Birkhoff-von Neumann Switch with Deflection Based Load Balancing

Srđan Durković and Zoran Čiča

Abstract — **Load balanced Birkhoff-von Neumann (LB-BvN) packet switches have low hardware complexity while achieving high performance. We propose a novel LB-BvN based switch that achieves 100% throughput for any admissible traffic scenario. The proposed switch uses the deflection mechanism to decrease overall hardware complexity of the switch. The delay and buffer bounds of the proposed switch are derived and analyzed using the network calculus theory. The proposed switch is compared to other LB-BvN based solutions.**

Keywords — **Birkhoff-von Neumann switches, load balancing, network calculus, packet switching.**

I. Introduction

THE network operators are installing higher and higher link capacities in order to support increasing traffic demands. Routers and switches must implement highly efficient packet switches to support these continuously increasing link capacities. It is imperative that the packet switch achieves high performances under any admissible traffic scenario. Also, the packet switch needs to be scalable, both in terms of supported number of ports and supported link capacities.

The Birkhoff-von Neumann (BvN) based switches are very popular because they avoid the problem of calculating the packet switch configuration on the fly like input queued switches. BvN switch uses the capacity decomposition approach [1] - [2]. Based on the traffic demands, a set of packet switch configurations is calculated. The calculated set of configurations is periodically repeated, thus, the problem of calculations in the real time is avoided (O(1) complexity to configure the packet switch). But, as the traffic demands dynamically change, the set of packet switch configurations should be frequently recalculated. The computation complexity is very high - $O(N^{4.5})$. Also, it is not easy to perform real time measurements of the traffic fluctuation.

Load-balanced two stage BvN (LB-BvN) switch was proposed to couple with these problems [3] – [4]. The packet switch in each stage periodically repeats N configurations. There are no calculations of the packet switch configuration. The first stage balances the incoming traffic to buffers that are placed between the first and second stage. The first stage tries to create the uniformly distributed traffic for the second stage to avoid the need for the recalculations of the packet switch configurations. The load-balanced BvN switch achieves high performances for a broad class of traffic scenarios. However, in some traffic scenarios the switch throughput can be severely decreased. Also, out-of-order problem occurs because the packets of the same flow go through different paths in the LB-BvN switch. The resequencing buffer is used at the output port to restore the original packet order. In [4], the authors proposed the load balancing of the packets according to flows not the arrival times to support the multicast flows and they placed the jitter control in front of the VOQs of the second stage. The role of the jitter control is to delay the packets so it seems that the packets enter the second stage like they entered the first stage. Thus, jitter control reduces the jitter caused by load balancing.

Many improvements of the LB-BvN switches are proposed in the literature. Two schemes, EDF (Earliest Deadline First) based scheme and frame based scheme, were proposed in [5] to provide guaranteed rate services in LB-BvN switches. Byte-Focal (BF) switch uses the VOQs in the first stage to perform the load balancing based on flows [6-7]. Each flow has a dedicated VOQ at the input port. These VOQs are served according to some scheduling scheme like round-robin or the longest queue first. Thus, it is guaranteed that the packets of the same flow are evenly distributed across the second stage. Practical BvN implementation is presented in [8]. This implementation uses a folded architecture, i.e. only one packet switch is used instead of two switches. Deflection compensated mechanism is proposed in [9]. In the case of bursty traffic some buffers may be overloaded, thus, the packets from the overloaded buffers are deflected to other ports to avoid packet losses. Frame based LB-BvN switches represent a class of LB-BvN switches that switch frames instead of individual packets [10] - [12]. A frame is defined as a set of N packets that belong to the same flow, where N is the number of ports. When a frame is completed at the input port, the frame is sent to the second stage. All packets in the frame experience the same delay through the switching stages. Thus, there is no difference in delays between the packets of the same frame, and as a consequence there are no packets out of order. The major drawback of the frame-based LB-BvN switches is large packet delay under light loads because a large time is needed to completely fill the frame. FOFF (Full Ordered Frames First) allows the

Paper received April 11, 2017; accepted June 20, 2017. Date of publication July 31, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Grozdan Petrović.

