Performance analysis of Crosspoint Queued Crossbar Switch with Weighted Round Robin scheduling algorithm under Unbalanced Bursty Traffic

Soko Divanovic, Milutin Radonjic, Igor Radusinovic
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
e-mail: soko.divanovic@gmail.com, {m.radonjic, igorr}@ieee.org

Gordana Gardasevic
Faculty of Electrical Engineering
University of Banja Luka
Banja Luka, Bosnia and Herzegovina
e-mail: gordanagardasevic@gmail.com

Abstract—The possibility of enabling Quality of Service (QoS) in crosspoint-queued packet switches has been presented in this paper. For this purpose, the weighted round robin algorithm was simulated and analyzed. The packet delay represents one of the most important parameters in modern networks. Therefore, besides the throughput, we have been investigating the delay that crosspoint queued switch introduce, under bursty unbalanced traffic. Along with the average delay, the truncated maximum cell delay is analyzed. Results show that the weighted round robin scheduling algorithm achieves throughput similar to output-queued switch, under unbalanced bursty traffic, while maintaining low cell delay.

Keywords-CQ switch; QoS; WRR; throughput; latency;

I. INTRODUCTION

Over the last decade, the development of modern Internet services, such as Internet Protocol Television (IPTV), cloud computing and social networking has led to exponential growth of Internet traffic. Significant progress in developing smartphones, tablets and other portable electronic devices, enabled much easier and accessible use of these services for end user, while imposing even greater traffic intensity. These facts put pressure on the backbone networks. In order to support and accommodate ever-growing traffic, the development of high performance packet switches (and routers) capable of handling such traffic is of high importance. Telecommunication networks require packet switches that are able to provide performance guarantees and to assure desired QoS to end users [1, 2]. Therefore, switches should be able to give guaranteed throughput, latency and flow separation.

A high throughput is usually the most important parameter to achieve. This is due to fact that high throughput maximize the utilization of long-haul links. In the near future, the implementation of commercial switches with throughput of hundreds Tb/s is expected. The second parameter that is desirable, especially for real time applications, is the guaranteed latency. For delay sensitive traffic such as video on demand (VoD), online gaming, IPTV and so on, this parameter is much more important than throughput. Finally, the flow separation is third desirable parameter, because it allows the network operator to provide different QoS parameters to different flows.

In order to achieve these requirements, many types of switching architectures have been proposed. Depending on the buffers position, there are several crossbar switching architectures: input-queued (IQ), output-queued (OQ), combined-input-crosspoint-queued (CICQ), combined-input-output-queued (CIOQ) and crosspoint-queued (CQ).

It is well known that OQ switch has superior performances. On the other hand, OQ switching fabric must operate with speedup that is equal to switch size (the number of ports N) and the memory bandwidth at each output buffer must be N times higher, in the worst case, than the line rate of the switch ports [3]. For a long time, IQ switch was attractive solution due to its simplicity. However, this solution has a major problem with head of line (HOL) blocking, which limits the throughput to 58.6% [3]. The HOL blocking problem can be solved with Virtual Output Queuing (VOQ) [3]. Various scheduling algorithms have been proposed that can achieve 100% throughput. However, many of these algorithms are very complex, and therefore increase overall latency of the switch.

A combination of IQ and OQ switches (CIOQ switch) has been proposed to support high throughput with minimal delay [3], but suffers from similar problems as an IQ switch. HOL blocking and complex scheduling is minimized with CICQ switch that has VOQ buffers at inputs and crosspoint buffers within crossbar fabric [3]. Scheduling is much easier with this switch since input and output scheduling can be performed separately. However, the CICQ switch needs permanent control communication between arbiter and linecards. Considering that linecards in modern switches can be quite distant from the switching fabric, long round-trip-time (RTT) in credit based mechanism can occur, which leads to significant performance degradation [4].

Due to the technological advancements that enabled implementation of large crosspoint buffers together with crossbar fabric on the single chip, crosspoint queued switches has recently drawn attention of research community [5]. A CQ switch does not require switching and/or memory speedup, avoids HOL blocking (because there is no input buffers) and does not require control communication between linecards and switching fabric. This architecture is also interesting because it is modification of the theoretical superior OQ switch. Therefore, OQ switch scheduling algorithms developed to provide QoS services can be easily implemented in the CQ switch.
All these benefits motivated us to observe scheduling algorithms, with and without QoS support, in order to analyze possibility to enable QoS services in the CQ switch. The throughput, average latency, and maximum latency of different granularity are observed and analyzed. We will show that it is possible to achieve desired performances, regarding the throughput and latency that CQ switch invokes, using scheduling algorithm with QoS support. These results are confirmed by comparison to the results achieved for QQQ switch.

