



The 6th International Conference on Ambient Systems, Networks and Technologies
(ANT 2015)

Probabilistic Prediction based Scheduling for Delay Sensitive Traffic in Internet of Things

Reema Sharma^a, Navin Kumar^{b,*}, Namratha B Gowda^a, T. Srinivas^d

^aDepartment of ECE, The Oxford College of Engineering, Bangalore, India, sharma80reema@gmail.com, namrathagowda25@gmail.com

^bDepartment of ECE, Amrita Vishwa Vidyapeetham (University), Bangalore, India, navinkumar@ieee.org

^dDepartment of ECE, Indian Institute of Science, Bangalore, India, tsrinu@ece.iisc.ernet.in

Abstract

This paper proposes a probabilistic prediction based approach for providing Quality of Service (QoS) to delay sensitive traffic for Internet of Things (IoT). A joint packet scheduling and dynamic bandwidth allocation scheme is proposed to provide service differentiation and preferential treatment to delay sensitive traffic. The scheduler focuses on reducing the waiting time of high priority delay sensitive services in the queue and simultaneously keeping the waiting time of other services within tolerable limits. The scheme uses the difference in probability of average queue length of high priority packets at previous cycle and current cycle to determine the probability of average weight required in the current cycle. This offers optimized bandwidth allocation to all the services by avoiding distribution of excess resources for high priority services and yet guaranteeing the services for it. The performance of the algorithm is investigated using MPEG-4 traffic traces under different system loading. The results show the improved performance with respect to waiting time for scheduling high priority packets and simultaneously keeping tolerable limits for waiting time and packet loss for other services.

Crown Copyright © 2015 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Conference Program Chairs

Keywords: Internet of Things; Service Differentiation; Queue Length; QoS; Queue Management, Delay Sensitive.

1. Introduction

Internet of Things (IoT) is the next evolution of the Internet where devices of different types and capabilities are connected through Internet protocol (IP) and web services to make intelligent decision and to exchange information

* Corresponding author. Tel.: +91 80 25183700 fax: +91 80 28440092.

E-mail address: navinkumar@ieee.org

without much relying on the human association¹. Various intelligent devices in smart home² and industry are usually connected with personal mobile devices through GSM, GPRS or 3G networks³. This resulted in exponential increase in the volume of Internet data and thus created a challenge for buffer management. At the same time the need for a scheduling scheme to ensure instant communication without queuing delays and packet losses especially for the sensitive data. The smart devices like wireless cameras used in closed camera television (CCTV) and home network with limited buffer capacity needs effective buffer management, packet scheduling and service differentiation to provide preferential treatment to delay sensitive traffic. Also, offering sufficient transmission opportunity within tolerable delay for the multi user video transmission applications is challenging.

For example, an intelligent CCTV camera takes a picture of an intruder and instantly sends a priority message together with the photograph of the intruder to the owner's mobile device. In this perspective, it is of vital importance to develop service models that guarantees QoS for delay sensitive applications in IoT¹. Current approaches provide unsatisfactory solutions for delay sensitive applications because it is observed that more video contents are captured than human can possibly handle⁴. It becomes more critical if the resources available are not sufficient. Furthermore, most of the existing slot allocation policies of scheduling such as round-robin or rate-adaptive round-robin are stationary because the allocation of the current slot does not depend on the allocation of previous slots⁵. In this paper, we propose a model to consider IoT applications which are more sensitive to delay.

Here, we propose a novel approach based on round robin policies and additionally, a unique characteristic of allocation of number of serviced packets in current slot is made dependent on the allocation of number of serviced packets in previous slot. This characteristic is implemented by calculating difference in probability of average queue length of high priority packets in consecutive cycles and allocating bandwidth to each service class based on this difference. The proposed model analyses finite capacity queues with service differentiation⁶ and provides a solution to evaluate the performance of traffic generated by smart devices under varying traffic conditions so as to ensure preferential treatment of high priority delay sensitive traffic. The algorithm is tested for different buffer sizes to understand the effect on currently small buffer in smart devices in IoT and prospective bigger sizes in the future. Also, for variable buffer size, it becomes important to record the increased queue length to approximate the required resources in the next round (cycle).