This paper is a revised and expanded version of the paper presented at the 24th Telecommunications Forum TELFOR 2016 [14].

Srđan Durković is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11120 Belgrade, Serbia (e-mail: srdjad6@gmail.com).

Zoran Čiča is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11120 Belgrade, Serbia (e-mail: zoran.cica@etf.bg.ac.rs).

transmission of incomplete frames to reduce the average packet delay [10]. However, transmission of incomplete frames as a consequence has packet out of order problem. Thus, FOFF requires the resequencing buffers. PF (Padded Frames) also allows transmission of incomplete frames [11]. However, the incomplete frames are padded with dummy packets to emulate full frames, thus, there is no packet out of order problem like in FOFF. However, the transmission of dummy packets underutilizes the internal link capacities. CR (Contention and Reservation) switch is a frame based scheme that uses frame based approach combined with feedback mechanism and individual packet transmissions [12]. Under light loads, CR allows transmission of individual packets to reduce the packet delay. Since the transmission of individual packet might be unsuccessful (the number of individual packets in buffers between the first and second stage is limited), a feedback mechanism is used to notify the input ports about the unsuccessful transmissions of their individual packets.

In this paper we propose a non-blocking switch that is based on the load-balanced BvN switch. The proposed switch combines some of the aforementioned approaches to achieve optimal implementation. At each input port, we use per flow round-robin load balancing to avoid the problem of original LB-BvN switch for some traffic scenarios where the packets destined for the same output are all forwarded to the same buffer between the first and second stage. We use folded architecture, where only one switch stage is used, to reduce the hardware requirements of the switch. Deflection of the balanced packets is performed because the folded architecture is used. Deflection mechanism forwards the balanced packets to their destination output ports. In the packet switch, we use the speedup of two due to the folded architecture. However, the frequency of the packet switch configuration changes does not need the speedup i.e. one packet switch configuration per time slot is performed. In the paper, we prove that the proposed switch is non-blocking for any admissible traffic scenario. Also, using the network calculus we derive the delay and buffer upper bounds for the proposed switch. The proposed switch is compared to other LB-BvN based switches.

The remainder of the paper is organized as follows. In the next section, we describe the proposed switch architecture in detail and give the proof that the proposed switch is non-blocking. In the third section, we derive the analytical delay and buffer upper bounds. The simulation results and the comparison of simulation results and derived analytical bounds are presented in the third section as well. The proposed switch is compared to other LB-BvN switches in the fourth section. The fifth section concludes the paper.

II. BvN Switch with Deflection Based Load Balancing

In this section, we give a detailed description of our proposed BvN switch with deflection based load balancing (BvN-DLB). At the end of the section, we give a proof that the BvN-DLB switch is non-blocking. We assume a fixed-size packet under the term packet. In the case of variable-size packets (i.e. IP packets), the variable-size packets would be split into fixed-size packets at the input port.

The BvN-DLB architecture is shown in Fig. 1. BvN-DLB switch performs the load balancing of the incoming traffic by deflecting the packets across all ports. Balancing is performed on per flow basis to eliminate the scenarios where the packets destined to the same output would be balanced across the small number of same ports as that could decrease the throughput of the switch. The deflection mechanism enables us to use only one packet switch instead of two which reduces the overall hardware requirements of our BvN-DLB switch.

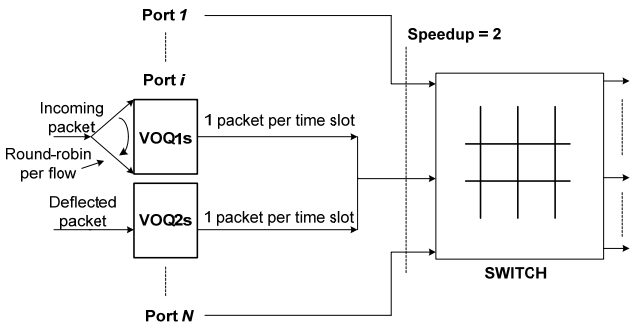

Fig. 1. BvN-DLB architecture.

