

A COMBINED LOW LATENCY AND WEIGHTED FAIR QUEUING BASED SCHEDULING OF AN INPUT-QUEUED SWITCH

¹D. Raghupathikumar and ²K. Bommanna Raja

¹Department of CSE, Government College of Engineering, Bodinayakkanur, Theni Dt Tamilnadu, India

²Department of BME, P.S.N.A. College of Engineering, Dindigul Tamilnadu, India

Received 2014-01-22; Received 2014-02-21; Accepted 2014-03-25

ABSTRACT

Input queuing has become dominant and popular building blocks for high speed crossbar switches with many ports and fast line rates because they require minimum speed-up of memory bandwidth. Input Queued switches with finite Virtual Output Queues guarantees QoS performance in terms of throughput and average delay. A switch performs two functions Queuing and Scheduling. Queue Management algorithm manages the size of the queues and drops packets when necessary or appropriate. Scheduling algorithms determine next packet to transfer and solves conflicts with the switching fabric. Fairness and Starvation are another two properties of IQ switches and it is analyzed in finite VOQ in this works. Fairness performs fair allocation of bandwidth among flows and prevents flows from misbehaving flows. Starvation of VOQ prevents serving High priority queue. The motivation behind this study is to schedule the HoL packets queued in finite VOQs by Framing with Low Latency Queueing (LLQ) and Weighted Fair Queueing (WFQ). This queueing technique of VOQ is measured in terms of throughput and average delay by fair allocation of bandwidth with WFQ and Starvation-free queue with LLQ.

Keywords: Input Queued Switch, Scheduling, Queuing, Low Latency Queueing, Weighted Fair Queueing

INTRODUCTION

The rapid growth of the Internet and quick implementation of the technology in recent years requires high speed switches and routers in backbone networks. This explosive growth also needs better Quality of Service (QoS) requirements. These outcomes bring many challenges and opportunities to the research on the high speed performance switches and routers by providing faster data rates and increased link speeds. Increasing the link speed and data rates needs large buffer size. Under buffered switches leads to packet loss, in turn suffer quality of service degradation especially for audio and video applications. Over buffered switches imposes increased latency, complexity, cost and power consumption (ATM, 1994). Owing to cons of both larger and smaller fixed sized buffer is used in this proposed

works to measure and analyze throughput and packet average delay. In this work, throughput and delay of the packet has been studied extensively in the context of ATM switching fabrics for fixed length packets (such as cells in ATM terminology (Awedeh and Mouftah, 1995)). In principle, an ATM switch shall perform the following two basic functions: Queuing and Scheduling. Two major Queuing organizations in switch have been proposed in the literature namely Output Queued (IQ) and Input Queued (OQ). Output queued switches can provide 100% throughput and arbitrary QoS efficiently, but they are infeasible (Awedeh and Mouftah, 1995) to implement at high speeds and high port densities due to the switch and memory speed requirements. Input Queued switches support better scalability and speedup features. However, IQ switches suffer from Head of Line Blocking (HoL) which limits throughput to 58.6%

Corresponding Author: D. Raghupathikumar, Department of CSE, Government College of Engineering Bodinayakkanur, Theni Dt Tamilnadu, India

when $N \rightarrow \infty$ (McKeown, 2004). To overcome HoL, Virtual Output Queue (VOQ) (McKeown, 1997; McKeown *et al.*, 1997) is provided with each input port following FIFO discipline. The number of VOQ depends on the size of the switch N . The other function is scheduling which manages cell transfer by selecting a cell according to proposed scheduling algorithm used and solves contentions with the switching fabric when two packets contend for the same port. Different styles of scheduling algorithms like iSLIP (Shreedhar and Varghese, 1995), PIM (Muppala and Hamdi, 1999) have been proposed in the literature. The proposed scheduling and queuing algorithms were works with an implicit assumption of infinite buffer space to achieve 100% throughput with degradation in average latency. Our proposed Framed Low Latency Weighted Fair Queueing (FL^2WFQ) measures throughput and packet delay in *finite size Virtual queue buffers*. The Weighted Fair Queueing (Eric, 2009) (WFQ) serves the packets in increasing order of their finish time which guarantees fairness among the competing flows. LLQ (Stephens *et al.*, 1999) is a combination of Priority Queueing (PQ) and WFQ. Low Latency Queueing (LLQ) gives priority to real-time traffic such as video, audio datas. LLQ especially dequeue packets with highest priority queue first.

2. ORGANIZATION OF THE PAPER

The rest of the paper is organized as follows. Section 3 briefly discuss the literature of IQ. Section 4 explains about Queuing strategies. Section 5 brief about the algorithm for scheduling of IQ switches Framed by Low Latency Weighted Fair Queueing. The simulated works and results are discussed in section 6 and 7 respectively. Finally, the paper is concluded in section 8.

3. BACKGROUNDS

ATM switches are represented by architectures using a non-blocking interconnection network. A non-blocking interconnection is a crossbar structure that guarantees absence of switching conflicts (internal conflicts) between cells addressing different switch output ports. ATM switches can be broadly classified into Time Division Switches (TDS) and Space Division Switches (SDS). In Time Division Switches, a single communication highway is shared by all input and output ports. The drawback of TDS is the single shared highway defines the capacity of the entire switch fabric and thus fixes an upper limit on the capacity beyond which it cannot grow and reduces the throughput. Space Division Switching in which single transmission-

path routing determination is accomplished in a switch by using switch path.

