*2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)*
***IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore***
*DOI: 10.7763/IPCSIT.2012.V53.No.1.02*

# CFSB: A Load Balanced Switch Architecture with O (1) Complexity

Shen Zhijun[+], Zeng Huashen and Gao Zhijiang

School of Information Science and Technology, Southwest Jiaotong University
Chengdu, Sichuan, China 610031

**Abstract.** In allusion to the limited ability of the Byte-Focal switch[1] under high-speed switching environment, this paper introduces a load balanced switch architecture called CFSB (Combine Flow Splitter with Byte-Focal), which uses the flow splitter to enable packets to be delivered without disordering. The CFSB switch has O (1) complexity and a better performance than Byte-Focal.

**Keywords:** Switch fabric;packet switching, load-balanced switch, flow splitter.

## 1. Introduction

With the rapid growth of Internet traffic and the increasing line rates, the relay system has gradually become the bottleneck of the network. Moreover, the traffic over the Internet has been confirmed to be bursty[2],and our simulation shows that the performance of traditional single stage switch like iSLIP is rapidly deteriorated under bursty traffic. In view of the situation described above, designing high speed switch architectures with good adaptability to bursty traffic become the hot topic in the field of switching technology.

For the O(1) complexity of crossbar connection pattern, the Load-Balanced Birkhoff-von Neumann switch architecture(LB-BvN), introduced by C.S. Chang *et al.* [3, 4]can meet the above-mentioned requirements well and thus makes it an appealing architecture to study. However, the main drawback of the load-balanced switch is that packets may be out-of-sequence. Several schemes have been proposed to solve this problem. All the existing schemes can be categorized into two approaches. The first approach is to prevent packets from being received out-of-sequence at the output ports, such as FFF[5] and Mailbox switch[6]. The second approach is to limit out-of-sequence packet to an upper bound and then using a re-sequencing buffer (RB) at the output port to reorder the packets, fox example, FCFS(First Come First Served)[7], EDF(Earliest Deadline First)[7], EDF-3DQ, FOFF(Full Ordered Frame First)[8] and the Byte-Focal switch[1].

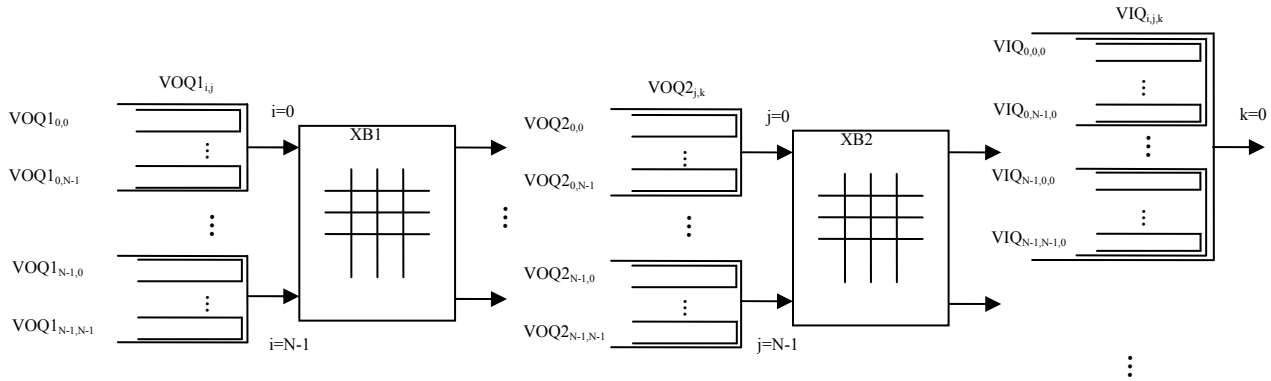


Fig. 1: The Byte-Focal switch architecture

[+] Corresponding author.
*E-mail address*: shensljx@sina.com.

The FCFS[7] uses the jitter control to restore packet's arriving order before the central buffer by delaying every packet to N(N-1) timeslots. Obviously, this scheme increases the average delay. The EDF[7] assigns a deadline to every packet to determine its departure time. Packets in the central buffers are served based on their deadline values. Searching the smallest timestamp in each VOQ is complex and costly. Although the complexity is reduced to O(N) in EDF-3DQ[5], it is equally hard to be accomplished under high-speed switching network. The FFF[5] needs a large amount of communication overhead flowing between the linecards to search for full frames. In FOFF[8], any partial frame will waste up to N timeslots regardless of its partial frame size. This increases the average delay, especially in light-load. The Mailbox switch can only achieve 75% throughput and the modified version achieves 95% throughput at the cost of additional re-sequencing buffer.

