Analytical Modeling of OpenFlow-Switches Based on the Buffering Design

Strahil Panev, Pero Latkoski, Dimitar Bogatinov

Abstract – **The overall performance of the OpenFlow (OF) switches greatly relies on the output buffer design. In this paper an analytical approach for modelling the average packet loss rate of two OF-switch buffering mechanisms is provided: standard single shared buffer, and an advanced isolated two priority buffer. The results proved that using buffers with non-preemptive prioritization is the clear design choice for the user plane of mobile core networks, as it provides significant performance gains when compared to traditional buffering.**

Keywords – **OpenFlow, Performance evaluation, Mobile networks, Queuing theory.**

I. Introduction

Software Defined Networking (SDN) and Network Function Virtualization (NFV) are the key technologies used by mobile operators to cope with traditional challenges, such as high operational costs, vendor lock-in, slow time to market etc. SDN separates the control and user plane and brings the benefit of having open and easy programmability and application development, whereas NFV allows network operators to manage and expand their network capabilities on demand using virtual, software-based applications where physical boxes once stood in the network architecture [1]. Today, the most widely spread protocol for control-user plane communication is OpenFlow (OF) [2].

In existing Long-Term Evolution (LTE) networks, SDN is widely used in data centers and the mobile core network. Initially, the concept of SDN was introduced in the Evolved Packet Core (EPC), by separating the Packet Data Network Gateway (PGW) and the Serving Gateway (SGW) into specific logical parts that belong to the control and user plane. It is important to note that SDN is embedded in the heart of the new 5G (5^{th} Generation) architecture since the very beginning, bringing many benefits for the newly defined 5Guse cases. One of the greatest advantages of introducing SDN in mobile core networks is the reduction of the overall latency. In an SDN-based core network, in case of an MN (Mobile Node) handover procedure, specific OF messages are exchanged between the user and the control plane. The so called "hard handover", includes a breakage of the existing user session followed by a set of OF signaling messages that include reconfiguration and management of network entities.

Article history: Received March 09, 2021; Accepted July 06, 2021

Strahil Panev is with the Faculty of Computer Science, International Slavic University "G. R. Derzhavin", Sv. Nikole, R. Macedonia, E-mail: strahil.panev@msu.edu.mk

Pero Latkoski is with the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia, E-mail: pero@feit.ukim.edu.mk

Dimitar Bogatinov is with the Military academy "General Mihailo Apostolski" - Skopje, University "Goce Delcev" - Shtip, R. Macedonia, E-mail: dimitar.bogatinov@ugd.edu.mk

A buffer in the OF-switch is typically used to temporary store the incoming packets before the switch can serve them. The capacity of the output buffers and the buffer design in the OF-switch, have a high impact on the average switch service time. Regarding the buffer design, the most straightforward implementation today is the use of a single buffer that is shared by all ports. This is the cheapest and least complex solution that is widely popular in real implementations. However, there are proposals for other types of buffer designs, that usually include the use of two isolated separate buffers, one only for the data plane, and the other exclusively for the control plane.

The work in this paper is an extension of our previous work [3], where the aim was to quantify the total handover delay experienced by MN, and compare this delay when using a traditional single buffer switch with a two-buffer priority switch design. In this work, the mathematical modelling of our previous work is extended, with aim to quantify the packet loss rate of the two already created systems that have different buffer designs. The novelty of the work presented here lies in the applicability of mathematical modelling to shape a traditional buffering system, and a more advanced buffering mechanism that uses packet prioritization. The main contributions of this paper, can be summarized in the following points:

1) A mathematical model based on queuing theory is proposed. This analytical approach models the controller as M/M/1, and the switch as GI/M/1/K queue. The QBD (Quasi-birth-death) processes are used to define the stationary behavior of the network, which later allows for performance metric calculation,
2) Extensive numerical analysis is performed to verify the validity of the model. Key factors that impact the packet loss rate are analyzed, such as arrival rate,and controller-to-switch mean service rate ratio,
3) The results obtained can be easily used by network designers when planning an SDN-based core network.

The rest of the paper is organized in the following way: In section 2, the past literature of interest is reviewed, and in section 3 the two compared systems and the analytical model are defined. In section 4, the results are presented and commented in detail on the findings. Finally, in section 5 a summarization of the highlights of this research is done.