There are two types of virtual output queues (VOQ) at each port - VOQ1 and VOQ2. Both types of queues are served in FIFO manner. VOQ is used instead of the FIFO memory because all queues are stored in the same physical RAM memory. VOQ1 queues are used for load balancing and they store the incoming packets. There are N VOQ1 queues at one port. $VOQ1_{ik}$ denotes the VOQ1 queue at port i that stores the packets that are deflected to port k. When a new packet that belongs to flow F_{ij} arrives, the flow's round robin pointer is used to determine the VOQ1 queue where the packet is written and then the flow's round robin pointer is incremented. F_{ij} denotes the flow from input port i to output port j. Round-robin pointers per flow add very little to the hardware complexity of the switch. There are N round robin pointers, and each of them requires $\log_2 N$ bits to be able to address N VOQ1 queues. These per flow round-robin pointers prevent the traffic scenarios where the packets of the same flow would be balanced via only one port or a small number of ports. If flows from several inputs would be balanced through the same port (or the same small set of ports), the throughput of the switch could be significantly decreased.

VOQ2 queues store the deflected packets and forward them to their final destination (corresponding output ports). There are N VOQ2 queues at one port. $VOQ2_{kj}$ denotes the VOQ2 queue at port k that stores the deflected packets whose final destination is the port j. When the deflected packet reaches the port k, the packet is written to a corresponding VOQ2 queue that is selected based on the final destination of the deflected packet.

During consecutive N time slots, a packet switch connects each port to every other port exactly once. When a port i is connected to a port j, one packet from the $VOQ1_{ij}$ queue is sent to port j and one packet from the $VOQ2_{ij}$ queue is sent to port j. Thus, the speedup of two is used as two packets at most are sent during one time slot. However, the

frequency of the packet switch configurations remains the same because there is only one packet switch configuration per time slot. Due to load balancing, packets from the same flow travel different paths through the switch, thus, the original order of the packets can be disrupted. A small resequencing buffer is used at the output port to restore the original order of packets.

The proof that BvN-DLB switch is non-blocking for any admissible traffic scenario is easy to derive. Let C denote the input and output link capacity of one port, and it is equal to the throughput of one packet per time slot. Let R_{ij} denote the rate of the flow F_{ij}. We assume that the buffer sizes are large enough to avoid any packet losses. In the next section, we derive the buffer upper bounds. Obviously, the relation:

$$\sum_{j=1}^{N} R_{ij} \le C, i = 1..N \tag{1}$$

must hold, as the total rate of all flows arriving at the same port cannot exceed the input link capacity. We assume the admissible traffic, so none of the output ports is overloaded:

$$\sum_{i=1}^{N} R_{ij} \le C,\ j = 1..N . \tag{2}$$

We balance evenly each flow across N ports, thus, the flow's rate that enters $VOQ1_{ik}$ (i=1..N, k=1..N) queue is:

$$\sum_{j=1}^{N} R_{ij} / N \le C / N . \tag{3}$$

In every N-th time slot, port i is connected to port k, thus, the $VOQ1_{ik}$ is served with C/N rate and according to (3) the VOQ1 queues cannot be overloaded.

The flow's rate of deflected packets that enter the $VOQ2_{kj}$ (k=1..N, j=1..N) queue can be easily calculated because all the flows destined to the same output are evenly balanced across all ports:

$$\sum_{i=1}^{N} R_{ij} / N \le C / N . \tag{4}$$

In every N-th time slot, port k is connected to port j, thus, the $VOQ2_{kj}$ is served with C/N rate and according to (4), the VOQ2 queues cannot be overloaded. Since none of the VOQ queues can be overloaded, the switch is stable (i.e. non-blocking) under any admissible traffic scenario.

III. Delay and Buffer Bounds

In this section, we analyze the delay and buffer bounds of our proposed BvN-DLB switch using the network calculus. Network calculus provides the analytical tool to calculate the guaranteed bounds for delay and buffering that are never violated [13]. Arrival curves are used to represent the upper bound of the incoming traffic, while the service curves are used to represent the service of the incoming traffic [13]. The min-plus convolution of arrival and service curve represents the lower bound in the time domain of the traffic outgoing from the service node [13]. The delay and buffer bounds (D and B) can be calculated by using this lower bound of the outgoing traffic and arrival curve of the incoming traffic as shown in Fig. 2 [13].