The key property of Byte-Focal[1] is that the DTS (Dynamic Threshold Scheme) ensures packets from the same flow to be distributed to the central buffers in a uniform and sequential manner. And then the packets can be reordered at the output port by their route information. Among these existing solutions, the Byte-Focal switch, with low complexity and relatively good performance, is understood as the better scheme. Fig.1 shows the Byte-Focal switch architecture.

Note that, DTS can't work properly without the sets $S_j(t)$ and $S_j^{'}(t)$ and the two sets need to be maintained every timeslot. Here, $S_j(t)$ and $S_j^{'}(t)$ have the same meaning as in [1](see Ⅲ.A).For $0 \leq | S_j(t) | \leq N$ and $0 \leq | S_j^{'}(t) | \leq N$, we can conclude that the complexity of the maintenance operation is O(N) in the worst case. Obviously, it makes the Byte-Focal switch impractical in high-speed switch while N is large. So this paper introduces an improved Byte-Focal switch named CFSB, which uses the flow splitter at the first stage to solve the out-of-sequence problem at the cost of O(1) complexity.

The rest of this paper is organized as follows. Section II introduces the CFSB switch. In section III, we prove the stability and analyze the average delay in CFSB switch. In section IV, we study the delay performance of the CFSB switch by the simulations. Section V concludes the paper.

To ease the presentation, we assume that packets are of the same size and time is slotted and synchronized so that a packet can be transmitted within a timeslot. In addition, we define flow $F_{i,k}$ as the sequence of packets that arrives at the $i^{th}$ input port of the first stage and are destined for the $k^{th}$ output port of the second stage.

## 2. The CFSB Switch Architecture

The CFSB switch architecture (see Fig.2) is very similar to the Byte-Focal switch except that it uses the flow splitter instead of the DTS at the first stage. It is this small change that makes the CFSB switch lower complexity and better performance. Both the crossbar use a deterministic and periodic connection pattern and at any timeslot t, for any input port u and output port v of a crossbar, the connection pattern (u, v) is given by

$$v = (u+t) \bmod N \tag{1}$$

where u=0,1,2,…,N-1, v=0,1,2,…,N-1, t=0,1,2….

To illustrate this point, some terms will be used are defined as follows:

$FS_{i,k}$. Flow splitter for flow $F_{i,k}$.

$VPQ_{i,j}$. Virtual path queue at input i, corresponding to output j of the first stage.

$VOQ_{j,k}$. Virtual output queue at input j, corresponding to output k of the second stage.

The CFSB switch has a flow splitter for each flow at each input port. Packets arriving at the first stage are spread to Virtual Path Queues (VPQs) for load balancing by the flow splitters. To keep track of the $VPQ_{i,j}$ that the next packet from $F_{i,k}$ will be sent to, each $FS_{i,k}$ has a pointer $P_{i,k}$. Based on $P_{i,k}$, $FS_{i,k}$ distributes all the packets belonging to $F_{i,k}$ to the N VPQs in a round-robin manner. This mechanism can guarantee packets from the same flow are distributed to the central buffers evenly at the cost of O(1) complexity. In addition, packets can be reordered by their route information and thus make it feasible to use VIQ structure to emit packets in order. For detailed information about VIQ structure, please refer to [1](see Ⅲ.B).

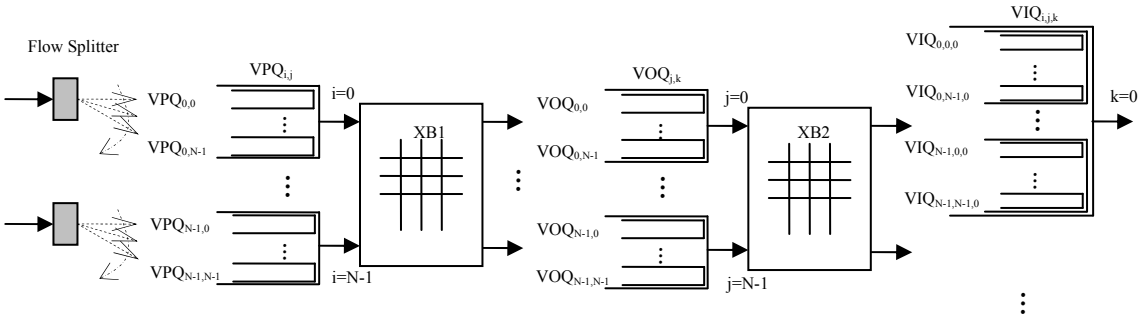


Fig. 2: The CFSB switch architecture

# 3. Stability and Delay Analysis

## 3.1 Stability

In this section, we do stability analysis for the CFSB switch. The two definition below are all from [1].