II. Previous Work

The very first analytical modelling of SDN networks was done by Jarchel et al. [4], where the authors proposed an analytical model which was later verified via a hardware testbed. The model used a M/M queue for the forwarding system at the switch, whereas a M/M-S queue is used to model the feedback system of the controller. Only a single switch and a single controller are modelled, which is the major drawback of the proposed mathematical approach. Yen et. al. [5], used a two-stage tandem network by using the M/M/1 queue, however their proposed model does not clearly separate the control and user plane traffic. The authors in [6] perform experiments which proved that using M/M/1 queue for the OF-switches is not recommended, instead they considered the M/G/1 model and discussed that using this type of queue mimics much better the real SDN implementations. By applying similar thinking, in [7] the authors used MMPP/1/1 (Markov modulated Poisson process) queue for both the switches and the controller, whereas the creators of [8] deployed the M/Geo/1 model. However, both did not take into consideration a distinction of the control and user plane packets.

Miao et. al. [9] deployed preemptive priority queuing at the switches and modeled them by using infinite buffers. Best to our knowledge, this was the first paper that deals with different buffer designs, by analyzing single shared buffering and comparing the results with a more advanced priority buffering mechanism. The results clearly showed that the average service time of the switches and the overall network performance is significantly improved when deploying priority buffering, yet again at the cost of increased complexity.The proposed analytical modeling in [10] and [11], modeled the low and high priority queues as finite by using the GI/M/1/K queue and by taking advantage of the QBD processes. Themodel implemented non-preemptive prioritization, which means that the packets in the queue with lower priority are served only in the case when there are no packets to be served in the high priority queue. In general, if the OF-switch uses finite buffering [12], the Markovian properties of the buffer queues are not preserved. The latter is the main condition for obtaining product-form analysis. A network with finite buffering can have its stationary behavior described only by using global-balance equations.

In our proposal, the switches are modeled by using finite buffering by taking advantage of the GI/M/1/K queue, whereas the controller is modelled as M/M/1 queue, with infinite buffer capacity. The design implements two isolated priority queues, one for the user plane, the other for the control plane. The QBD processes approach to obtain the stationary state distribution of the network is used, which allowed us to calculate the average packet loss rate.

III. Analytical Model

In our previous work [3], a mathematical approach was proposed to model the OF-based signalling messages exchanged within the procedure of “hard” handover. Two systems that use different OF-switch buffer designs were compared, single shared buffer, and two isolated buffers that incorporate prioritization. The aim was to quantify the handover delay experienced by the MN when moving from one switch to another within the SDN-based mobile core network. In this work, quantifying the average packet loss of the two systems is of interest.

Fig. 1 shows the two system designs, Fig. 1a depicts the model called *Shared Finite Buffering (Model SFB)*, Fig. 1b gives the model called *Priority Finite Buffering (PFB)*. Model SFB deploys a switch with a single shared buffer without priority, where the traffic entering the switch is the sum of the traffic returned by the controller, and the external traffic of the incoming packets from the outside. This type of queue is described as GI/M/1/K queue (finite buffer), as the arrival rate follows a general distribution due to the mix of traffic that has independent arrival rates. Model SFB has two Classes: *Fast Path* (FP), used by the switch for processing all packets, and *Controller Path* (CP) for all the packets processed by the controller. The Class CP in this case is modelled as M/M/1 queue (infinite queue) and has a mean service rate of μ_c, and the Class FP is modelled with a mean service rate of μ_s. Model PFB incorporates priority queuing, and each switch has two classes: *Class Slow Path* (SP), that is used to receive the packets from the controller and to forward the *Port-Status* messages, and *Class Fast Path* (FP), where a packet is only processed in case there are no packets in the Class SP. μ_s is the mean service rate of the switch that is shared by both classes. Non-preemptive priority queuing is used, meaning that Class FP is served only when Class SP has no packets. N is the number of MNs connected to a single switch, while K is the finite size of the buffer at the switch for Model SFB. For Model PFB the total queue capacity of the switch is K, Class SP has queue capacity of K_1, whereas Class FP has K_2, such that $K=K_1+K_2$. Furthermore, β was used to indicate the probability of *Packet-In* messages sent to the controller when no match is found in the flow table of the switch, and α to indicate the probability of *Port-Status* messages, which are sent from the switch indicating the start of the handover procedure.

The basic idea is to design a system that will allow to mimic a handover behavior. The relevant sequence of OF-messages exchanged due to handover signaling, is marked with red on Fig. 1 for Model SFB and for Model PFB. The sequence of events for Model SFB is designed as follows: (1) a *Port-Status (off-port)* message is received at switch 1, it is processed in the common Class FP at switch 1, and sent to the controller ($\alpha T_{fp}^{(1)}$, where T_{fp} is the throughput of Class FP); (2) the *Port-Status* message is then processed by the common Class CP, forwarded to switch 1, and an extra packet is generated to inform switch 2 about the *off-port* for the MN at switch 1 ($\alpha T_{cp}^{(2)}$, where T_{cp} is the throughput of Class CP); (3) a totally synchronized system is assumed, meaning that immediately upon receiving the *off-port* message switch 2 knows that an *on-port* message is received at Class FP at switch 2, and this message is similarly forwarded from switch 2 to the controller ($\alpha T_{fp}^{(3)}$); (4) this packet is then processed by the controller and sent to inform both switch 1 and switch 2 about the change in the network ($\alpha T_{cp}^{(4)}$); (5) after the packet of interest is processed at switch 2 (and switch 1), it is dropped ($\alpha T_{fp}^{(5)}$). The only difference in the sequence of messages of model PFB compared with model SFB, is that when the packet is returned from the controller, it now enters a specialized buffer of Class SP, and is immediately processed and sent either back to the controller, or on the external output path.