The traffic in the packet networks is bursty. The arrival curve defined as:

$$\alpha(t) = \begin{cases} 0, & t = 0 \\ \sigma + Rt, & t > 0 \end{cases} . \tag{5}$$

can be used to describe the bursty traffic, where σ represents the burstiness and R represents the average rate of the traffic. This arrival curve is shown in Fig. 2. For each flow F_{ij} we use the definition (5) with parameters σ_{ij} and R_{ij}. Each VOQ in BvN-DLB switch receives the service in every N-th time slot with the rate C. The service curve is shown in Fig. 3. However, in our analysis we use the rate latency service curve that is based on the average service rate C/N of the VOQ (Fig. 3). The use of the rate latency service curve results in a slightly more pessimistic delay and buffer bounds, but the calculation of the bounds is significantly more simple. In both service curves, the latency is set to (N-1)T_s, where T_s is the duration of one time slot. This latency represents the worst case when the packet that entered empty VOQ has just missed the connection for that VOQ and must wait the (N-1) time slots to be served.

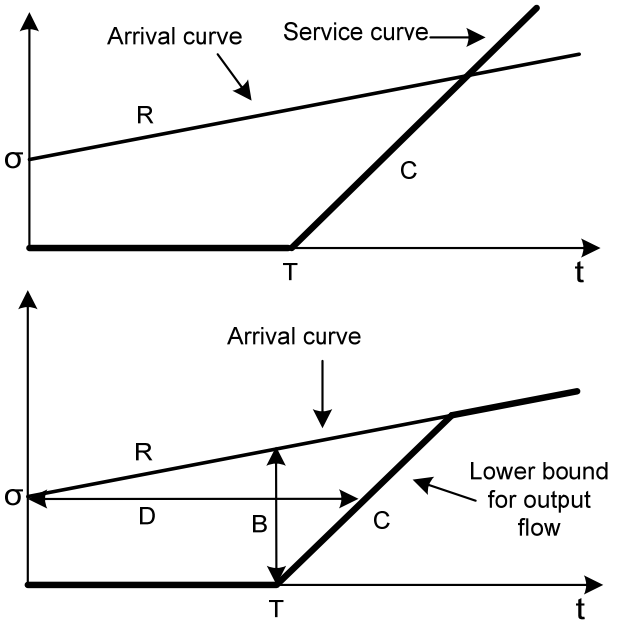

Fig. 2. Delay and buffer bounds.

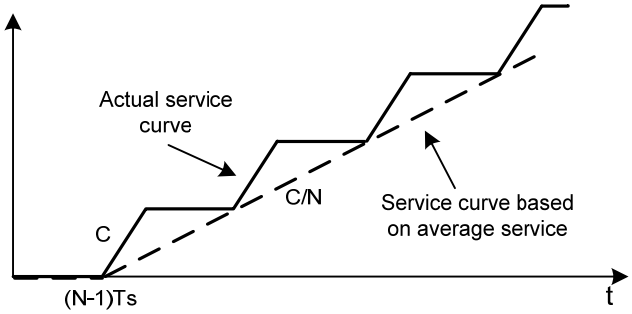

Fig. 3. Service curves.

Because we use per flow round-robin load balancing, the flow F_{ij} is split into N equal subflows, each entering one VOQ1 queue at the port i. The arrival curve of one subflow is (we show only the value for t>0):

$$\alpha_{ij}(t) = 1 + \sigma_{ij} / N + R_{ij} t / N . \tag{6}$$

The burstiness is not σ_{ij}/N, but $1+\sigma_{ij}/N$ due to the fact that we split the flow on packet basis which means that each VOQ1 queue at port i will receive an integer number of packets. However, σ_{ij}/N does not have to be an integer number, thus, some VOQ1 queues might receive one packet beyond the limit σ_{ij}/N. The aggregate of all subflows that enter $VOQ1_{ik}$ (k=1..N) queue is:

$$\alpha_{VOQ1ik}(t) = \sum_{j=1}^{N}\left(1+\sigma_{ij}/N + R_{ij}t/N\right). \quad (7)$$