The rest of the paper is organized as follows. System model and assumptions are described in Section 2. In Section 3, the prediction model is formulated and solved to attain the desired performance. Results are presented in section 4, while section 5 states the conclusion and future work.

2. Probabilistic Model and Its Assumptions

In the proposed model it is assumed that the arriving traffic is classified into highest priority (emergency traffic), medium priority (constant rate traffic) and low priority (normal traffic) traffic and is stored in different queues. In this policy a unique characteristic of calculation of current departure packets from each priority queue is proposed. Increase/decrease in average queue length of high priority queue is predicted in consecutive cycles and based on this, the required bandwidth for each service class is calculated. In other words, it determines a weighting coefficient⁷ for each queue to calculate the average number of packet departures from the queue before moving to the next queue. The weighted round robin scheduler serves all the queues in one cycle. Scheduling is performed at the beginning of each cycle.

Let the highest priority queue has a buffer queue size of B1 and other service classes have a buffer queue size of B2. We assume that packet arrivals occur independently for each service class and follow a Poisson process with a mean arrival rate of $\lambda = \sum_{n=0}^{n_{max}} n \cdot P(n)$ packets per cycle as shown in Fig.1, where $P(n)$ represents the probability of n packet arrivals in a cycle and n_{max} denotes the maximum number of packet arrivals. Let $\lambda_{HA}, \lambda_{HQ}, \lambda_{HL}$ are respectively the number of packets arrived, packet stored in the queue and packet lost from high priority queue. Similarly, $\lambda_{MA}, \lambda_{MQ}, \lambda_{ML}$ are respectively the number of packets arrived, packet stored in the queue and packet lost from medium priority queue and $\lambda_{LA}, \lambda_{LQ}, \lambda_{LL}$ are respectively the number of packets arrived, packet stored in the queue and packet lost from low priority queue. We consider a probabilistic Markov Chain Based Model to describe the system.

The following points illustrate the design approach:

- Starting with cycle C_0 at time slot t_0 , the scheduler assumes the probability of instantaneous queue length q_0^1 (probability of number of packets stored in the queue), probability of average queue length Q_0^1 and probability of weights assigned to highest priority queue W_0^1 as 0, 0 and 0.3 respectively. Similarly probability of instantaneous queue length (q_0^2, q_0^3) and probability of weights (W_0^2, W_0^3), assigned to medium and low priority queues are 0, 0.3 and 0, 0.4 respectively.
- At cycle C_I and time slot t_I , the actual traffic randomly arrives in the respective queue. The scheduler measures the probability of instantaneous queue length, probability of average queue length (calculated by equation (1)) and calculates the weights (to allocate bandwidth dynamically) assigned to highest priority queue.
- During the whole process of dynamic bandwidth allocation, the weights will be dynamically updated based on the prediction of difference in the average queue length (increase or decrease from previous cycle) and addition of the previous cycle's weight to this.
- Based on this weight prediction of high priority queue, the weights or bandwidth allocated to medium and low priority traffic is calculated for the same cycle.
- When the probability of average queue length of the highest priority queue is decreased at current cycle than the previous cycle, the bandwidth allocation value of current cycle will be decreased.
- When the probability of average queue length of highest priority queue is increased at current cycle than the previous cycle, then its bandwidth will be increased based on thresholds used and the amount of increase in the average queue length.
- In comparison with previous time slot, the maximum amount of increased weight at current time slot will not be more than M_c (where $M_c = \text{upper} - \text{previous weight}$).
- The unconsumed bandwidth from high priority traffic is allocated to medium traffic and if there remains unutilized bandwidth with medium traffic, it will be allocated to the low priority traffic.

From this model, it is easy to understand that the weight allocated at each cycle is a random variable. Also, the weight assigned to the current cycle is only dependent on the weight assigned at the previous time slot. If probability of average queue length of the highest priority packet at current cycle is larger than the previous, then the amount of bandwidth allocated will probably be the larger. Therefore, considering changes of average queue length of the highest priority queue, the weight of each queue can be considered as a Markov chain.