This paper is organized as follows. The CQ switch and simulated traffic model are presented in Section 2. The methodology used in our work is explained in Section 3. Performance analysis is presented in Section 4. Conclusions are drawn in Section 5.

II. CROSSPOINT QUEUED CROSSBAR SWITCH

A. CQ switch architecture

The CQ switch has \( N \) inputs, \( N \) outputs, and \( N^2 \) crosspoint buffers implemented within the crossbar fabric [5, 6], as shown in Fig. 1.

In the CQ switch, all the buffering is moved from the linecards to the switching fabric. Therefore, the HOL blocking is eliminated and the need for control communication between linecards and switching fabric is avoided, which reduce switch implementation complexity. Incoming cells that are arriving at input \( i \), and are addressed to the output \( j \), are simply forwarded directly to the crosspoint buffer \( B_{i,j} \). The cell is queued at the crosspoint buffer if that buffer is not full; otherwise, the cell is discarded. In each time slot, the output scheduler chooses one of the non-empty crosspoint buffers from the common output line and forwards its head-of-line cell to an output linecard. The selection of buffers that will be served is performed by some scheduling algorithm. In this process, output ports are independent from each other.

The size of crosspoint buffers has a large impact on the CQ switch performance. However, a crosspoint buffer size is limited by the amount of available memory on the chip, and by the size of the switch. Considering that \( N \times N \) CQ switch has \( N^2 \) crosspoint buffers, the available memory for each buffer is \( N^2 \) times smaller than total on-chip memory. This used to be a problem, but with the advancement of the technology, the implementation of large crosspoint buffers became possible [5].

B. Analyzed scheduling algorithms

In this paper, we have analyzed the performance of CQ switch with following scheduling algorithms: Longest Queue First (LQF) [6], Frame-Based Round Robin (FBRR) [6] and Weighted Round Robin (WRR) [7]. The special focus will be paid to WRR scheduling algorithm, because it is an algorithm with QoS support, and according to our knowledge it was not analyzed within CQ switch in available literature.

In our previous work, we compared many scheduling algorithms and showed that LQF algorithm provides better throughput than RR based algorithms, while FBRR algorithm invokes the lowest latency [8]. Therefore, in order to properly evaluate performances of WRR scheduling algorithm, we compared it with LQF algorithm (to evaluate throughput) and FBRR algorithm (to evaluate latency).

LQF algorithm services buffer with the highest occupancy on the particular output line, in each time slot.

FBRR algorithm services the same buffer until it departs all cells that were in the buffer at the time when its service started. The buffer occupancy at the time it starts to be served is called a “frame.” After departure of all cells from the frame, the next buffer on the same output line is serviced in the circular, round robin order (the service is always performed in the same order determined by the buffer position).

WRR algorithm services buffers on the particular output line \( j \) according to the weights of corresponding flows. The weight of a flow \( i, j \) is the number of cells that WRR scheduler departs from buffer \( B_{i,j} \) in one frame, before moving to the next buffer in a round robin order. The frame size is equal to the sum of all weights for buffers on the same output line. The minimum weight is 1, and the maximum weight is equal to crosspoint buffer occupancy. There are two different ways for the assignment of these weights: dynamically and offline.

The dynamical assignment requires continuous traffic monitoring and permanent weights adjustment to the traffic pattern. In offline assignment, an offline analysis of traffic is conducted and the matrix of weight factors (numbers from 0 to 1) is calculated. By dividing all elements in column \( j \) of this matrix with minimal element of that column, minimal weights for each flow are derived. This procedure is performed for each column and weight matrix is obtained.

In our simulations we used offline approach to derive weight of each flow. Moreover, every weight is multiplied by 10 (scaling minimal weight to 10), which gives more time for each flow to send their cells.

III. SIMULATION MODEL

In our simulations, we assume that incoming packets have fixed size length (referred to as cells). The time is divided into equal time slots, where each slot’s length corresponds to the requirement for transferring one cell.

The arrival traffic is bursty and modeled with IBP traffic model [6]. Each input port is described by the two state ON-
OFF model, as shown in Fig. 2, where both active and idle periods are geometrically distributed.