Crossbar is the basic switching fabric for high speed Space Division Switches. A crossbar switch can transfer up to N packets in parallel from different input ports to different output ports without conflicting by using crossbar constraints. Input Queued (IQ) switch employs crossbar switching fabric for transfer of cells. The architecture of IQ switch is shown in **Fig. 1**.

Generally speaking, a switch has four components:

- Input buffers
- Output buffers
- Switching fabric
- Scheduler

The cells arrived at the input ports are buffered at *Input buffers*. The cells destined to another link from the input buffers are buffered at the *Output buffers* present in the output ports. The *Switching fabric* is configured by *Scheduler* using scheduling algorithms to match the input and output ports and atmost one cell is transferred from one input port to output port in one timeslot via crossbar fabric. Depending on the position of buffer the switch can be classified as Output Queuing, Input Queuing and Shared Queuing. Each queuing has its own pros and cons.

Output Queuing (OQ) switch architecture having buffer at the output port and buffer of infinite size can always achieve better throughput for all kinds of traffic (McKeown *et al.*, 1997; Sommers *et al.*, 2005; Demers *et al.*, 1989). However OQ suffers with the internal speedup (Parkeh and Gallager, 1993) problem of the switch. Because packets destined for the same output port may arrive simultaneously from many input ports, the output buffers need to enqueue the packets at a much higher rate. In other words, the switching fabric and buffer needs to operate at N times faster than link rate which needs scalable increase in link rates and its impractical for high speed switches with large number of ports. However, only a single cell may be served by an output port, thus causing possible *output contention*. The Shared Queue (SQ) approach still provides for output queuing, but rather than have a separate queue for each output, all memory is pooled into one completely shared buffer and shared by all input and output lines. The recirculation may cause out-of-sequence errors between cells in the same virtual connection unless steps are taken to prevent it.

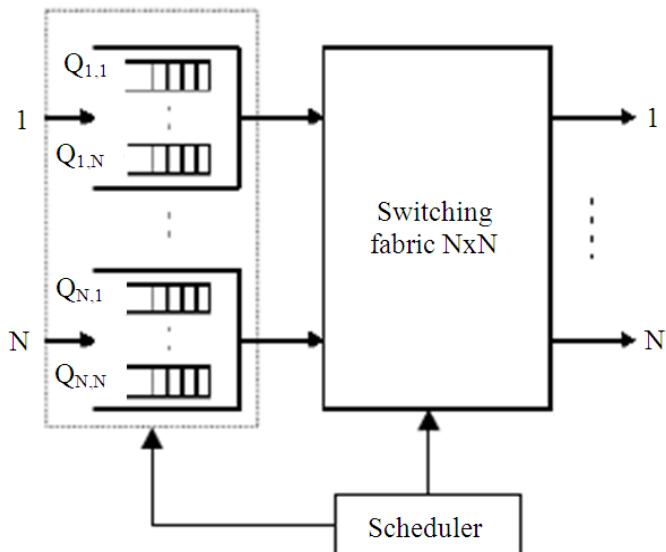


Fig. 1. Schematic diagram of an input-queued switch

The shared memory should operate in the aggregate rate of both input ports and the output ports and hence very complex in implementing high speed switches. Owing to the benefits of Input Queueing, IQ switch is proposed to use in this studies.

3.1. Input Queuing

Input Queuing becomes very appealing for switches with fast line rates or with a large number of ports. The Input Queued switch provides a lowcost architecture for cross bar based switches designing and it is attractive for very high bandwidth. This is due to that both the memories and the switch fabric needs only to operate at the same speed as the line rate which is independent of N . During a switching cycle cells of a fixed length are to be switched from inputs to outputs via switching fabric. The queues are served according to the First Come First Serve discipline. When the packet at the head of the FIFO queue is blocked, all the packets behind it are prevented from being transmitted even if the output port they are destined to is idle. This is due to Head of Line (HoL) blocking. HoL blocking limits the throughput of each input port to a max of 58.6% under uniform traffic and is much lower than that of burst traffic (Shreedhar and Varghese, 1995). To overcome this problem, each input queue maintains FIFO queue for each output, hence a total of $N \times N = N^2$ queues are present. This separation of queues eliminates performance degradation due to HoL blocking and the queue is aid to be Virtual Output Queue (VOQ) or Destination Queue (DQ).

Throughput and delay are the two main quantities with regards to the performance of a switch scheduling algorithm. A packet scheduling algorithm is “stable” algorithm if it achieves 100% throughput. Packet scheduling refers to the process that decides the order in which the packets need to be processed so as to have an optimal throughput. In addition to it, scheduling algorithm should provide bandwidth guarantees to flows and delay guarantees. A scheduling algorithm selects a Match (or) Matching M . A matching problem of an IQ switch can be matched to a bipartite graph G . A Bipartite graph $G = (V, E)$ consists of $2N$ vertices. IQ switch with input ports i corresponds to the left side vertices and output ports j corresponds to the right side vertices of a bipartite graph. A weight metric is associated with all the edges E of the graph G . A subset of admissible edges such that no two edges in M have a common vertex i.e., it never happens that two cells are transferred from input port i to output j thus satisfying the bandwidth restrictions imposed by the crossbar. A match can be maximum matching and *maximal matching*. A maximum matching is the largest size matching made on a given graph. A maximal matching is one no further edge can be added without modifying an already matched edge.