3. The Probabilistic Prediction Based Joint Packet Scheduling and Bandwidth Allocation Model

The probabilistic prediction based joint packet scheduling and dynamic bandwidth allocation mechanism refers to the one that differentiates the services:

- based on priority,
- measures the probability of traffic increased at current time slot from the previous time slot,
- predicts the number of packets to be scheduled at current time slot, and
- amount of bandwidth to be allocated to each service.

Since the analytical procedure of each intermediate node is the same, we randomly select a node to analyze the algorithm. The structure of the node and scheduling mechanism is shown in Fig. 1.

Now the Markov chain formulation can be described to calculate the average number of packet departures $T_c^{n,s}$ over the cycle c for n th service class within time slots s . The system has a state space as follows: $X_c^n = \{q_c^n, Q_c^n, W_c^n\}$; where n denotes the service class to be selected during cycle c ($n=1$ denotes highest priority class, $n=2$ denotes medium priority class and $n=3$ denotes low priority class to be selected in one cycle (round)). A cycle is a duration of time which consists of s time slots of variable length according to the packet size (variable packet size is considered) and number of departed packets. It also allows all the queues to service some packets in a cycle. q_c^n denotes the current queue length of n th service class during cycle c . W_c^n denotes the weights assigned to n th service class in cycle c and all the values should be non negative integers.



Fig.1 The System Model

The probability distribution for state vector X_c^n in cycle c is denoted by $\prod_c^n = \prod(q_c^n = j, Q_c^1 = k, W_c^n = w$ for $n \in \{1,2,3\}$, $j \in \{0,1,2,\dots,B1\}$ for $n=1$ and $\{0,1,2,\dots,B2\}$ for $n=2$ and 3 , $k \in \{0,0.01,0.02 \dots B1'\}$ when $n=1$ and $w \in \{0, 0.1,0.2,\dots,1\}$ where $\prod(q_c^n = j, Q_c^1 = k, W_c^n = w$ denotes the probability that the values of state variables q_c^n, Q_c^n, W_c^n are j,k,w respectively. The state vector for the first cycle for $n = 1$ is denoted by $X_1^1 = \{q_1^1, Q_1^1, W_1^1\}$. According to the scheduling scheme the value of this state vector is initialized with $\prod_1^1 \{0,0,0.3\}$. The probability distribution of state vector X_c^n is calculated based on state transition matrix $P(X_{c+1}^n/X_c^n)$.

Let a state transition occurs at the end of each cycle. $P(X_{c+1}^n/X_c^n)$ can be represented as in matrix form as:

$$\begin{matrix} P(0,0,0.3|0,0,0.3) & P(1,0.1,0.3|0,0,0.3) & \dots & P(B1,B1',0.7|0,0,0.3) \\ P(0,0,0.3|1,0.1,0.3) & \dots & \dots & P(B1,B1',0.7|1,0.1,0.3) \\ \dots & \dots & \dots & \dots \\ P(0,0,0.3|B1,B1',0.7) & \dots & \dots & P(B1,B1',0.7|B1,B1',0.7) \end{matrix}$$

where the transition probability $P_{(j',k',w'|j,k,w)}$ is calculated as:

$$\begin{aligned} P_{(j',k',w'|j,k,w)} &= P(q_{c+1}^n = j', Q_{c+1}^n = k', W_{c+1}^n = w' | q_c^n = j, Q_c^n = k, W_c^n = w) \\ &= \frac{P(q_{c+1}^n = j', Q_{c+1}^n = k', W_{c+1}^n = w', q_c^n = j, Q_c^n = k, W_c^n = w)}{P(q_c^n = j, Q_c^n = k, W_c^n = w)} \\ &= \frac{P(W_{c+1}^n = w' | W_c^n = w, Q_{c+1}^n = k', Q_c^n = k, q_{c+1}^n = j', q_c^n = j)}{P(q_c^n = j, Q_c^n = k, W_c^n = w)} \cdot P(q_c^n = j, Q_c^n = k, W_c^n = w, Q_{c+1}^n = k', q_{c+1}^n = j') \\ &= P(Q_{c+1}^n = k' | Q_c^n = k, q_c^n = j) \cdot P(q_{c+1}^n = j' | q_c^n = j) \cdot P(W_{c+1}^n = w' | W_c^n = w, Q_{c+1}^n = k', Q_c^n = k) \end{aligned} \tag{1}$$