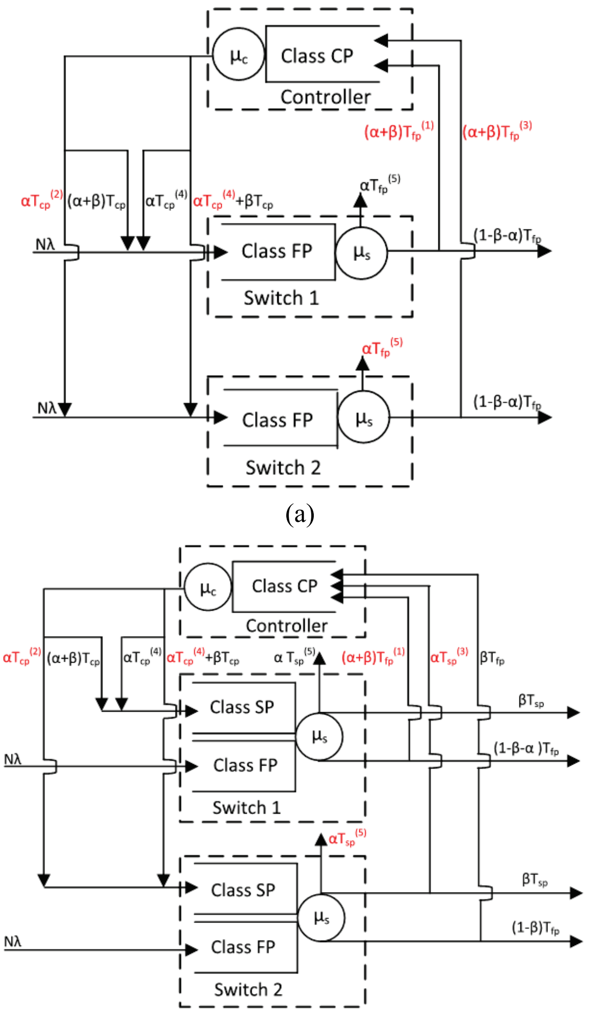


Fig. 1. Systems used for handover modelling at the switch: (a) Model SFB, (b) Model PFB

TABLE I
PERMISSIBLE TRANSITIONS FOR MODEL SFB

	from	to	rate
Packet arrives at switch 1	(x, y, z)	(x, y+1, z)	$N\lambda$
Packet arrives at switch 2	(x, y, z)	(x, y, z+1)	$N\lambda$
Packet forwarded from switch 1 to controller(1)	(x, y>0, z)	(x+1, y-1, z)	$(\alpha+\beta)\mu_s$
Packet serviced from controller to switch 1	(x>0, y, z)	(x-1, y+1, z)	μ_c
Packet serviced from controller to switch 2(2)	(x>0, y, z)	(x-1, y, z+1)	$\alpha\mu_c$
Packet forwarded from switch 2 to controller(3)	(x, y, z>0)	(x+1, y, z-1)	$(\alpha+\beta)\mu_s$
Packet serviced from controller to switch 1(4)	(x>0, y, z)	(x-1, y+1, z)	$\alpha\mu_c$
Packet serviced from controller to switch 2(4)	(x, y, z>0)	(x-1, y, z+1)	μ_c
Packet dropped at switch 1(5)	(x, y>0, z)	(x, y-1, z)	$\alpha\mu_s$
Packet dropped at switch 2(5)	(x, y, z>0)	(x, y, z-1)	$\alpha\mu_s$
Packet departs from switch 1	(x, y>0, z)	(x, y-1, z)	$(1\text{-}\beta\text{-}\alpha)\mu_s$
Packet departs from switch 2	(x, y, z>0)	(x, y, z-1)	$(1\text{-}\beta\text{-}\alpha)\mu_s$