Given the rate latency service curve it is easy to calculate delay (D) and buffer (B) bound for the $VOQ1_{ik}$ queue as shown in Fig. 2:

$$\begin{aligned} D_{VOQ1ik} &= (N-1)T_s + \sum_{j=1}^{N}\left(N+\sigma_{ij}\right)/C \\ B_{VOQ1ik} &= \sum_{j=1}^{N}\left(1+\sigma_{ij}/N + R_{ij}(N-1)T_s/N\right) \end{aligned} \quad (8)$$

The subflow arrival curves at the output of the $VOQ1_{ik}$ have the same average rate, but increased burstiness [10]:

$$\sigma_{ij}' = 1+\sigma_{ij}/N + R_{ij}\left((N-1)T_s + \sum_{\substack{j=1\\ j\neq i}}^{N}\left(N+\sigma_{ij}\right)/C\right)/N. \quad (9)$$

However, all bursts of flows on input link i cannot arrive simultaneously at the same moment. Therefore, we can use a more optimistic equation for increased burstiness, where the influence of the other flows is omitted from (9):

$$\sigma_{ij}' = 1+\sigma_{ij}/N + R_{ij}(N-1)T_s/N. \quad (10)$$

The flows, destined for the same output j, enter the $VOQ2_{kj}$ queue at the port k (k=1..N). These flows have increased burstiness. We can calculate the delay and buffer bound of VOQ2 queues in the same way as we did for the VOQ1 queues (service curve is the same as in the VOQ1 case):

$$\begin{aligned} D_{VOQ2kj} &= (N-1)T_s + \sum_{i=1}^{N}\left(N+\sigma_{ij}'\right)/C \\ B_{VOQ2kj} &= \sum_{i=1}^{N}\left(1+\sigma_{ij}'/N + R_{ij}(N-1)T_s/N\right) \end{aligned} \quad (11)$$

The maximum delay through BvN-DLB switch $D_{\max}$ and maximum buffer requirements per port $B_{\max}$ can be found combining the bounds for VOQ1 and VOQ2 queues given in (8) and (10):

$$\begin{aligned} D_{\max} &= \max_{i,j,k=1..N}\left(D_{VOQ1ik} + D_{VOQ2kj}\right) \\ B_{\max} &= \max_{i=1..N}\left(\sum_{k=1}^{N}\left(B_{VOQ1ik} + B_{VOQ2ik}\right)\right) \end{aligned} \quad (12)$$

It is easy to notice from (8) and (10) that the bounds linearly grow with the increase of the burstiness. This behavior represents a very important and good property of our proposed switch.

We have simulated the proposed switch in order to inspect the correctness of the derived bounds. We have simulated uniform (u), hot spot (h) and diagonal (d) traffic scenarios [6]. In all three scenarios, packets are generated to fit the arrival curve shown in Fig.2. The burst size σ for all flows is set to 100 packets in all three scenarios. The rate R is set for each flow to correspond the given scenario. In uniform scenario, the destinations of the packets at some input port are equally distributed, i.e. the rate R for each flow is set to u/N, where u is the load at the input port. In hot spot scenario each input port has its own 'hot spot' output port. For input port i, the hot spot output port is output port i. At the input port, 50% of packets are destined for the corresponding hot spot output port, while other packets are equally distributed to other output ports. Thus hot-spot flows have the rate R set to $u/2$, while other flows have their rate R set to $u/2(N-1)$. In diagonal scenario, at the input port i, 50% of packets are destined to output port i, and 50% of packets are destined for output port i+1. Thus, the diagonal flows have their rate R set to $u/2$. For all three scenarios, we have also tested the special case where the bursts arrive one after another at all input ports, and the bursts are synchronized according to their destinations. For example, in uniform and hot spot scenarios, at all input ports: burst for the output 0 first arrives, then for the output 1 and etc. This 'special' case should be one of the worst case scenarios. We simulate various switch sizes and they all exhibit the same behavior. Thus, to avoid redundancy, we present the results for the switch size 32x32.