In (1), the state variables q_{c+1}^n is independent of Q_{c+1}^n and W_{c+1}^n . Besides, q_{c+1}^n does not depend on Q_c^n and W_c^n . Similarly, Q_{c+1}^n does not depend on W_c^n . The corresponding queue status q_{c+1}^n at the beginning of cycle c can be expressed as:

$$q_{c+1}^n = q_c^n - d_c^n + a_c^n \tag{2}$$

where a_c^n denotes the number of packets that are arriving the n th queue at cycle c ; d_c^n represents the number of packets departing from n th queue during cycle c . The number of packets departure in cycle c can be given as:

$$d_c^n = \min(q_c^n, b_c^n) = \min(j, b_c^n) \tag{3}$$

where b_c^n is the number of packets to be scheduled and calculated according to the proposed scheme for queue n during cycle c (explained later).

The corresponding queue has to be selected in a round robin fashion, so we calculate n as:

$$n = (n \text{ mod } 3) + 1 \tag{4}$$

Based on (2), (3) and (4), we have $a_c^n = q_{c+1}^n - q_c^n + d_c^n = j' - j + \min(j, b_c^n)$ and the conditional probability $P(q_{c+1}^n = j' | q_c^n = j)$ is obtained as:

$$P(q_{c+1}^n = j' / q_c^n = j) = \begin{cases} P(A = a_c^n) & , \quad \text{if } j' < Q, 0 \leq a_c^n \leq nmax \\ \sum_{a_c^n = a_c^n}^{nmax} P(A = a_c^n), & \text{if } j' = Q, 0 \leq a_c^n \leq nmax \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Some of the packets will be dropped from each service class if the corresponding buffer is full.

The conditional probability, $P(Q_{c+1}^n = k' / Q_c^n = k, q_c^n = j)$ is calculated from (6) and conditional probability $P(W_{c+1}^n = w' / W_c^n = w, Q_{c+1}^n = k', Q_c^n = k)$ is calculated from (7):

$$P(Q_{c+1}^n = k' / Q_c^n = k, q_c^n = j) = (1 - 0.01) * P(Q_c^1 = k) + 0.01 * P(q_{c+1}^n = j' | q_c^n = j) \quad (6)$$

$$P(W_{c+1}^n = w' / W_c^n = w, Q_{c+1}^n = k', Q_c^n = k) = \begin{cases} P(Z = W_1^1) & \text{if } n = 1 \text{ and } 0 \leq k' < 0.083 \\ P(Z = \alpha n) & \text{if } n = 1 \text{ and } 0.083 \leq k' < 0.3667 \\ P(Z = U) & \text{if } n = 1 \text{ and } 0.3667 \leq k' \leq 1 \\ P(Z = W_1^1) & \text{if } n = 2 \\ P(Z = 1 - \sum_{n=1}^2 W_c^n) & \text{if } n = 3 \end{cases} \quad (7)$$

where αn is the change in average queue length, $\alpha n = Q_{c+1}^n - Q_c^n$, $W_{c+1}^n = W_c^n + \alpha n.T$, $\alpha n = \frac{W_{c+1}^n - W_c^n}{T}$, $T = \frac{0.3}{(Max_{Th} - Min_{Th})}$, $b_c^n = w' * (\frac{BW_{Link}}{Size_{of_packet}})$. Here $Max_{Th} = 0.3667$ and $Min_{Th} = 0.083$ are maximum and minimum

threshold values of probability of average queue length of high priority queue. These values are chosen as an indicator to assign the appropriate weights. W_1^1 is initial weight of 0.3 in first cycle and U is maximum weight of 0.7 for high priority queue. The probability distribution $\prod_{c+1}^n = \prod_c^n * P(X_{c+1}^n / X_c^n)$. The total packets serviced in one cycle can be calculated as: $T_c^n = \sum_{n=1}^3 d_c^n$. Total high priority packets departed in N cycles are: $Y = \sum_{c=1}^N d_c^1$. Total medium priority packets serviced in N cycles can be calculated as: $Z = \sum_{c=1}^N d_c^2$. And the total low priority packets serviced in N cycles can be calculated as: $D = \sum_{c=1}^N d_c^3$, where $c=1,2,\dots,N$.