The SFB system is architected as a continuous time Markov chain $\{n_{cp}(t), n_{fp1}(t), n_{fp2}(t)$, where $t > 0\}$, where $n_{cp}(t)$, $n_{fp1}(t)$, and $n_{fp2}(t)$, represent the number of packets in the controller, switch 1, and switch 2, respectively. X, y, and z present a set of values for $n_{cp}(t)$, $n_{fp1}(t)$, and $n_{fp2}(t)$, where $x \in Z_+$, $y \in Z_+^{(\leq K)}$, and $z \in Z_+^{(\leq K)}$. The allowed transitions of the Markov process are given in Table I. The PFB system is similarly architected as a continuous time Markov chain $\{n_{cp}(t), n_{sp1}(t), n_{fp1}(t), n_{sp2}(t), n_{fp2}(t)$ where $t > 0\}$, where $n_{cp}(t)$, $n_{sp1}(t)$, $n_{fp1}(t)$, $n_{sp2}(t)$, and $n_{fp2}(t)$ represent the number of packets in the controller, switch 1 (Class SP), switch 1 (Class FP), switch 2 (Class SP), and switch 2 (Class FP), respectively. V, w, x, y, z present a set of values for $n_{cp}(t)$, $n_{sp1}(t)$, $n_{fp1}(t)$, $n_{sp2}(t)$, and $n_{fp2}(t)$, where $v \in Z_+$, $w \in Z_+^{(\leq K1)}$, $x \in Z_+^{(\leq K2)}$, $y \in Z_+^{(\leq K1)}$, $z \in Z_+^{(\leq K2)}$. The allowed transitions of the Markov process are given in Table II.

TABLE II
PERMISSIBLE TRANSITIONS FOR MODEL PFB

	from	to	rate
Packet arrives at switch 1, Class FP	(v, w, x, y, z)	(v, w, x+1, y, z)	$N\lambda$
Packet arrives at switch 2, Class FP	(v, w, x, y, z)	(v, w, x, y, z+1)	$N\lambda$
Packet forwarded from switch 1 to controller, Class FP(1)	(v, 0, x>0, y, z)	(v+1, 0, x-1, y, z)	$(\alpha+\beta)\mu_s$
Packet serviced from controller to switch 1, Class SP	(v>0, w, x, y, z)	(v-1, w+1, x, y, z)	μ_c
Packet serviced from controller to switch 2, Class SP(2)	(v>0, w, x, y, z)	(v-1, w, x, y+1, z)	$\alpha\mu_c$
Packet forwarded from switch 2 to controller, Class SP(3)	(v, w, x, y>0, z)	(v+1, w, x, y-1, z)	μ_s
Packet forwarded from switch 2 to controller, Class FP	(v, w, x, 0, z>0)	(v+1, w, x, 0, z-1)	$\beta\mu_s$
Packet serviced from controller to switch 1, Class SP(4)	(v>0, w, x, y, z)	(v-1, w+1, x, y, z)	$\alpha\mu_c$
Packet serviced from controller to switch 2, Class SP(4)	(v>0, w, x, y, z)	(v-1, w, x, y+1, z)	$(1+\alpha)\mu_c$
Packet dropped at switch 1, Class SP(5)	(v, w>0, x, y, z)	(v, w-1, x, y, z)	$\alpha\mu_s$
Packet dropped at switch 2, Class SP(5)	(v, w, x, y>0, z)	(v, w, x, y-1, z)	$\alpha\mu_s$
Packet departs from switch 1, Class FP	(v, 0, x>0, y, z)	(v, 0, x-1, y, z)	$(1\text{-}\beta\text{-}\alpha)\mu_s$
Packet departs from switch 2, Class FP	(v, w, x, 0, z>0)	(v, w, x, 0, z-1)	$(1\text{-}\beta)\mu_s$
Packet departs from switch 1, Class SP	(v, w>0, x, y, z)	(v, w-1, x, y, z)	μ_s
Packet departs from switch 2, Class SP	(v, w, x, y>0, z)	(v, w, x, y-1, z)	μ_s

In a small network operator (with ~ 20 Gbps total network throughput), the number of handover requests is around 40 pkt/sec on network level (data taken from a real mobile operator's statistics) for around 200k simultaneously active users (with ongoing sessions). If a small throughput for a single MN of ~ 1Mbps (for MTU (Maximum Transmission Unit) of 1500 Bytes, this is 83 pkt/sec) is considered, for the smallest value of β = 0.01 and one MN, this results in 0.83 pkt/sec. So, for a single MN, when comparing the worst case of 0.83 pkt/sec for the *Packet-In* messages, and an average of 0.0002 pkt/sec (40 pk/sec divided by 200k users) for the *Port-Status* messages, it is obvious that $\alpha << \beta$ and $\alpha << 1$. Therefore, $(\alpha+\beta)\mu_s \sim \beta\mu_s$, $(1-\beta-\alpha)\mu_s \sim (1-\beta)\mu_s$, and $(1+\alpha)\mu_c \sim \mu_c$ is considered. These approximations are used when calculating the sub-matrices of the QBD process, and for the permissible states in Table I and Table II to define the sub-matrices of the transition rate generator matrix *Q*. The matrices are then input to Matrix-Analytic Methods (MAM) to compute the stationary distribution probabilities. The details of generating the submatrices are explained in [10] and will not be repeated here.

