Surat.



Dr. (Mrs.) U. D. Dalal presently working as Associate Professor in Electronics Engineering Department of S. V. National Institute of Technology, Surat, INDIA. She has 20 years of academic experience. She completed her B.E. (Electronics) from SVRCET, Surat in 1991 and obtained M.E. (Electronics & Communications) from DDIT, Nadiad with Gold Medal. She obtained Ph D from SVNIT in 2009. Her research interest includes Wireless and Mobile Communication, OFDMA & MIMO technology, Fiber Optics and ISDN/ATM. She has published 100+ conference and journal papers at national and international level. Her book on "Wireless Communication" is published by Oxford University Press in July 2009. She is member of IEEE and life member of IETE, IE, ISTE & IJERIA.