Figs. 4 and 5 show D_{max} and B_{max} for the simulated switch. In graphs, w denotes the aforementioned special case, and b denotes the delay/buffer bound. The log scale is used for y axis. Note that bounds for the uniform and hot-spot scenarios are equal. D_{max} and B_{max} are lower in the normal scenarios than in the worst case scenarios, because the worst case scenario is unlikely to happen. For the same reason, the gap between normal and worst case scenarios increases as the traffic burstiness increases. The derived delay and buffer bounds are not violated even in the worst case scenarios, thus, the bounds can be used for deriving the delay guarantees and switch buffer dimensioning. D_{max} and B_{max} do not grow significantly with the load increase because the traffic burstiness has a larger impact than the load on the switch behavior. This behavior indicates that the traffic shapers that would smooth (i.e. decrease) the traffic burstiness could increase the overall network performance.

IV. Performance comparison

In this section we compare the proposed BvN-DLB to other most popular existing LB-BvN schemes (EDF, BF, FOFF, CR). We compare the performances in terms of average packet delay and maximum packet delay, both measured in time slots. The comparison is performed for two traffic scenarios: the Bernoulli uniform scenario and Bernoulli hot spot scenario. In both scenarios, the packets arrive at the input ports according to the Bernoulli process. In the uniform scenario, the destinations of the packets at some input port are equally distributed. In the hot spot scenario each input port has its own 'hot spot' output port. For input port i, the hot spot output port is output port i. At the input port, 50% of packets are destined for the corresponding hot spot output port, while other packets are equally distributed to other output ports. We show the performance comparison results for 32x32 switch. Note that the comparison results are the same for other switch sizes, so we omit the display of results for other switch sizes to avoid redundancy.

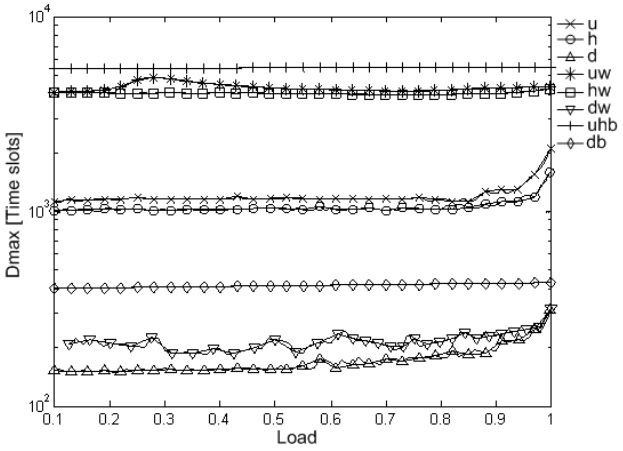

Fig. 4. Maximum delay.

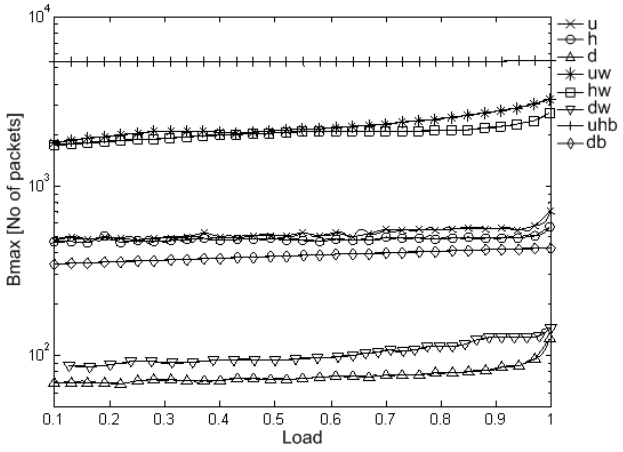

Fig. 5. Buffer requirements per port.