The main performance metric of interest is the packet loss rate. A new parameter called relative packet loss gain (Pl_{rel}) between the Model SFB and Model PFB is defined. If Pl_{rel} has a positive value, it means that Model PFB has lower packet loss rate compared to Model SFB. The relative packet loss gain is calculated as:

$$Pl_{rel} = \frac{(Pl_{tot}^{sfb} - Pl_{tot}^{pfb})}{Pl_{tot}^{sfb}} \times 100\%, \quad (1)$$

where the total packet loss rate for Model SFB (Pl_{tot}^{sfb}) is the sum of the packet loss rates of the Class FP in both switches, whereas the total packet loss rate for Model PFB (Pl_{tot}^{pfb}) is the sum of the packet loss rates of the Class SP and Class FP in both switches, both are given with the following equations:

$$Pl_{tot}^{sfb} = Pl_{fp1}^{sfb} + Pl_{fp2}^{sfb}, \quad (2)$$

$$Pl_{tot}^{pfb} = Pl_{fp1}^{pfb} + Pl_{fp2}^{pfb} + Pl_{sp1}^{pfb} + Pl_{sp2}^{pfb}. \quad (3)$$

The average packet loss rate indicates the average number of packets that are blocked by any of the respective classes out of the total incoming packets. For Model SFB, for the Class FP/Switch 1 (Pl_{fp1}^{sfb}), and Class FP/Switch 2 (Pl_{fp1}^{sfb}), and for the Model PFB, for the Class FP/Switch 1 (Pl_{fp1}^{pfb}), Class SP/Switch 1 (Pl_{sp1}^{pfb}), Class FP/Switch 2 (Pl_{fp2}^{pfb}), Class SP/Switch 2 (Pl_{sp1}^{pfb}), the loss rates can be expressed as follows:

$$Pl_{fp1}^{sfb} = 1 - \frac{T_{fp1}^{sfb}}{N\lambda}, \quad (4)$$

$$Pl_{fp2}^{sfb} = 1 - \frac{T_{fp2}^{sfb}}{N\lambda}, \quad (5)$$

$$Pl_{fp1}^{pfb} = 1 - \frac{T_{fp1}^{pfb}}{N\lambda}, \quad (6)$$

$$Pl_{fp2}^{pfb} = 1 - \frac{T_{fp2}^{pfb}}{N\lambda}, \quad (7)$$

$$Pl_{sp1}^{pfb} = 1 - \frac{T_{sp1}^{pfb}}{T_{cp}^{pfb}}, \quad (8)$$

$$Pl_{sp2}^{pfb} = 1 - \frac{T_{sp2}^{pfb}}{T_{cp}^{pfb}}. \quad (9)$$

The throughput of each Class is the sum of probabilities that the Class has at least one packet to forward with the mean service rate of μ_s for the switch, and μ_c for the controller. T_{fp1}^{sfb}, T_{fp2}^{sfb}, T_{cp}^{sfb} is the throughput of Model SFB for Class FP/Switch 1, Class FP/Switch 2, and Class CP, respectively. Similarly, T_{fp1}^{pfb}, T_{sp1}^{pfb}, T_{fp2}^{pfb}, $T_{sp2}^{pfb}T_{cp}^{pfb}$, is the throughput of Model PFB for Class FP/Switch 1, Class SP/Switch 1, Class FP/Switch 2, Class SP/Switch 2, and Class CP, respectively.The equations for the Class throughput were mathematically derived in detail in [3] and will not be repeated here.

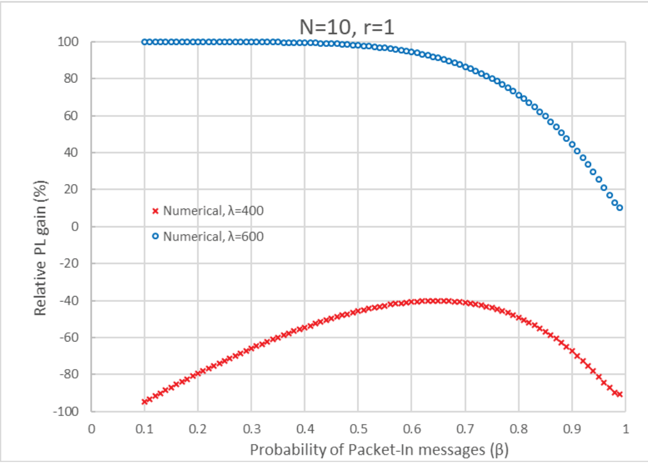
IV. Results

Next, the packet loss rate in both proposed systems, Model SFB and Model PFB, is compared. The parameters are described in Table III. The probability of Packet-In messages is in the range of 0.1 to 1, the controller-to-switch service rate, $r = \mu_c/\mu_s$, varies from 0.1 to 2, μ_s is set at 10000 pkt/s. The arrival rate of the packets per MN is 400 pkt/s and 600 pkt/s, and the maximum transmission unit (MTU) is 1500 Bytes. The numerical analysis is performed in MATLAB.