A maximum size matching is one that finds the maximum numbers of matches between input and output ports. This would provide the highest possible instantaneous throughput in a given timeslot. The complexity of solving MSM is $O(N^{5/2})$. It provides fairness, QoS support and good throughput under uniform and identically distributed (i.i.d) Bernoulli

traffic's. It can lead to instability under inadmissible traffic and they can even lead to starvation. There are several MSM algorithms like iSLIP (Shreedhar and Varghese, 1995), PIM (Muppala and Hamdi, 1999) iFAIR (Meckeown, 1999), FIRM (Anderson *et al.*, 1993) were proposed in the literature.

A Maximum Weight Matching is one that finds a matching M which maximizes the total weight $W(t) = \sum W_{ij}(t)$ at timeslot t provided a weights $W_{ij}(t)$ is attached to the edges of graph G . A MSM is a special case of MWM with all weights $W_{ij}(t) = 1$ at timeslot t . The complexity of solving MWM is $O(N^3)$ which infeasible to implement at high speed links. Some of the scheduling algorithms are LQF (Kumar *et al.*, 2004), OCF (Serpantos and Antoniadis, 2000) and LPF (Mekkittikul and McKeown, 1996). However, MWM has intrinsically high computation complexity that is translated into long resolution time and high hardware complexity. This makes it prohibitively expensive for a practical implementation with currently available technologies.

The Scheduling in IQ switch is also done by Random Matching methods. The basic idea of randomized scheduling is to select the best matching from a set of random matches. Randomized scheduling algorithms have been proposed for input queued switches in an attempt to simplify the scheduling problem. TASS (McKeown, 1995) is the basic randomized algorithm proposed by Tassiulas. A group of randomized algorithms including APSARA, LAURA and SERENA were proposed in the literature (Mekkittikul and McKeown, 1998).

4. QUEUING STRATEGIES

Queuing is a mechanism which enqueues and dequeue packets stored in the buffer according to some scheduling methods implemented in the scheduler. Queuing mainly depends on the size of the buffer and algorithm used to manage the queue. The Buffer size is an important perspective of queue management and causes packet loss when overflows and degrades throughput when underflows. It is the measure of QoS parameters which causes queuing delay and delay-variance in core routers and switches due to real-time applications. The first proposed rule of thumb define the buffer size is that to select the buffer size equal to the Bandwidth-Delay Product (BDP) of the outgoing link (Tassisulas, 1998).

In order to cope with transient congestion on links, backbone routers will often implement large buffers. Unfortunately, while these buffers are good for throughput, they can substantially increase latency and

cause TCP connections to behave very burstily during congestion. Williamizer and Song (1994) it is analyzed that optimal value of the buffer size is required to fully utilize the link capacity. It includes in (Avrachenkov *et al.*, 2005) link utilization increases and packet loss is reduced with the increase of buffer size until a certain threshold value. Further increases of the buffer size will increase link utilization but increases queuing delay which in turn affects throughput and incurs large end to end packet delay. Therefore, several recent works imposed on a smaller size buffers due to its practical benefits. Owing to the merits of small sized buffers, in this works VOQ of *finite size* is assumed. As discussed earlier the other aspect of Queuing is the algorithm used to schedule the packets.

The basic and widely adopted queuing scheme is First Come First Serve (FCFS) (Avrachenkov *et al.*, 2002) which services packets based on their arrival time. A Packet with a high priority with late arrival time must wait until a packet with a low priority with early arrival time is serviced. Thus, FCFS lacked with fairness in priority. The next scheme services packets based on their priority is called Priority Queuing (PQ). Since regardless of the packets arrival time high priority packet will be serviced and again low priority packets will suffer with high delay of backlogged in the queue itself. The optimal solution for priority packets is by providing fairness among competing packets is known as Fair Queuing. A Variant of fair queuing is Weighted Fair Queuing (WFQ) which provides better bandwidth guaranteed, bounded delay and weighted fair sharing at the packet level.

4.1. Weighted Fair Queuing (WFQ)

WFQ (Eric, 2009) was first introduced by Demers *et al.* (1989). Weighted Fair Queuing is sort-based packet scheduling algorithm to approximate GPS (McKeown, 2007). Generalized Processor Sharing (GPS) assumes that the input traffic is infinitely divisible and all sessions will be serviced at the same time. WFQ schedules packets by calculating a virtual finish time according to their arrival time, size and their associated weight. The scheduler calculates a virtual finish time upon a packet is arrived in the queue. The virtual finish time here represents time at which the same packet would finish to be served. WFQ arranges packets in the ascending order of the virtual finish time. It guaranteed that each flow gets its shares of bandwidth proportional to the assigned weights. A variant of WFQ was proposed in the literature with an aim to reduce the complexity in calculating the virtual finish time. Self-Clocked Fair Queueing (SCFQ) (Parekh and Gallager, 1993) calculates finish time by the packet currently being

transmitted. Start-time Fair Queueing (SFQ) (Golestani, 1994) uses the starting time of the packet currently in service. Among the packets already began in service Worst-Case Fair Weighted Queueing (WF²Q) (Goyal *et al.*, 1997) transmits the packet with lowest finish time. The proposed WFQ dynamically manages traffic flows and have better bandwidth guaranteed and *fairness* than proposed queueing algorithms in the literature. Fairness in scheduling is essential to protect flows from other misbehaving flows which is caused due to malfunction of software on routers or end-nodes and provide end-to-end service differentiation. It is found in (Bennet and Zhang, 1996; Stephens *et al.*, 1999). Fair scheduling is very essential in routers and switches.