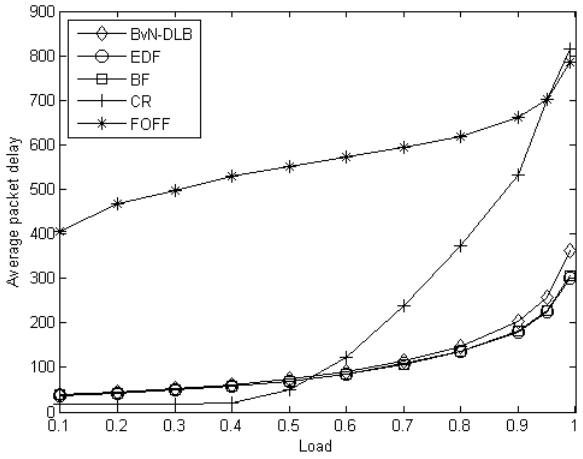

Fig. 6. Average packet delay for Bernoulli uniform scenario.

Figs. 6 and 7 show the average packet delay for uniform and hot spot scenario, respectively. We omit FOFF from Fig. 7 because FOFF has the worst performance in the same way as in Fig. 6. CR achieves the best results under light loads. However, under medium and heavy loads, CR has a significantly larger average packet delay than non-frame based schemes, because at these loads CR dominantly behaves as a classic frame based solution. Our proposed BvN-DLB achieves a similar performance as EDF and BF. Note that EDF and BF do not use a folded architecture, thus, they require two physical switches, unlike our BvN-DLB. Also, EDF requires search through corresponding VOQ in the second stage to find the packet with the earliest deadline which limits the scalability of the EDF switch.

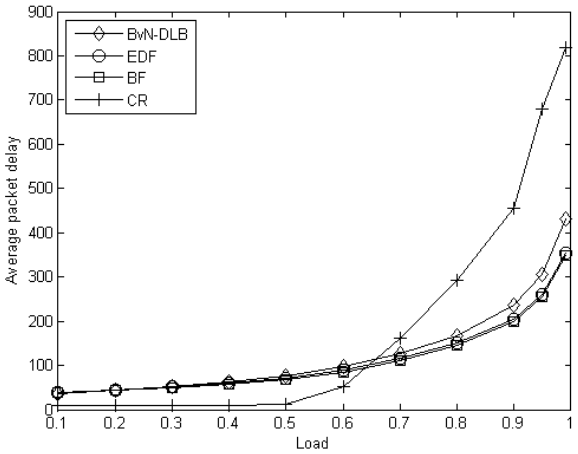

Fig.. 7. Average packet delay for Bernoulli hot spot scenario.

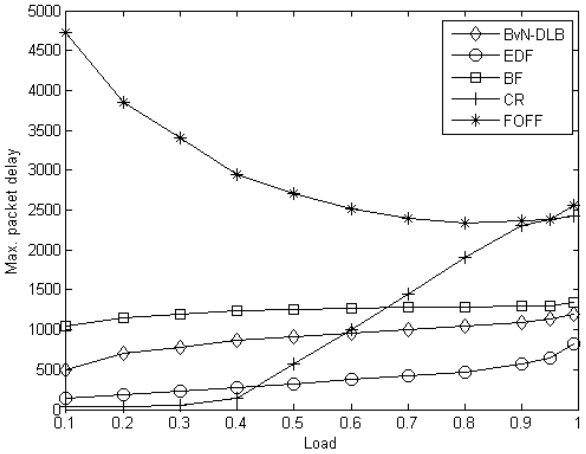

Fig. 8. Maximal packet delay for Bernoulli uniform scenario.

Figs. 8 and 9 show the maximal packet delay for uniform and hot spot scenario, respectively. Again, we omit FOFF from Fig. 9 because FOFF has the worst performance in the same way as in Fig. 8. Under medium and heavy loads, CR has a worse maximum packet delay than BF, EDF and our BvN-DLB. EDF achieves the lowest maximum packet delay. However, as mentioned earlier, EDF requires search through corresponding VOQ in the second stage to find the earliest deadline packet. This process requires multiple memory accesses that limit the EDF scalability in terms of supported port speed.

The performance comparison of our proposed BvN-DLB to other LB-BvN schemes shows that our proposed scheme achieves a good performance that is comparable or even better than the performance of other schemes, while achieving low hardware complexity.