![ON-OFF traffic model](image)

Figure 2. ON-OFF traffic model

IBP traffic is modeled with average burst size ($Bs$), which represents the average duration of active (ON) state. As shown in Fig. 2, if the input port is in the ON state, it will stay in that state with probability $1-a$, or switch to the OFF state with probability $a$. Similarly, if the input port is in the OFF state, it will stay in that state with probability $1-b$, or switch to the ON state with probability $b$. These probabilities are derived directly from the average burst size ($Bs$) and input load ($p$), and are given by the following equations:

$$a = \frac{1}{Bs}$$

$$b = \frac{pa}{1 + pa - p}$$

To properly evaluate WRR algorithm performance, the incoming cells are distributed non-uniformly to all output ports, according to the unbalanced probability ($w$) parameter. This is performed because the uniform distribution produces weights of flows that are mutually almost equal and WRR algorithm behaves like ordinary RR algorithm. The traffic load from input $i$ to output $j$ is derived from equation (3) [6]:

$$p_{i,j} = \begin{cases} p \left( \frac{w+1-w}{N} \right), & \text{if } i = j \\ p \frac{1-w}{N}, & \text{otherwise} \end{cases}$$

The parameter $p$ represents the input load and $N$ represents the switch size. The aggregate offered load for output port $j$, considering all input ports, is $p$. The offered traffic is uniform when $w=0$ and is completely directional from port $i$ to port $j$ when $w=1$. The latter means that all traffic from input port $i$ is addressed only for output port $j$, where $i=j$.

IV. PERFORMANCE ANALYSIS

In our previous work [8] - [10], standard parameters such as throughput, average cell delay and cell loss probability, were analyzed for CQ switch with LQF and FBRR scheduling algorithms. However, in order to obtain better simulation analysis, in this paper we performed additional measurements of truncated maximum packet delay. We measured maximum cell delay of fraction ($v$) of all accepted cells. Measurements for different delay depths (level of granularity) were conducted, whereby $v$ took values of 90%, 99%, 99.9%, and 99.99%. This analysis provides additional information and gives better insight in the CQ switch performance, for different scheduling algorithms, in terms of delay guarantees. In QoS performance guarantees analysis, the fraction of time is very important. The maximum cell delay for different fraction $v$ is given, in the case of long enough buffer length, in order to maximize throughput for all algorithms. Thus, the fair comparison of results can be achieved.

All simulations are performed for ten million time slots on 32×32 CQ switch with buffer lengths of 128, 256 and 512. Buffer lengths are given by the number of cells that can be accommodated. Simulations were performed for average burst size $Bs=32$, different values of input load ($p = 0.7$, 0.8, 0.9, and 0.95), and unbalanced probability ($w = 0$, 0.3, 0.5, and 0.7). Finally, these results are compared with those achieved for 32×32 OQ switch under the same traffic conditions. In order to perform the correct analysis, CQ and OQ switch must have an equal amount of available memory dedicated to buffers. Considering that OQ switch has only one buffer for each output port, and CQ switch has 32 buffers for each output port, in order to have the same memory amount, OQ switch buffers are 32 times longer than buffers in crosspoints of CQ switch.

A. Throughput

In order to investigate the impact of unbalanced traffic on CQ switch, the throughput of a CQ switch is observed for different values of input load and unbalanced probability. Fig. 3 depicts the throughput of CQ switch, for different scheduling algorithms and buffer lengths, as a function of unbalanced probability. Together with these results, the throughput of an OQ switch is presented.

As can be seen from Fig. 3, for buffer lengths $L=128$ the throughput depends on used scheduling algorithm. LQF algorithm, as expected, gives the best throughput, nearly identical to OQ switch, while the WRR algorithm gives a similar result (difference is within 1.5%), which is much better than the one achieved with FBRR algorithm. Interesting results occur when unbalanced probability

![Throughput as a function of unbalanced probability, for $p=0.9$ and buffer lengths $L=128$ and $L=256$](image)
increases. In our previous work [6] we showed that the worst performance for each observed algorithm is achieved when the unbalanced probability is \( w = 0.5 \).

However, the throughput of WRR algorithm actually slightly improves with the increase of the unbalanced probability. This can be explained by the fact that WRR algorithm is implemented in a way that most loaded flows (i.e. corresponding buffers) gets most time to be serviced. Even for a worst case of unbalanced traffic, where unbalanced probability is equal to 0.5, the throughput of WRR algorithm is very good and less than 1% smaller than throughput of LQF algorithm.

From Fig. 3 one can also notice that significant improvement is achieved when buffer lengths are increased to \( L = 256 \). In that case, WRR algorithm obtains throughput that is nearly identical to throughput achieved using LQF scheduling algorithm. In addition, it can be seen from Fig. 3 that for \( w = 0.5 \) (considered as the worst case of unbalanced probability), WRR algorithm with buffer lengths \( L = 128 \) provides better throughput than FBRR algorithm with twice longer buffers (i.e. \( L = 256 \)).

Considering that the largest difference between used algorithms is around \( w = 0.5 \), further results for CQ switch performance will be given for this value of unbalanced probability.