4.2. LOW LATENCY QUEUEING (LLQ)

LLQ combines Priority Queue (PQ) with WFQ. LLQ guarantees the delay of real-time traffic and currently recommended for Voice over IP and streaming applications like video and audio data. Typically, there is one Priority Queue and some Weighted Fair Queues. Real-time traffic is queued to the priority queue and all other traffic is allocated to the WFQ priority queues. The LLQ scheduler initially and always checks for any packet in the highest priority queues, if any, then LLQ departs a packet from the highest priority flows. If there are no packets in the low latency queue, then normal scheduler logic applies to the other weighted fair queue. If two flows obtain the same priorities, the packets will depart according to CBWFQ policy. In this way, it reduces delay and jitter in real-time traffic. WFQ service the rest of the traffic. The priority queue is serviced before any of the weighted fair queues, thus allowing realtime traffic to be processed as fast as the network elements allow (CSI, 1999). It is analyzed in (NPT, 2003; Chen *et al.*, 2012) that LLQ was used to measure only queuing delay for real-time flows. The effects of “Under run buffer” in broadband networks is analyzed with LLQ in (Wu *et al.*, 2005).

4.3. FLOW DIAGRAM OF COMBINED WFQ AND LLQ

The **Fig. 2** illustrates the working of combined WFQ and LLQ.

4.4. PROPOSED FL²WFQ

Framed Low Latency Weighted Fair Queueing (FL²WFQ) works as follows. The priority of order high to low is assigned to VOQ_{1,n}, VOQ_{1,n-1} to VOQ_{1,1} in turn. The packet with highest priority is said to be Priority Queue and the remaining queues are said to be Weighted

Fair Queue. Each VOQ has its own priority and bandwidth share. The bandwidth equivalent of weight is assigned from high to low to VOQ_{1,n}, VOQ_{1,2} and VOQ_{1,1} in turn. The same set of priority and bandwidth allocation is also assigned to other input ports upto N. Let us aware there are N flows each associated with N VOQs according to its priority and bandwidth. A flow f is defined as a sequence of packets having a set of common characteristics such as combination of source and destination IP address, port number and possibly the application generating the packets. Each flow is assigned to a VOQ_{ij}. P^f(k) represents the kth packet in flow f. The following parameters are considered for calculating the packet Virtual finishing time:

P^f(k) = Packet k arrived at flow f and assigned to VOQ_{ij}

A^f(k) = Arrival time of packet k in flow f to VOQ_{ij}

L^f(k) = Length of packet k in flow f to VOQ_{ij}

W^f = Weight of flow f

V^f(j) = Virtual Finishing time of packet k in flow f to VOQ_{ij}

When the next packet k+1 arrives in VOQ_{ij} and its Virtual finishing time is computed as Equation (1):

$$V^f(k+1) \max\{V^f(k), S^f(k+1)\} + L^f(k) / W^f \quad (1)$$

S^f(k+1)-Priority of packet k+1 in VOQ_{ij}.

As time is slotted, at unit of time a packet P^f(k) is arrived to any VOQs. Whenever a packet is arrived it's weight of W^f and priority S^f is assigned to VOQ_{ij} of WFQ and PQ as discussed above. Upon arrival of first packet at time A^f(k) and it's Virtual Finish Time V^f(j) is calculated. Subsequent virtual finish time V^f(k+1) is calculated based on length of the packet L^f(k) and its assigned priority S^f(k+1). Length of the packet is fixed with size 53 in bytes. The packet with virtual finishing time and its assigned priority packets will be scheduled. The scenario can be better explained with a **Fig. 3**.

If the VOQs are served according to WFQ, then the order of packets would be sent as P¹(1), P¹(2), P²(1), P³(1), P²(2), P³(1). If FL²WFQ is used, the order in which packets are sent would be P³(1), P²(1), P²(2), P¹(1), P¹(2), P³(1) and as shown in **Fig. 4**. Since VOQ_{1,3} has the highest priority, P³(1) will be sent first. Therefore FL²WFQ could reduce the delay for high-priority packets and however a low priority packet would suffer with a serious delay. To overcome this dynamic sliding Frame is defined with a variable timeslot. A sliding frame consists of set of packets whose virtual finishing time lies within the virtual time interval.