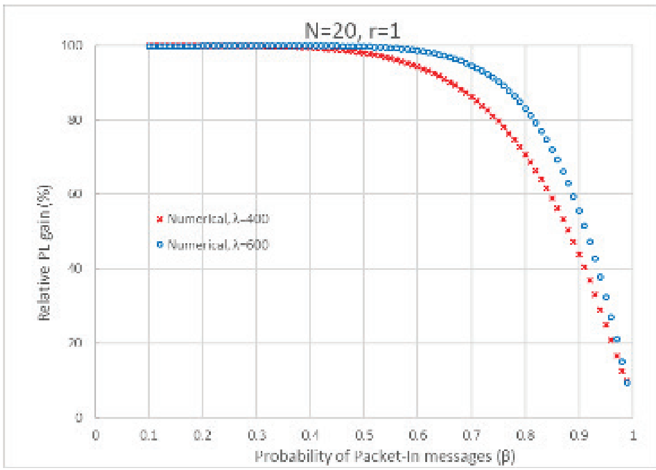
TABLE III
PARAMETER SETTING

Parameter	Value
Probability of Packet-In messages, β	0.1 - 1
Mean service time of the switch, μ_s (pkt/sec)	10000
Controller to switch mean service rate, *r*	0.1 - 2
Arrival rate, λ (pkt/sec)	400, 600
Maximum Transmission Unit, MTU (Bytes)	1500
Number of mobile nodes per switch, *N*	1-40

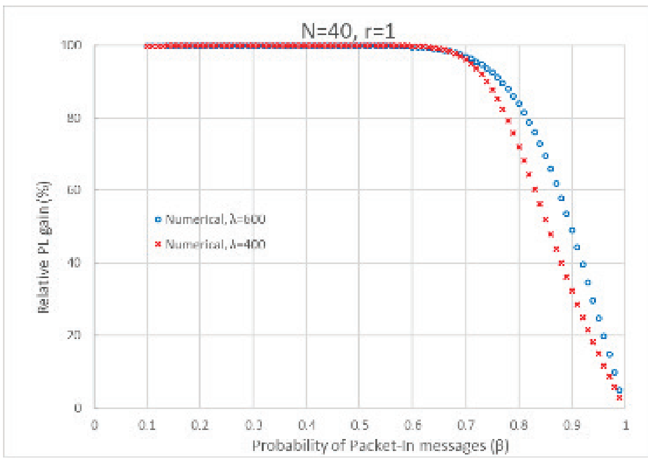
In Fig. 2 the relative *PL* gain is analyzed, as defined in Eq. (1). β varies on the *x*-axis, whereas *r* is fixed to 1. The difference between Fig. 2a, Fig. 2b and Fig. 2c, is that the number of MN changes with $N = 10$, $N = 20$, and $N = 40$, respectively. When analyzing Fig. 2a, it can be concluded that for lower values of β, for the case of $\lambda = 600$, Model PFB exhibits almost 100% reduction in the average packet loss rate, when compared to Model SFB. For values of β higher than 0.6, this *PL* reduction decreases. In Fig. 2a, for $\lambda = 400$, negative values for the packet loss gain can be observed, which basically means that for all β, Model SFB exhibits better results. Interestingly, as β increases for the lower curve in Fig. 2a, the improvement when compared to Model PFB decreases, but only up to $\beta = 0.65$, afterwards the *PL* of Model SFB again improves when comparing to Model PFB. The graph in Fig. 2b however, for $N = 20$, shows a different result. Again, for $\lambda = 600$, the reduction in *PL* of Model PFB is almost 100% for lower β, but the same applies also for $\lambda = 400$. The difference is that for $\lambda = 400$, after $\beta = 0.5$, there is a decrease in the relative *PL* gain, whereas for $\lambda = 600$, this decrease starts at $\beta = 0.6$. So, for Fig. 2b, Model PFB exhibits better results. Finally, Fig. 2c is analyzed, where $N = 40$. For all β values, and both arrival rates, Model PFB shows better performance. The relative *PL* gain of almost 100% decreases at higher values of β, at $\beta = 0.7$ the gain tends to deteriorate. To conclude, Model PFB has lower average packet loss rate compared to Model SFB in almost all scenarios of interest. As the number of MNs increases, this improvement is highly dominant even for higher β values, and generally as the arrival rate becomes higher, Model PFB has superior performance when compared to Model SFB. The only case where Model SFB shows better results is when the arrival rate and the number of MNs is low, however for mobile networks, the very opposite is true.