By increasing crosspoint buffer lengths, the difference between used algorithms further decreases. This is shown in Fig. 4, where the throughput as a function of input load, for unbalanced probability \( w = 0.5 \) and buffer length \( L = 512 \), is presented. From Fig. 4 one can conclude that very good throughput is achieved, for any simulated scheduling algorithm, with buffer length of \( L = 512 \), even for the worst case of traffic unbalance (\( w = 0.5 \)). These results will be further confirmed by the cell loss probability analysis, given in the next subsection.

B. Cell loss probability

Cell loss probability as a function of input load is presented in Fig. 5. These results are given for unbalanced probability \( w = 0.5 \), and crosspoint buffer lengths \( L = 512 \). Under these conditions, certain algorithms have not discarded any cell during the simulations. Therefore, their cell loss probability equals to zero. Considering that zero values cannot be drawn in logarithmic scale, bars that missing from Fig. 5 represent these cases.

For input loads \( \rho \leq 0.7 \), the CQ switch with WRR scheduling algorithm does not have any lost cell, as well as the CQ switch with LQF algorithm and OQ switch. With the increase of input load, the cell loss probability increases, but remains within the value of \( 10^{-4} \). Cell loss probability of \( 10^{-5} \), when the input load is equal to \( \rho = 0.5 \), is considered acceptable in [11]. Therefore, results achieved in our analysis, where the cell loss probability is analyzed for much heavier traffic, are quite satisfactory.

C. Cell delay

The throughput analysis of a CQ switch indicates that very good throughput can be achieved, for particular traffic model and given parameters. Similar results are obtained for input loads of \( \rho = 0.7 \) and \( \rho = 0.8 \), regardless of observed scheduling algorithm. Therefore, cell delay analysis will be performed for these input loads.

Like within throughput analysis, unbalanced IBP traffic (\( w = 0.5, B_s = 32 \)) is used in order to observe performance of CQ switch with crosspoint buffer lengths of \( L = 512 \). Beside average cell latency, maximum cell latency of different granularity (\( \nu \)) is analyzed. These results are shown in Fig. 6 and Fig. 7.
By observing these diagrams, one can notice that average cell latency is almost the same for all observed scheduling algorithms. Precisely, the difference in average cell latency for different scheduling algorithms remains within two time slots. These results suggest the need for further analysis by observing maximum cell latency. Namely, this analysis allows identification of differences between the observed scheduling algorithms and better analysis of latency that switch inserts.

Latency analysis of every accepted cell shows that most cells have relatively small latency. If we observe the maximum latency for \( v < 90\% \) of accepted cells, the difference between scheduling algorithms is almost unnoticeable. With increase of granularity, the difference becomes obvious, as shown in Fig. 6 and Fig. 7. LQF scheduling algorithm shows the worst performance, while OQ switch has best results, regarding the maximum latency. This is expected considering the fact that OQ switch is theoretically superior switch. It is used to represent a referent level of performances, which other switches tend to achieve.

From Fig. 6 and Fig. 7, one can see that even when the level of granularity is \( v = 99.9\% \), WRR scheduling algorithm has maximum latency nearly identical to FBRR. Only when the granularity is higher, a noticeable difference between these algorithms occurs. This is not the case for LQF algorithm, where maximum latency is significantly higher.

If we take into account that our previous research pointed FBRR scheduling algorithm as the one that provides lowest latency, results obtained and presented in this section for WRR algorithm can be considered as very satisfactory. We can actually underline that this algorithm provides very small latency (close to OQ switch), while providing very good throughput.

V. CONCLUSION

In this paper, analysis of throughput and latency that CQ switch imposes for different scheduling algorithms is presented. Special focus has been paid to performance of WRR algorithm. We also presented one more way to analyze cell latency. By analyzing different fractions of accepted cells, various granularity of maximum cell latency is obtained. This provided additional information in our switch performance analysis. We showed that WRR algorithm achieves very good performance, regarding the throughput and cell latency, by comparing it to other scheduling algorithms and OQ switch. In addition, we showed that WRR scheduling algorithm improves the throughput when unbalanced probability increases, which is not the feature of the others observed algorithms. This makes WRR algorithm very suitable for unbalanced traffic patterns.

Future analysis will be focused on the influence of frame size to WRR scheduling algorithm performance. Besides, we intend to develop and implement some variations of WRR and other scheduling algorithms capable of providing dynamic guarantees for QoS services, and perform the switch analysis with these algorithms.

ACKNOWLEDGMENT

This work is partly supported by the Ministry of Science of Montenegro and Ministry of Education and Science of Bosnia and Herzegovina under the bilateral Project “Performance analysis of CQ packet switch from aspect of QoS guarantees”, and by the Ministry of Science of Montenegro under grant 01-451/2012 (FIRMONT).

REFERENCES