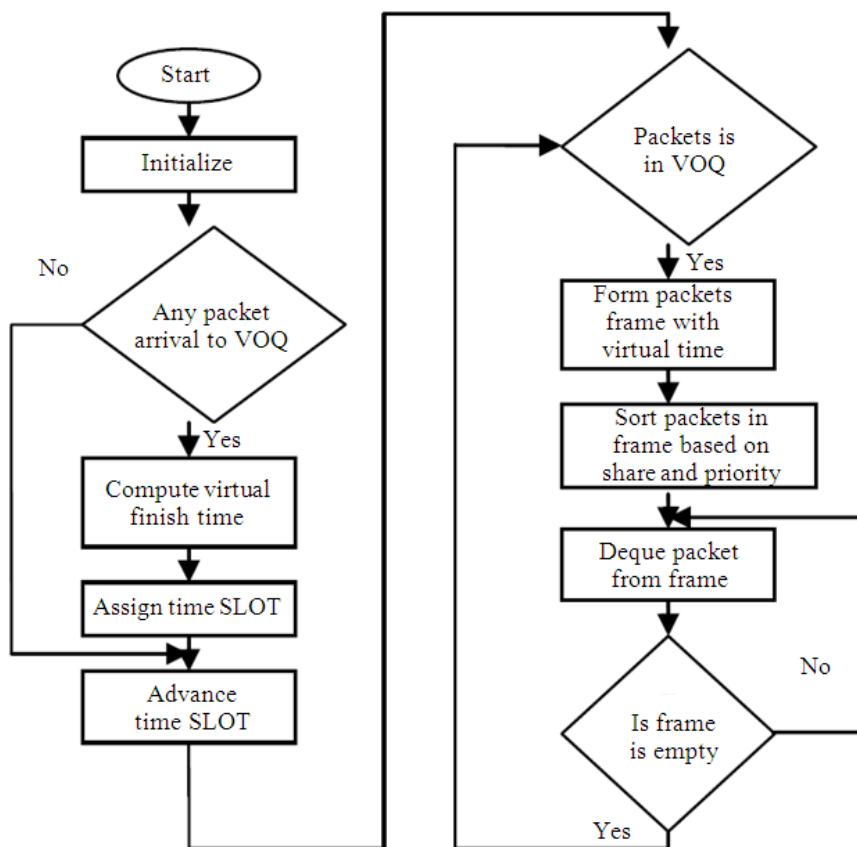


Fig. 2. Flow diagram of combined WFQ and LLQ

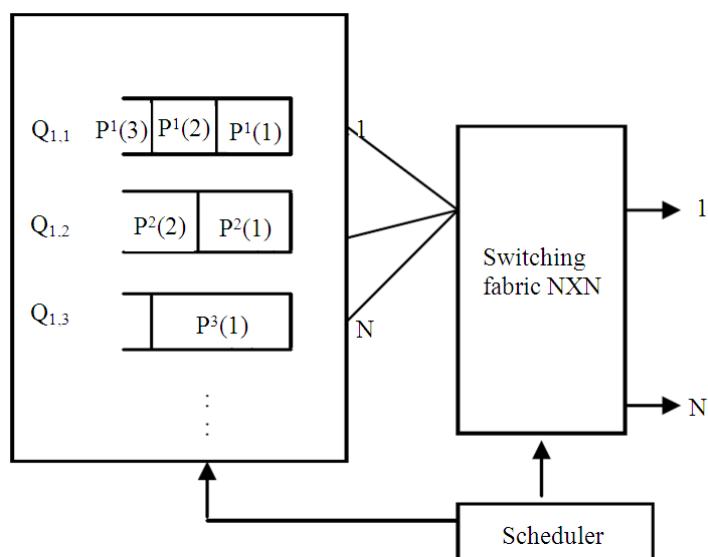


Fig. 3. Packets scheduled by FL^2WFQ

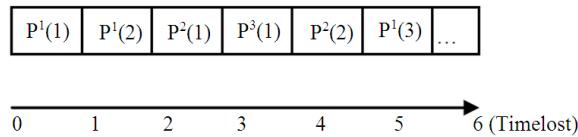


Fig. 4. Packet Virtual Finishing Time (low to high)

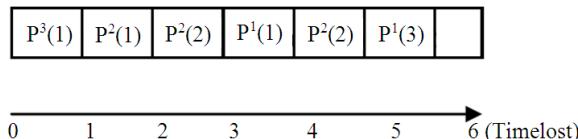


Fig. 5. Packet Departure from Frame

During each timeslot almost one packet is dequeued from any input port from the frame as given in **Fig. 5**. Before any packet is departed from the input port, IQ scheduler ensures that all packets inside the frame have the similar virtual finish time. According to fixed priority levels the relative packet service order inside the sliding frame has been changed. The packet with the highest priority is chosen for scheduling i.e., to transfer from an input port to an output port. Even though changing the packet scheduling order by priority levels inside the sliding frame will guarantee the bandwidth according to its weight and reduces delay for high priority packets. Also, low priority packets inside the frame would get its share and transmit early without having long queueing delay. The balance between priority (Low Latency) packets and share-driven (WFQ) packets were occupied inside the frame. In this study both bandwidth and delay are effectively controlled by an IQ switch. When more than two packets from any input ports compete to the same output port, then the conflicts is resolved according to CBWFQ policy. Then the packet with highest priority would transmit during that slot. The choice of the sliding frame size determines the effectiveness of scheduling. If the frame size is set to large enough, then the scheduler behaves like priority-based scheduler. In other case the frame size is set to zero and it is similar in operation to WFQ scheduler and the priority of the packet will not be taken into account. By setting dynamic frame size high priority, low share packet cannot suffer with delay and also, low priority, high share packet receives guarantee bandwidth.