Definition 1: Let Q be the total queue length of a system and the system is said to be stable if E [Q] $<\infty$.

Definition 2: $Q_i(t)$ work-conserving: The $i^{th}$ input port is said to be $Q_i(t)$ work-conserving if it can be idle only if there are less than $Q_i(t)$ packets in the buffer of the input port.

Since packets from the same flow must be distributed into VPQs in a round-robin manner, the CFSB switch guarantees that the cumulative number of packets placed to each VPQs for a given flow differs by at most one. Let $q_{i,k}(t)$ be the queue length of $VPQ_{i,k}$ at timeslot t, $q_i(t)$ be the total queue length of all the VOQs at timeslot t, then we have Lemma 1.

Lemma 1: For an N×N CFSB switch and any i,j,k=0,1,⋯,N-1

$$|q_{i,u}(t)-q_{i,v}(t)| \leqslant N \tag{2}$$

Lemma 2: Each input port of the CFSB switch is N(N−1)+1work-conserving.

Proof: If $q_{i,j}(t)>0$ for some j=0,1,2⋯,N-1, we have $q_i(t) \leqslant N(N-1)$ according to the Lemma 1.

If $q_i(t) > N(N-1)$, there must be $q_{i,j}(t)>0$ for any j=0,1,⋯,N-1. It means that all the VPQs are not empty and the system can't be idle.

Since each input buffer receives at most one packet per timeslot, we can conclude that the system can't be idle while $q_i(t) \geqslant N(N-1)+1$.

Theorem 1: The re-sequencing buffer at each output port in CFSB switch is bounded by $2N^2-N$.

Proof: Reference [7] has shown that if each packet experience the same amount of delay at the first stage, N(N-1) timeslots, their departures form the first stage will be in the same order as their arrivals. And if the packets arrive at the second stage with their original sequence, reference [8] has shown that the maximum re-sequencing buffer size is $N^2$. Therefore, we have the above conclusion.

The second stage queue has been proved to be bounded in [4]. Based on the above analysis, we conclude that the CFSB switch is stable.

## 3.2 Average Delay Analysis

In this section, we analyze the average delay of the CFSB switch under uniform traffic. We assume a uniform Bernoulli i.i.d traffic model with the rate of λ.

*1) First Stage Delay:* Let $a_{i,j}(t)$ be the arrival to $VPQ_{i,j}$ in timeslot t and $q_{i,j}(t)$ be the length of $VPQ_{i,j}$ in timeslot t. Let $P_{i,k}(t)$ be the value of $P_{ik}$ at timeslot t. Since $P_{i,k}$ is updated in round-robin manner, cis a homogeneous Markov Chain whose state space is {0,1,2,⋯,N-1}. There must be an invariant distribution $\pi_i$ ={1/N,1/N,⋯,1/N}as t→∞. This means that $a_{i,j}(t)$ can be approximated by a Bernoulli process with a rate of λ /N. Assume that input i at the first stage is connected to output j at time T. Then we have the following recursive equation for $VPQ_{i,j}$

$$q_{i,j}(T+n) = q_{i,j}(T) + \sum_{s=1}^{n} a_{i,j}(T+s), n = 1, \cdots, N-1. \tag{3}$$

$$q_{i,j}(T+N) = \max\{q_{i,j}(T) + \sum_{s=1}^{N} a_{i,j}(T+s) - 1, 0\}. \tag{4}$$

According to [9], The recursion (4) has the solution

$$E[q_{i,j}(T)] = \frac{N-1}{N} \cdot \frac{\lambda^2}{2(1-\lambda)}, T = 0, N, 2N, \cdots. \tag{5}$$

$$E[q_{i,j}(T+s)] = E[q_{i,j}(T)] + s \cdot \frac{\lambda}{N},\ s = 1, 2, \cdots, N-1. \tag{6}$$

$$E[q_{i,j}(\infty)] = \frac{N-1}{N} \cdot \frac{\lambda}{2(1-\lambda)}. \tag{7}$$