4. Simulation Results and Analysis

We have investigated many performance measures of the proposed algorithm using actual MPEG-4 traffic traces (highest priority queue) under different system loading. Also, the traffic distribution for other medium and lower service class is considered as Poisson. Packets of variable sizes are considered where the size varies from 272 bytes to 356456 bytes for video packets. Size of medium class packets varies from 200 bytes to 500 bytes and for low priority traffic packet sizes are varying from minimum 100 bytes to 2500 bytes packets. The various performance measures considered to evaluate the performance of this packet scheduling scheme are packet lost ratio for different service queues by considering different buffer sizes, waiting time for packets of different services again by considering different buffer sizes and throughput for different service classes.

Waiting time in buffer at each node increases the risk of end to end delay. So waiting time for high priority service in the intermediate nodes should be less to achieve small end to end delay and to resolve the problem of jitter. Also, packet loss for high priority queue should be negligible. For the medium and low priority queue, the losses should be the minimum.

In a WRR scheduler, Average waiting time (A_n) of packets in n th queue is obtained as⁸:

$$A_n = \frac{[(Total_{linkBW} - (BW_{nlink})) * (avg_{pkt_{sz}})]}{Total_{linkBW} * 2(1 - (\frac{\rho_n * (BW_{nlink})}{\mu_n * Total_{linkBW}})) * Total_{linkBW}} \quad (8)$$

where BW_{nlink} is the bandwidth utilized by n th queue to schedule packets, $avg_{pkt_{sz}}$ is the average size of the packet to be scheduled; $Total_{linkBW}$ is bandwidth of the channel, ρ_n is arrival rate of packets in n th queue, μ_n is service rate of packets from n th queue.

Average packet dropped ratio represents that an arriving packet is dropped at the queue because the queue status is full. Average packet dropped ratio for high priority, medium and low priority queue is given by:

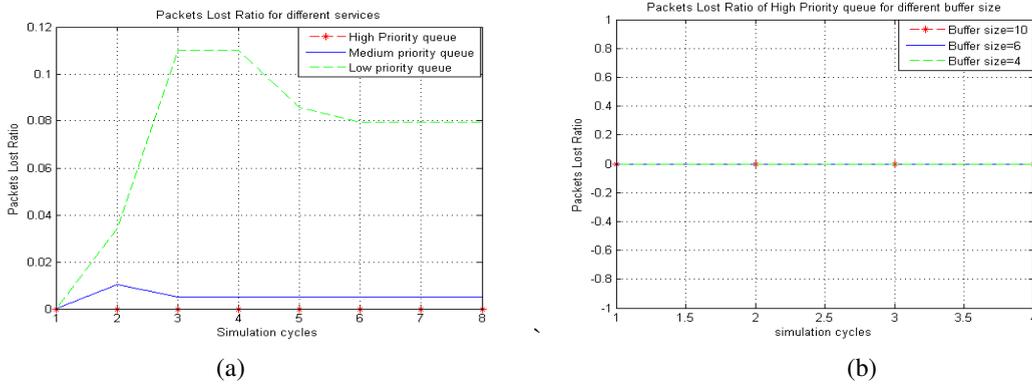


Fig.2: Packet lost ratio: (a) For different service classes with maximum buffer size (10,100,100); (b) High priority queue for different buffer size (10,6,4)

$$Average\ Packet\ Dropped\ Ratio = \begin{cases} \frac{\sum_{c=1}^N (Q_c^1 - B1)}{\sum_{c=1}^N Q_c^1} & \text{if } Q_c^1 > B1 \text{ and } n = 1 \\ \frac{\sum_{c=1}^N (Q_c^n - B2)}{\sum_{c=1}^N Q_c^n} & \text{if } Q_c^n > B2 \text{ and } n = 2 \text{ or } 3 \end{cases} \quad (9)$$