5. SIMULATIONS

In this study simulation is carried by using open source Network Simulator (NS2) (www.isi.edu/nsnam/ns2). N×N internally non blocking

Input Queued switch with N input and N output ports as given in **Fig. 1** is considered in this works. For each input port i , there are N fixed-sized Virtual Output Queues $VOQ_{i,j}$, $1 < i, j \leq N$ and for N output buffers associated with N output ports. The cells arriving at input i and destined for output j are buffered in finite size VOQ_{ij} at timeslot t . A switch with packet arrival time of $A^f(k)$ with arrival rate of λ_{ij} and mean service rate of μ_{ij} to an input port at discrete interval of time t is assumed. Each arrival process A_{ij} is Poisson and is stationary and mutually independent. Let $A_{ij}(n)$ and $D_{ij}(n)$ be the cumulative number of cells that arrive at and departure from VOQ_{ij} respectively. The arrival process $A(n) = \{A_{ij}(n) = \sum A^f(k)\}$ satisfies the Strong Law of Large Numbers (SSLN) given below Equation (2):

$$\lim_{n \rightarrow \infty} \frac{A_{ij}(n)}{n} = \lambda_{ij} \quad (2)$$

The number of packets in VOQ_{ij} at time t is denoted by $Q_{ij}(t)$. The length $Q_{ij}(t)$ of VOQ_{ij} at time t is given by Equation (3):

$$Q_{ij}(t+1) = Q_{ij}(t) - S_{ij}(t) - S_{ij}(t) + A_{ij}(t+1) \quad (3)$$

And it is used in the calculation of packet delay in a queue and it is represented as C_q . $S_{ij}(t)$ refers to speed of the switching fabric in which atmost one cell is transferred from an input port to an output port and it assume to be one. The input traffic is admissible or uniform if it satisfies the following constraints Equation 4 and 5:

$$\sum_{i=1}^N \lambda_{ij} \leq 1 \quad (4)$$

$$\sum_{j=1}^N \lambda_{ij} \leq 1 \quad (5)$$

where, N is the number of input and output ports.

The output queue follows G/D/1 in (McKeown, 1997) and M/D/1 is followed in IQ switches in (McKeown *et al.*, 1997). In this study, we assume each input queue is an M/G/1/K with service time equal to packet waiting in HoL. Under the assumption of uniform traffic and well structured finite sized VOQs and output queues and the packets arrived at HoL of the VOQ_{ij} are scheduled by FL²WFQ. The FL²WFQ works as discussed in section 4.4. The Fairness and Starvation are the two important properties of the IQ switch that are resolved by combining WFQ and LLQ queueing. WFQ guaranteed Fairness and ensure that all the VOQs get their share and their turn to transfer the cells. The fairness of VOQs is verified by assigning percentage of weight equivalent to bandwidth share according to WFQ policy. A high share to low share is fixed starting from flow of VOQ₁₁ to VOQ_{IN} in turn. Similarly a fixed high to low priority is assigned from a flow of VOQ_{1,N} to VOQ_{1,1} in turn to queue video, audio and data packets respectively with an aim to ensure Low Latency Queueing. These packets are first classified according to CBWFQ policy and then assigned to the corresponding VOQ. Virtual Finish time of each packet is calculated by using the Equation (1). A snap shot of packets from each VOQ is taken for a varying timeslot and dynamic frame is formed and then scheduled by FL²WFQ algorithm. The performance of IQ switch's VOQ are then analyzed in terms of throughput and average delay by varying finite buffer size and switch size.

6. RESULTS

The cell size in flow L^f(k) considered in this work is of 53 bytes (according to ATM terminology) consisting of 48 bytes of data and 5 bytes of control information. The packets and cells are used synonymously. We

assume the following parameters and their values in **Table 1** for IQ switch in our study:

| | |
|-------------------|---|
| N×N | = Number of input and output ports |
| VOQ _{ij} | = VOQ hold packet arriving to input i and destined for output j |
| C _q | = Current capacity of the VOQ _{ij} |
| B | = Buffer size of VOQ _{ij} |
| W ^f | = Weight of VOQ _{ij} |
| S ^f | = Priority of flow to VOQ _{ij} |

Our analysis is first initiated with switch size N = 4 and with each VOQ_{ij} is set to 200 cells. The arrival time of packet is calculated with assumption of 0 arrival time of first packet to any VOQ_{ij}. The bandwidth allocation and priority of VOQ_{ij} is set according to the values given in the **Table 1**. The Virtual finishing time of each arriving packet is then computed. The switch size considered is N = 4, N = 8 and the offered load varies from 0.1 to 0.8 with an Erlang distribution.

A throughput graph for switch size N = 4 is plotted for varying buffer sizes as shown in **Fig. 6**. It is inferred from the graph that the throughput of an IQ switch increases linearly with respect to time. Throughput is proportional to the size of the VOQ. Buffer and VOQ are used interchangeably. Increasing the size of VOQ accommodate more number of cells which in turn maximizes throughput to considerable value. This assessment is followed by comparing performance of packet delay with varying buffer sizes and switch size. The measurement of average packet delay is plotted as graph in **Fig. 7**. In this graph it is found that delay of packet decreases by increasing the VOQ's size. Both are inversely proportional to each other. This realizes that buffer size is an important factor to maximize the throughput and minimize packet delay and thereby reducing less number of packet losses.