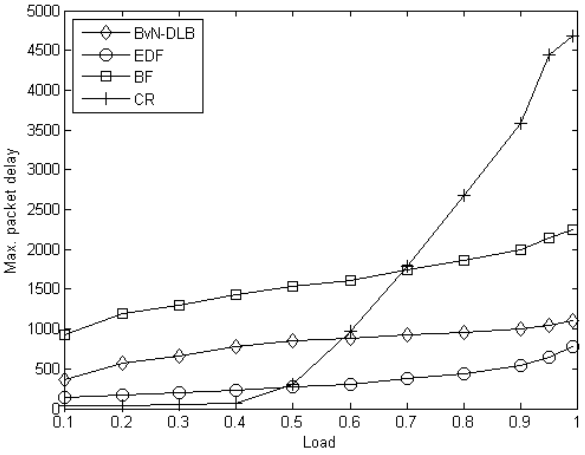

Fig. 9. Maximal packet delay for Bernoulli hot spot scenario.

V. Conclusion

In this paper, we propose the BvN-DLB switch that has very good performances. We show that the proposed switch is non-blocking and we derive the delay and buffer upper bounds of the switch for the bursty traffic. The BvN-DLB switch has very low hardware complexity which makes it very attractive for practical implementation in routers and switches.

References

[1] T.T. Lee and C.H. Lam, "Path switching-a quasi-static routing scheme for large-scale ATM packet switches," *IEEE Journal on Selected Areas in Communications*, vol.15, no.5, pp.914-924, June 1997.

[2] C. S. Chang, W. J. Chen, and H. Y. Huang, "On service guarantees for input-buffered crossbar switches: a capacity decomposition approach by Birkhoff and von Neumann," *Proc. of IWQoS '99*, pp.79-86, June 1999.

[3] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering," *Computer Communications*, vol.25 , no.6, pp.611-622, April 2002.

[4] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches, part II: multistage buffering," *Computer Communications*, vol.25 , no.6, pp.623-634, April 2002.

[5] C.S. Chang, D.S. Lee, and C. Y. Yue, "Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches," *Proc. of INFOCOM 2003*, pp.1622-1632, April 2003.

[6] Y. Shen, S. Jiang, S.S. Panwar, H.J. Chao, "Byte-focal - a practical load balanced switch," *Proc. of HPSR 2005*, pp.6-12, May 2005.

[7] Y. Shen, S.S. Panwar, H.J. Chao, "Design and performance analysis of a practical load-balanced switch," *IEEE Transactions on Communications*, vol.57, no.8, pp.2420-2429, August 2009.

[8] C.T. Chiu, Y.H. Hsu, and al., "A scalable load balanced Birkhoff-von Neumann symmetric TDM switch IC for high-speed networking applications," *Proc. of ISCAS 2007* , pp.2754-2757, May 2007.

[9] J. Zhang, T. Ye, T.T. Lee, F. Yan, W. Hu, "Deflection-Compensated Birkhoff-von-Neumann Switches," *Proc. of WOCC 2013*, pp. 518-522, May 2013.

[10] I. Keslassy, and al., "Scaling internet routers using optics," *Proc. of SIGCOMM 2003*, Karlsruhe, Germany, Aug. 2003.

[11] J.J. Jaramillo, F. Milan, and R. Srikant, "Padded frames: a novel algorithm for stable scheduling in load-balanced switches," *IEEE/ACM Transactions on Networking*, vol.16, no.5, pp.1212-1225, 2009.

[12] Chao-Lin Yu, Cheng-Shang Chang, Duan-Shin Lee, "CR switch: A load-balanced switch with contention and reservation," *IEEE/ACM Transactions on Networking*, vol.17, no.5, pp.1659-1671, 2009.

[13] J.Y. Le Boudec and P. Thiran, "Network Calculus: A Theory of Deterministic Queuing Systems for the Internet,", Springer, 2001.

[14] S. Durković and Z. Čiča, "Birkhoff-von Neumann switch with deflection based load balancing," *2016 24th Telecommunications Forum (TELFOR)*, Belgrade, 2016, pp. 1-4.