(a)

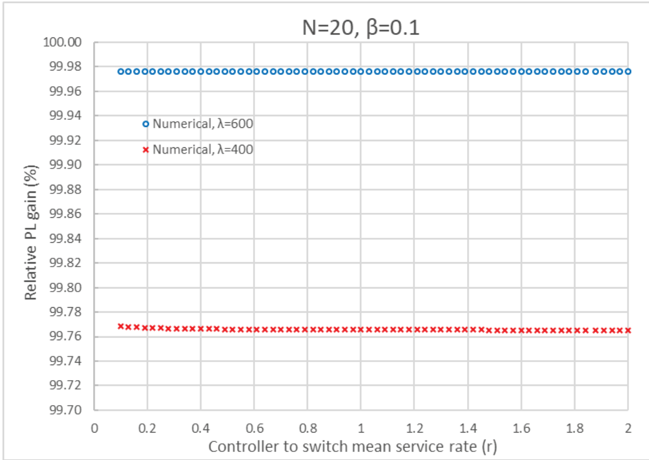


(b)

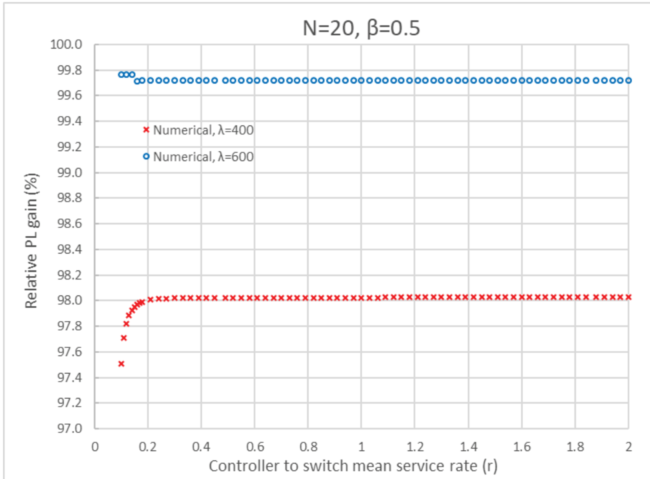


(c)

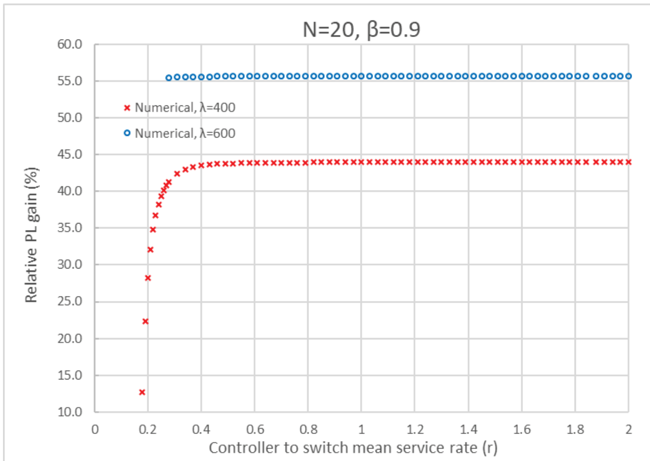
Fig. 2. Relative packet loss gain for: (a) $N = 10$, (b) $N = 20$, (c) $N = 40$. ($r = 1$)



(a)



(b)



(c)

Fig. 3. Relative packet loss gain for: (a) β = 0.1, (b) β = 0.5, (c) β = 0.9 ($N = 20$)