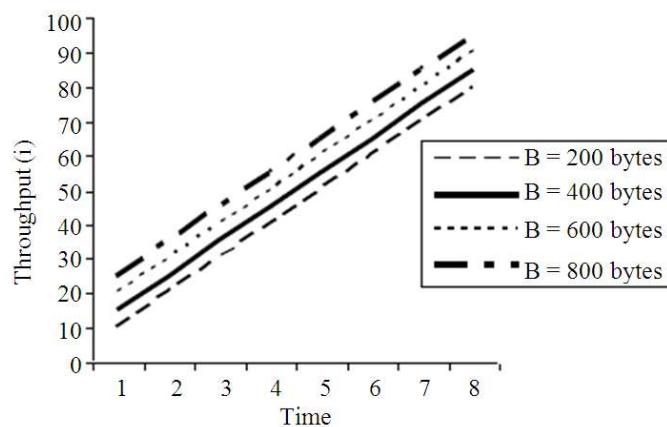
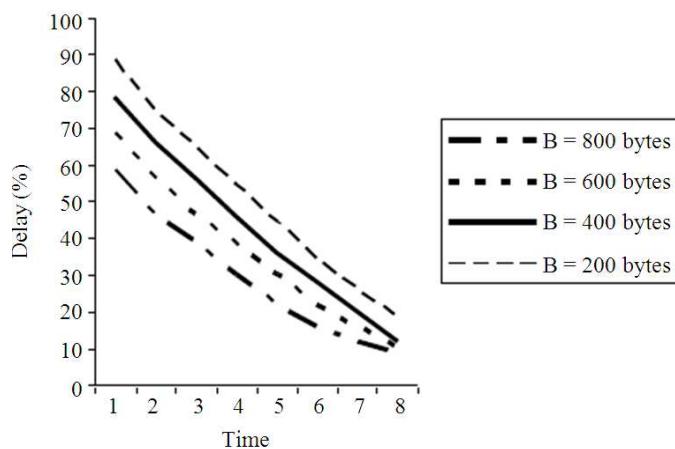


Fig. 6. Throughput Vs time (N = 4)

**Fig. 7.** Delay Vs time (N = 4)**Table 1.** Parameter values

| | |
|--|----------------------------------|
| N×N | 4,8 input and output ports |
| Q_{max} | 1000 packets |
| B | 200, 400, 600, 800, 1000 packets |
| $W^f(VOQ_{1,1} \text{ to } VOQ_{1,N})$ | 40, 30, 20, 10% |
| $S^f(VOQ_{1,N} \text{ to } VOQ_{1,1})$ | 1, 2, 3, 4 |

7. CONCLUSION

In this study, the novel architecture of IQ a new high-performance switch that achieves performance close to that of OQ have been presented. A measure of throughput and average packet delay was analyzed by varying buffer size. The simulation was done using NS2 and the results of the both fairness and starvation was studied by combining both WFQ and LLQ. The previous scheduling algorithm proposed in the literature was run with an implicit assumption of infinite buffer space with no queue management. This paper was analyzed with the alternate scheduling algorithm with finite buffer space and queue management. The result shows that the proposed works perform well by increasing the throughput of an Input-Queued switch with minimum delay when scheduled by FL²WFQ with different buffer sizes and thereby discarding less number of packets and reducing delay. In future, the proposed scheduling algorithm is to be simulated in IQ in order to measure packet losses in finite VOQs.

8. REFERENCES

Anderson, T.E., S.S. Owicky, J.B. Saxe and C.P. Thacker, 1993. High-speed switch scheduling for local-area networks. ACM Trans. Comput. Syst., 11: 319-352. DOI: 10.1145/161541.161736

- ATM, 1994. ATM User-Network Interface Specification: Version 3.1. 1st Edn., Prentice Hall PTR, ISBN-10: 013393828X, pp: 396.
- Avrachenkov, K., U. Ayesta and A. Piunovskiy, 2005. Optimal choice of the buffer size in the Internet routers. Proceedings of the 44th IEEE European Control Conference on Decision and Control, Dec. 12-15, IEEE Xplore Press, pp: 1143-1148. DOI: 10.1109/CDC.2005.1582312
- Avrachenkov, K.E, U. Ayesta, E. Altman, P. Nainy and C. Barakat, 2002. The Effect of router buffer size on the TCP performance. Proceedings of the LONIIS Workshop on Telecommunication Networks and Teletraffic Theory, pp: 116-121.
- Awedeh, R.Y. and H.T. Mouftah, 1995. Survey of ATM switches architectures. Comput. Netw. ISDN Syst., 27: 1567-613. DOI: 10.1016/0169-7552(94)00081-4
- Bennett, J.C.R. and H. Zhang, 1996. WF²Q: Worst-Case fair weighted fair queueing. Proceedings of the IEEE 15th Annual Joint Conference of the IEEE Computer Societies Networking the Next Generation, Mar. 24-28, IEEE Xplore Press, San Francisco, CA., pp: 120-128. DOI: 10.1109/INFCOM.1996.497885
- Chen, L.H., I. Ming, J. Tzong and G. Huey, 2012. Credit-based low latency packet scheduling algorithm for real-time applications. Proceeding of the IEEE International Conference on Communications Networks and Satellite, Jul. 12-14, IEEE Xplore Press, Bali, pp: 12-14. DOI: 10.1109/ComNetSat.2012.6380768