Applying little's formula, the average queuing delay at the first stage under uniform traffic is

$$d_1 = \frac{N-1}{2(1-\lambda)} \cdot \tag{8}$$

*2) Second Stage Delay:*Since the load-balancing of the CFSB switch converts arriving packets into uniform traffic, each VOQ has Bernoulli arrivals with rate $\lambda / N$ . Therefore, using the same derivation as in the first stage, the queuing delay at the first stage is

$$d_2 = \frac{N-1}{2(1-\lambda)} \cdot \tag{9}$$

*3) Re-sequencing Delay:*Note that packets from the same flow may experience different delays at the first stage due to the different lengths of VPQs. This means that packets belonging to the same flow may depart from the first stage in an uncoordinated fashion and it may results in a worse re-sequencing delay for the CFSB switch in comparison to Byte-Focal.

## 4. Simulation Studies

Some simulation settings that will be used to test the performance of the CFSB switch are outlined here. In our simulations, the switch size N is set to be 32, and all inputs are equally loaded on a normalized scale $\lambda \in (0,1)$ . In this paper, we use the following traffic scenarios to test CFSB switch.

Uniform i.i.d.: $\lambda_{i,k} = \lambda / N$

Burst traffic: ON-OFF model is used to generate the burst traffic and the average burst length is set to be 64. We assume that packets within the same burst are destined to the same destination, which is uniformly distributed over N output ports.

Strong Diagonal i.i.d.: $\lambda_{i,i} = 2\lambda / 3, \lambda_{i,k} = \lambda / 3,$ for $i \neq k$ .

We compare the average delay induced by different switch architecture, such as Byte-Focal, CFSB, 4-iSLIP and OQ (Output Queuing). As seen in Fig.3, since the load balancing stage has no effect at all under uniform traffic, the load balanced switch architecture such as CFSB and Byte-Focal has a greater average delay than the single-stage switch like iSLIP. However, the CFSB switch has a smaller average delay as compared with Byte-Focal.

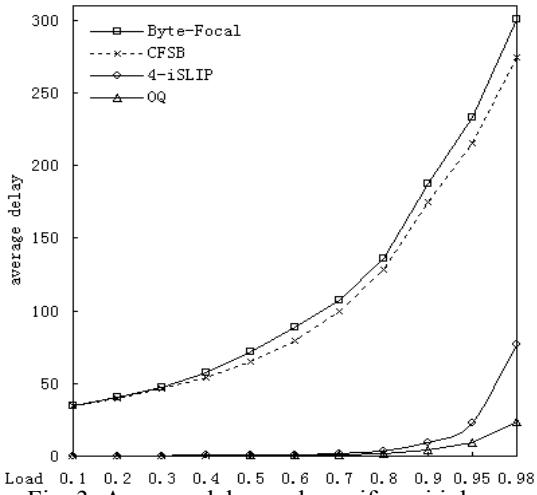

Fig. 3: Average delay under uniform i.i.d.

Fig.4 shows the average delay of various switches under bursty traffic. The performance of the single-stage switch like iSLIP is rapidly deteriorated with the load raised. In contrast, the load balanced switch architecture performs very well and the CFSB switch has a better performance in comparison to Byte-Focal again.

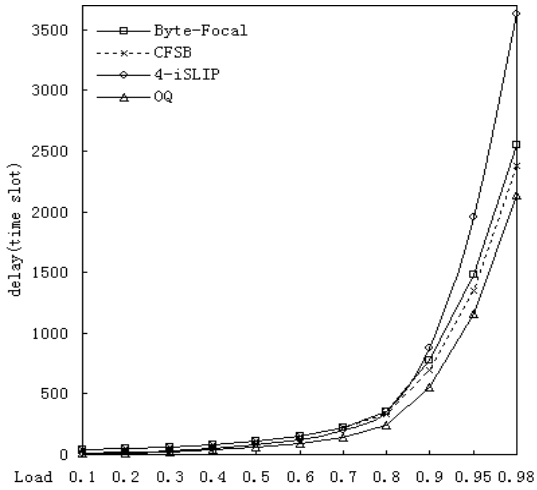

Fig. 4: Average delay under bursty traffic

Strong diagonal i.i.d. is a much skewed loading and the iSLIP becomes unstable near the load of 0.8, as we can see Fig.5. Compared to the Byte-Focal switch, CFSB has a much better delay performance. Fig.6 shows the 3-stage queuing delays of the Byte-Focal and CFSB switch under uniform i.i.d. As we can see that the second stage delay is comparable and the first stage delay of the CFSB is much smaller than that of Byte-Focal. This explanation of this phenomenon is as follows.