The average user throughput represents sum of average number of packets of a particular service type that are successfully transmitted in each cycle. Average user throughput is given by Eq 10.

$$Average\ user\ Throughput = \sum_{c=1}^N \left(\frac{T_c^n}{Q_c^n} \right) \text{ where } n = 1,2,3 \quad (10)$$

Fig.2(a) shows the packet lost ratio per cycle for high priority, medium priority and low priority queue when maximum buffer sizes for the different service classes are 10,100,100 respectively. Source is continuously generating packets of size 272 bytes to 356456 bytes at a data rate varying from 43.5kbps to 2.8Mbps for high priority queue. It is clear that there is no packet loss for high priority class using proposed scheduler scheme. Another source is continuously generating packets of the size 200 bytes to 500 bytes at a data rate varying from 352kbps to 504kbps for medium priority queue. In this case the average 0.005% packets of medium priority is lost. The third source is generating packets of size varying from 100 bytes to 2500 bytes at a data rate of 16kbps to 400kbps. In this case, an average of 0.11% of packets is lost. Fig.2(b), 3(a) and 3(b) show the packet lost ratio of high priority queue, medium priority queue and low priority queue respectively with different buffer sizes. We tested the algorithm for many values and plotted only few because of lack of space and we observed with different readings that for any buffer size > 2, there is no packet loss for high priority queue. For medium priority class, the packet loss ratio is negligible for buffer size 100 and it increases by 10% for reduction of buffer size by half and 25% for reduction of buffer size by 1/4th. Also, for low priority traffic with buffer size >=100, the packet lost is almost 10% but it increases when we take the buffer size <100 and become worse if we reduce it to <60. In Fig. 3(b) we can see that in between 10 to 30 simulation cycles the packet lost from low priority is increased to 90% and then comes down to 60%. This much packet lost has happened due to three conditions: (i) The packet size of high priority packets during this cycle is very high and between range 200000 bytes to 346546 bytes and more bandwidth of the link is allocated to high priority queue and rest to medium priority queue; (ii) the buffer size is <=50 so packets will not be stored but lost; (iii) Data rate is high i.e. approximately 20 packets arrives in queue in each cycle. So if buffer size for low priority queue is taken more than 100 then the waiting time for low priority packets will increase which is definitely not an issue as these packets are not delay sensitive but packet lost will be less (which is desired).

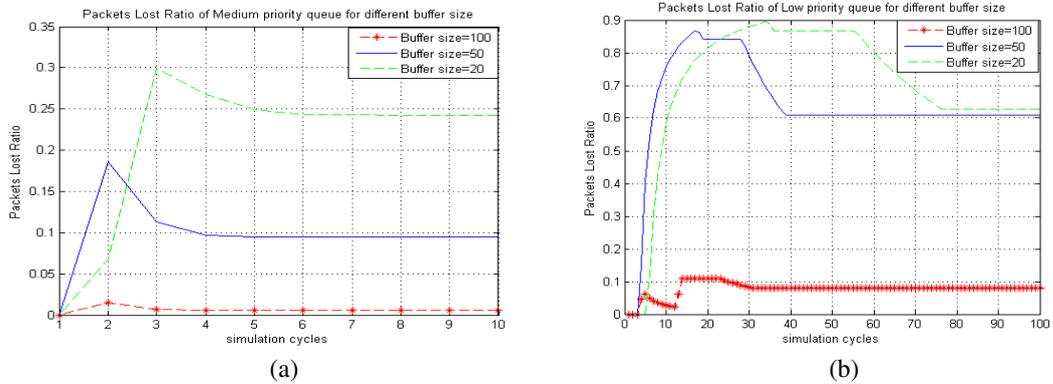


Fig.3. Packet loss ratio for: (a) medium priority queue for different buffer sizes (100,50,20); (b) low priority queue for different buffer sizes (100,50,20)