Now in Fig. 3, r is varied on the x-axis, and β is increased in Fig. 3a, Fig. 3b, and Fig. 3c by using a fixed value of β = 0.1, β = 0.5, β = 0.9, respectively. By analyzing Fig. 3a, as r increases from 0.1 to 2, almost linear PL gain can be noticed, 99.8% for $\lambda = 600$, and around 99.76 for λ = 400. This means that Model PFB shows superior results for all r values and for both arrival rates of interest. In Fig. 3b, $\beta = 0.5$, similar curve shapes are noticed, but now the *PL* gain is slightly decreased to 99.7%, and 98% for $\lambda = 600$, and $\lambda = 400$, respectively. Additionally, for $\lambda = 400$, for very small values of β of around 0.1, a non-linear dependency is observed, which promptly increases from 97.4% for $r = 0.1$, to 98% for $r = 0.2$. This means that if r is very small, and for lower arrival rates, Model PFB has slightly lower *PL* gain. Similar, but much more visible behavior can be seen on Fig 3c, where $\beta = 0.9$. For values of $r = 0.18$ and $\lambda = 400$, the relative *PL* gain is 12.5%, whereas for $r = 0.4$ is around 44%. For $\lambda = 600$, such non-linear dependency is not seen, but it must be stressed that the *PL* gain in Fig. 3c, for both arrival rates, is significantly lower in comparison with Fig. 3a and Fig. 3b. The *PL* gain is now in the region of 44% to 56%, whereas previously it was above 97%. To conclude, Model PFB shows superior performance for all scenarios of interest, especially if β is not that high, and if the value of r is higher. The later only applies for the case of $\lambda = 400$ in Fig. 3b and Fig. 3c, where a non-linear curve for the very small values of r was seen. This generally means that increasing r does not have impactful performance boost, except for smaller arrival rates and the smallest r values, which is not typical for mobile networks.

V. Conclusion

In this paper the aim was to compare the average packet loss rate of two systems that deploy different OF-switch buffer designs, Model SFB that used a traditional shared buffering, and Model PFB that incorporated non-preemptive prioritization where the control packets are always prioritized. The proposed mathematical model heavily relied on queuing theory and the use of QBD processes. The numerical analysis was validated in MATLAB. The results revealed that when r and N are kept as constants, and β varied from 0.1 to 1, the packet loss rate increased as both β and the arrival rate increased. Furthermore, the Model PFB was superior as the sharp increase in the packet losses started at higher β values when compared to Model SFB. It was also proved that varying the parameter r did not have an impactful performance influence for mobile networks where β is low, the number of MNs is high, and the arrival rates are high. For future work, we plan to run simulations to verify and compare the numerical analysis.

References

[1] J.H. Cox, J. Cung, S. Donovan, et al. "Advancing Software-Defined Networks: A Survey", *IEEE Access*, vol. 5, pp. 25487-25526, 2017, doi: 10.1109/ACCESS.2017.2762291

[2] Open Networking Foundation, *OpenFlow Switch Specification*, version 1.3.1. Tech. Republic., 2012.

[3] S. Panev and P. Latkoski, "Performance Analysis of Handover Delay and Buffer Capacity in Mobile OpenFlow-based Networks", *International Journal of Communication Systems*, vol. 33, no. 15, 2020, doi: 10.1002/dac.4529

[4] M. Jarchel, et al., "Modeling and Performance Evaluation of an OpenFlow Architecture", *2011 23rd International Teletraffic Congress (ITC)*, pp. 1-7, 2011.

[5] T.-C. Yen and C.-S. Su, "An SDN-based Cloud Computing Architecture and its Mathematical Model", *2014 International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*, vol. 3, pp. 1728-1731, 2014, IEEE, doi: 10.1109/InfoSEEE.2014.6946218

[6] U. Javed, A. Iqbal, S. Saleh, et al., "A Stochastic Model for Transit Latency in OpenFlow SDNs", *Computer Networks*, vol. 113, pp. 218-229, 2017, doi: 10.1016/j.comnet.2016.12.015

[7] Y.-C. Lai, A. Ali, et al., "Performance Modeling and Analysis of TCP Connections over Software Defined Networks", *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254078

[8] K. Sood, S. Yu, and Y. Xiang., "Performance Analysis of Software Defined Network Switch using M/Geo/1 Model", *IEEE Communications Letters*, vol. 20, no. 12, pp. 2522-2525, 2016, doi: 10.1109/LCOMM.2016.2608894

[9] W. Miao, G. Min, Y. Wu, and H. Wang, "Performance Modelling of Preemption-Based Packet Scheduling for Data Plane in Software Defined Networks", *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 2015, doi: 10.1109/SmartCity.2015.48

[10] Y. Goto, B. Ng, et al., "Queueing Analysis of Software Defined Network with Realistic OpenFlow–Based Switch Model", *Computer Networks*, vol. 164, December 2019, doi: 10.1016/j.comnet.2019.106892

[11] D. Singh, B. Ng, Y.V. Lai, Y.D. Lin, W. Seah, "Full Encapsulation or Internal Buffering in OpenFlow Based Hardware Switches", *Computer Networks*, vol. 167, 2020, doi: https://doi.org/10.1016/j.comnet.2019.107033

[12] A. Mondal, S. Misra, and I. Maity, "Buffer Size Evaluation of Open-Flow Systems in Software-Defined Networks", *IEEE Systems Journal*, vol. 13, no. 2, pp. 1359-1366, June 2019, doi: 10.1109/JSYST.2018. 2820745