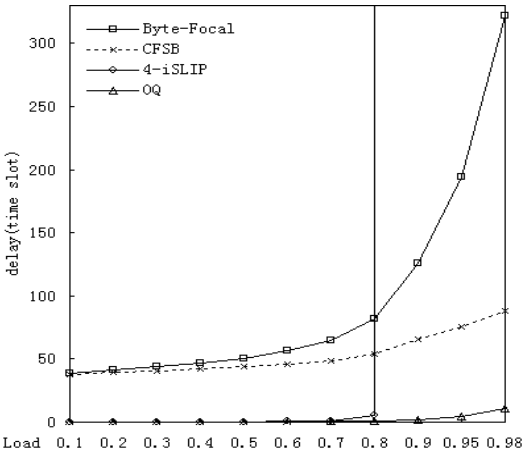

Fig. 5: Average delay under strong diagonal i.i.d.

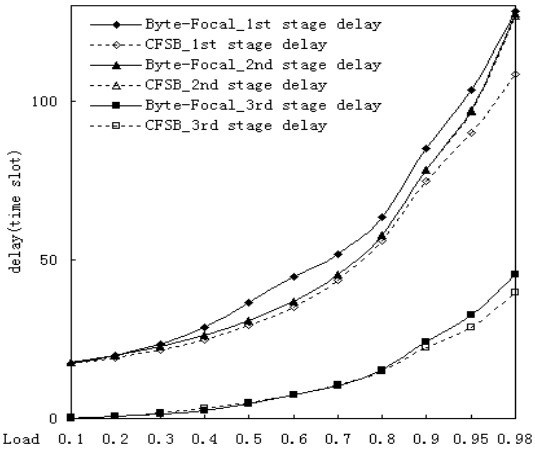

Fig. 6: 3-stage delays under uniform i.i.d.

Due to the fact that only the packets at the head of $VOQ1_{i,k}$ can send the request to the arbiter, the intuition for the Byte-Focal switch not performing well under certain circumstances, is that it may get trapped in "pseudo head-of-line(PHOL) blocking mode". Fig.7 shows the example of PHOL mode in a 4×4 Byte-Focal switch.

Without loss of generality, we assume that $VOQ1_{i,0}$ is served and the input i is connected to the output 3 at the first stage in timeslot t-1. Also, we assume that $P_{i,0}$=0,$P_{i,1}$=3, $P_{i,2}$=0 and $P_{i,3}$=1 at the beginning of the timeslot t, as shown in Fig. 7a.

During timeslot t, according to the rule of DTS, the new connection pattern allows packet A0 to be transferred to the buffer and $P_{i,0}$ is updated to 1. At the same time, packet C4 arrives at $VOQ1_{i,2}$, as shown in Fig. 7b.

During timeslot t+1, packet A1 is picked out to be transmitted and $P_{i,0}$ is updated to 2. Packet D1 arrives at $VOQ1_{i,3}$, as shown in Fig.7c.

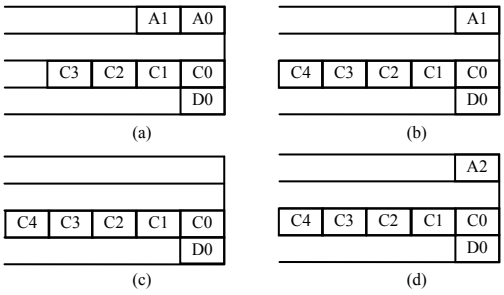


Fig. 7 Input port i of Byte-Focal.(a) timeslot t; (b) timeslot t+1; (c) timeslot t+2; (d) timeslot t+3.

During timeslot t+2, as seen in Fig.7d, packet A2 arrives at $VOQ1_{i,0}$, but no packet can be transferred to the second stage according to DTS. As a matter of fact, for $P_{i,2}$=0 and $P_{i,3}$=1, we know that packet C2 and D1 are transferable. Moreover, it is a similar case to packet C3 in timeslot t+3. This means that the Byte-Focal switch failed to make full use of the bandwidth at the first stage because of the head-of-line blocking.