Waiting time required for packets to be scheduled from high priority, medium priority and low priority queues is shown in Fig.4(a),4(b), and 4(c). It is observed that waiting time for high priority packets with buffer size 10, 6 and 4 is the same. Similarly it is almost similar for medium priority packets also even with varying buffer sizes to 100,50,20. The average waiting time for packets of low priority for different buffer sizes is more when buffer size is ≥ 100 as the packets will be waiting for more time in queue for low priority packets. Waiting time also depends on the variable size of the packets. If packet size is more waiting time in the queue will increase because in that particular cycle it will be scheduled only if its required bandwidth is \geq allocated bandwidth otherwise it may take 2 or more than 2 scheduling cycles. So the buffer size has more impact on lost packets and with the proposed scheduling scheme if buffer size for low priority queue is kept high (>100) packet loss will be reduced but waiting time of packets in the queue will be increased. Buffer size for high

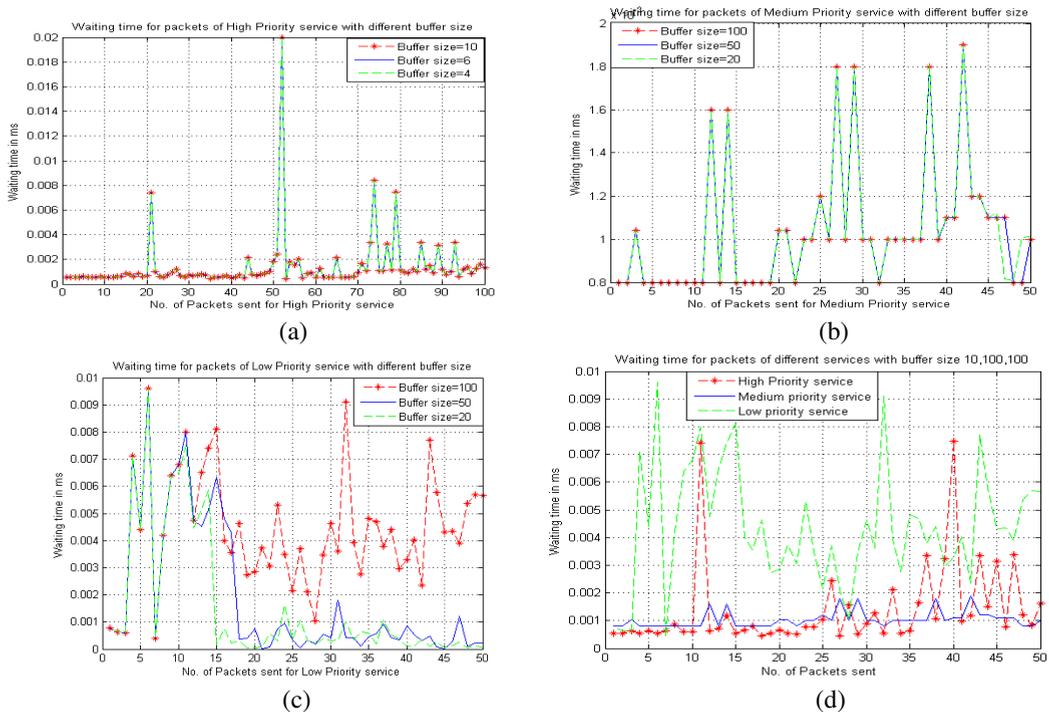


Fig.4. Waiting time for packets of (a) high priority service with different buffer sizes;(b) medium priority service with different buffer sizes; (c) low priority service with different buffer sizes; (d) high, medium and low priority services with buffer size 10,100,100

priority queues can be kept less because scheduler dynamically allocates more bandwidth to high priority to guarantee them better QoS and hence waiting time is less for those packets.

Fig. 4(d) shows that waiting time required for high priority queue is less than medium priority which is further less than low priority queue. This is generally very important to meet guaranteed QoS in terms of end to end delay for delay sensitive applications. It also shows some peaks in red (high priority) i.e. the waiting time is increased for high priority packets. These peaks only comes when continuous packets of big size arrives in high priority queue and it may take 2 or more cycles to schedule that packet. So the results show the advantage of the proposed scheduler with respect to waiting time in the router for scheduling high priority packets and simultaneously keeping tolerable limits for waiting time of other services. Even, throughput is high and packet lost is low for high priority and medium priority packets.