- CSI, 1999. Cisco 121016 gigabit switch router. Applications Note, Cisco Systems Inc.
- Demers, A., S. Keshan and Shenker, 1989. Analysis and Simulation of a fair queueing algorithm. Proceedings of the Symposium on Communications Architecture and Protocols, Sept. 25-27, ACM Press, New York, USA., pp: 1-12. DOI: 10.1145/75246.75248
- Demers, A., Srinivasan, L. Zhang, Keshav and S. Shenke, 1989. Weighted fair queuing.
- Eric, 2009. Scheduling of input queued switch. IEEE.
- Golestani, S.R., 1994. A Self-Clocked fair queueing scheme for broadband applications. Proceedings of the 13th Networking for Global Communications, Jun. 12-16, IEEE Xplore Press, Toronto, Ont., pp: 636-646. DOI: 10.1109/INFCOM.1994.337677
- Goyal, P., H.M. Vin and H. Cheng, 1997. Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks. IEEE Trans. Network., 5: 690-704. DOI: 10.1109/90.649569
- Karol, M.J, M.G. Hiluchyi and S.P. Morgan, 1987. Input versus output queuing on a space-division switches. IEEE Trans. Commun., 35: 1347-1356. DOI: 10.1109/TCOM.1987.1096719
- Kumar, N., Rong, Pan and Devarat, 2004. Fair scheduling in input-queued switches under inadmissible traffic. Proceedings of the Global Telecommunications Conference, Nov. 29-Dec. 3, IEEE Xplore Press, pp: 1713-1717. DOI: 10.1109/GLOCOM.2004.1378274
- McKeown, 2007. A Low Latency Scheduling of IQ.
- McKeown, N., 1995. Scheduling algorithms for inputqueued cell switches. Ph.D Thesis, University of California at Berkeley.
- McKeown, N., 1997. A fast switched backplane for a gigabit switched router. Department of Electrical Engineering,
- McKeown, N., I. Keslassy and G. Appenzeller, 2004. Sizing router buffers. Proceedings of the 2004 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, (PCC '04), ACM Press, New York, USA., pp: 281-292. DOI: 10.1145/1015467.1015499
- McKeown, N., M. Izzard, A. Mekkittikul and W. Ellersick, 1997. Tiny tera: A packet switch core. IEEE Micro, 17: 26-33. DOI: 10.1109/40.566194
- McKeown, N., 1999. Islip scheduling algorithm for input-queued switches. Trans. Netw., 7: 188-210. DOI: 10.1109/90.769767
- Mekkittikul, A. and N. McKeown, 1998. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 29-Apr. 2, IEEE Xplore Press, San Francisco, CA., pp: 792-799. DOI: 10.1109/INFCOM.1998.665102
- Mekkittikul, A. and N. McKeown, 1996. A starvationfree algorithm for achieving 100% throughput in an input-queued switch. Department of Electrical Engineering,
- Muppala, G.N.K. and M. Hamdi, 1999. Analysis of nonblocking ATM switches with multiple input queues. IEEE/ACM Trans. Netw., 7: 60-74. DOI: 10.1109/90.759320
- NPT, 2003. White paper authored by foursticks CTO Alan noble for equator. One Pte Ltd., Network Performance Technology.
- Parekh, A.K. and R.G. Gallager, 1993. A generalized processor sharing approach to flow control in integrated services networks: The single node-case. ACM/IEEE Trans. Netw., 1: 334-357. DOI: 10.1109/90.234856
- Serpanos, D.N. and P. Antoniadis, 2000. FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues. Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, (CCS '00), IEEE Xplore Press, Tel Aviv, pp: 548-555. DOI: 10.1109/INFCOM.2000.832228
- Shreedhar, M. and G. Varghese, 1995. Efficient fair queuing using deficit round robin. Proceedings of the Symposium Conference on Applications Technologies, Architectures and Protocols for Computer Communication, Aug. 28-Sept. 01, ACM Press, Cambridge, MA, USA., pp: 231-242. DOI: 10.1145/217382.217453
- Sommers, J., P. Barford, M. Duffield and A. Ron, 2005. Improving accuracy in end-to-end packet loss measurement. Proceedings of the 2005 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Aug. 22-26, ACM Press, Philadelphia, PA, USA., pp: 157-1680. DOI: 10.1145/1080091.1080111
- Stephens, D.C., J.C.R. Bennett and H. Zhang, 1999. Implementing scheduling algorithms in high-speed networks. IEEE J. Selected Areas Commun., 17: 1145-1158. DOI: 10.1109/49.772449
- Awdeha, R.Y. and H.T. Mouftah, 1995. Survey of ATM switch architectures. Computer networks and ISDN systems, 27: 1567-1613. DOI: 10.1016/0169-7552(94)00081-4

Tassisulas, L., 1998. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 29- Apr. 2, IEEE Xplore Press, San Francisco, CA., pp: 533-539. DOI: 10.1109/INFCOM.1998.665071

Villiamizer, C. and S. Song, 1994. High performance TCP in the ASNET ACM SIGCOM computer communication review, 24: 45- 60.
Wu, E.H.K., I. Ming, Hsieh and H.T. Lai, 2005. A novel low latency packet scheduling scheme for broadband networks. Springer-LNCS 3768, pp 1015-1026.