Table 1. The 3-stage delays under uniform i.i.d.

| **load** | **0.1** | **0.2** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** | **0.95** | **0.98** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Byte-Focal_3rd stage delay** | 0.17 | 0.58 | 1.35 | 2.62 | 4.64 | 7.30 | 10.29 | 15.11 | 24.20 | 32.65 | 45.26 |
| **CFSB_3rd stage delay** | 0.21 | 0.76 | 1.70 | 3.08 | 4.95 | 7.44 | 10.68 | 14.90 | 22.17 | 28.87 | 39.60 |

Admittedly, this kind of blocking differs from traditional HOL in IQ(Input Queuing) in many aspects. To distinguish between the two, we call this kind of blocking occurred in the Byte-Focal switch pseudo head-of-line blocking (PHOL). Simulation results show that although the Byte-Focal can achieve 100% throughput, the effect of PHOL on the delay performance is not negligible.

In CFSB switch, all the packets are buffered in VCQs according to their path instead of their final destination, the PHOL mode is avoided. It is this point that makes the CFSB switch a better first-stage delay performance than that of Byte-Focal.

Form theorem 1, we know that the CFSB switch may result in a greater re-sequencing delay than that of Byte-Focal. Our simulation data (see Table I) show that the Byte-Focal switch really has a smaller re-sequencing delay than that of CFSB in light-load. However, for the elimination of the PHOL blocking, the CFSB switch performs better in re-sequencing delay than Byte-Focal in high load.

## 5. Conclusion

In this paper, we propose an improved Byte-Focal switch named CFSB, which produced by combining flow splitter with the Byte-Focal switch. The CFSB switch does not need any centralized scheduler and can achieve 100% throughput under a broad class of traffic matrices. The key property is that the CFSB switch reduces the complexity of the whole switching process to O(1) and this will make CFSB do a better job under high-speed switching environment. Although the re-sequencing buffer is extended to $2N^2$-N and the re-

sequencing delay may be greater than that of Byte-Focal, our simulation results show that the CFSB switch has better performance than Byte-Focal, especially in high-load.

## 6. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China No.60773102, and also in part by Foundation of Sichuan University (Next Generation Internet Architecture).

## 7. References

[1] Yanming Shen, Shivendra S. Panwar, and H. Jonathan Chao, “Design and performance analysis of a practical load-balanced switch,” IEEE Transactions on Communications, vol. 57, no. 8, pp. 2420-2429, 2009.

[2] WE Leland, MS Taqqu, W Willinger, and DV Wilson, “On the self-similar nature of Ethernet traffic (extended version),” IEEE/ACM Transactions on Networking (ToN), vol. 2, no. 1, pp. 1-15, 1994.

[3] Cheng-Shang Chang, Wen-Jyh Chen, and Hsiang-Yi Huang, "Birkhoff-von Neumann input buffered crossbar switches," Proceedings - IEEE INFOCOM. pp. 1614-1623.

[4] Cheng-Shang Chang, Duan-Shin Lee, and Yi-Shean Jou, “Load balanced Birkhoff-von Neumann switches, part I: One-stage buffering,” Computer Communications,vol.25,no.6,pp. 611-622, 2002.

[5] Isaac Keslassy, and Nick McKeown, "Maintaining packet order in two-stage switches," Proceedings - IEEE INFOCOM. pp. 1032-1041.

[6] Cheng-Shang Chang, Duan-Shin Lee, Ying-Ju Shih, and Chao-Lin Yu, “Mailbox switch: A scalable two-stage switch architecture for conflict resolution of ordered packets,” IEEE Transactions on Communications, vol. 56, no. 1, pp. 136-149, 2008.

[7] Cheng-Shang Chang, Duan-Shin Lee, and Ching-Ming Lien, “Load balanced Birkhoff-von Neumann switches, part II: Multi-stage buffering,” Computer Communications, vol.25,no.6,pp.623-634, 2002.

[8] Isaac Keslassy, Shang-Tse Chuang, Kyoungsik Yu, David Miller, Mark Horowitz,Olav Solgaard, and Nick McKeown,"Scaling Internet Routers Using Optics,"Computer Communication Review.pp.189-200.

[9] M Schwartz, Broadband integrated networks, p.^pp. 190-200, New York: Prentice Hall PTR,Upper Saddle River, NJ, 1996.