Fig.5 describes the throughput for different services. Throughput is high for all services with more buffer size (10, 100,100) as shown with red bars for high, medium and low respectively. The throughput of medium and low priority is decreased more by reducing buffer sizes to (4,20,20). This is because if data rate is high i.e. more number of packets are arriving in the same cycle for all three services, the scheduler gives priority to high service class, medium service class and low priority class in order. Because of this prioritization there is no loss for high priority packets. Simultaneously medium and low priority packets need to wait for small time to get service so if buffer size is less for them some packets will be lost due to buffer overflow.

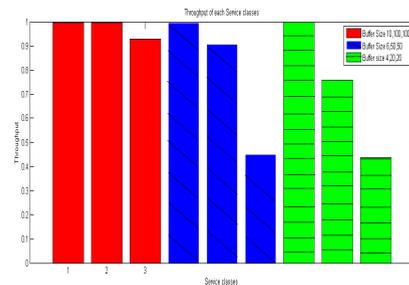


Fig.5. Throughput of high priority, medium priority and low priority services under different buffer sizes

5. Conclusion and Future Scope

The proposed packet scheduling scheme performs better in terms of throughput, less scheduling time for packets of high and medium priority of services. Also, for IoT applications where large amount of data is sent and where smaller buffer sizes are available such as with sensors and other devices; normally it is challenging to provide better QoS. In such a scenario this scheduler can be used. Scheduler is tested for different buffer sizes with same data rate and no degradation in QoS is found for the high priority service. However, a very small effect on medium priority packets is recorded by reducing buffer size to 1/4th. But the performance of low priority services is degraded by lowering the buffer size. So, if buffer size for low priority services is considered more, this scheduler provides guaranteed services for high priority, medium priority and better performance for low priority services in case of high buffer size. It is also tested for different data rates and variable packet sizes and the results are found satisfactory, according to the service type required in terms of packet lost, throughput and waiting time.

We are in the process of optimizing the performance of the scheduler. It is expected that optimal performance may reach for most of the classes and average buffer size.

References

1. Irfan Awan and Muhammad Younas, "Towards QoS in Internet of Things for Delay Sensitive Information", *MobiWIS 2013 workshops, Springer International Publishing Switzerland CCIS 183*, pp 86-94, 2013.
2. Ricquebourg, V, Menga D, Durand D, Marhic B, "The Smart Home Concept: Our Immediate Future" *IEEE International Conference on E-Learning in Industrial Electronics*, 23-28, 2006.
3. Luigi Atzori, Antonio Iera, Giacomo Morabito, "The Internet-Of-Things: A Survey", *Computer Networks Journal ELSEVIER*, Vol. 54, pp. 2787-2805, 2010
4. Yen Kuang Cheng, Shao-Yi Chien "Perceptual video camera for Internet of things" *IEEE Proc. Of Visual Communications and Image Processing (VCIP)*, pp 1-7, 2012.
5. Jie Xu, "Non-Stationary Resource Allocation Policies for Delay constrained Video Streaming : Application to Video over Internet- of Things-Enabled Networks" *IEEE Journal on Selected Areas in Communications*, (Volume:32, Issue:4), pp 782-794, 2014.
6. Long B.Le, Ekram Hossain, Aathiru S Alfa, " Service Differentiation in Multirate wireless Networks With Weighted Round Robin Scheduling and ARQ Based Error Control", *IEEE Transactions on Communications*, Vol. 54, No.2, pp.208-215, February 2006.
7. Xianmin Geng, Aiwen Luo, Zhinjun Sun, and Yu Cheng, "Markov Chains Based Dynamic Bandwidth Allocation in Diffserv Network" *IEEE Communication letters*, Vol. 16, No. 10, October 2012.
8. Valadimir Hottmar, Bohumil Adamec, "Analytical Model of a Weighted Round Robin Service System", *Hindawi Publishing Corporation, Journal Of Electrical and Computer Engineering*, pp.1-6, Vol. 2012.