Report on Paper "Queueing Analysis of Software Defined Network with Realistic OpenFlow–based Switch Model"

Yunhua Zhao

December 14, 2020

Introduction

Traditional networks need switches or routers to compute the next hop and sometimes also need them to do some processing on packets, the ability of switches or routers matter the loss of packets, Qos of the network...Comparing to the traditional networks, Software Defined Networking (SDN) is the latest network architecture that does for networking, SDN releases switches and routers, it separate the control plane and data plane, the control plane does the most of computing part, switches and routers just need to follow the table which the control plane assigned. There are several protocols used in SDN, Openflow is the most widely used one.

To date, analytical modeling research for SDN typically uses two different approaches, queuing theory and network calculus. Proper adaptation of queuing theoretic modeling techniques and analytical methods can contribute to accurate performance evaluation of SDN-based production networks that are exposed to new user demands continually, facilitating early identification of potential traffic hotspots and/or bottlenecks which carriers and network providers can quickly remedy before they become problems. This paper proposed a queuing model of an OpenFlow–based software defined network that aims to model the operation of the network devices as accurately as possible.

QUEUING MODEL

A critical functionality that the paper addressed by queuing analytic modeling research is the different treatment of packets arriving at a switch from the controller. See figure 1.

There are two types of input for the data panel switches:

- External input with Poisson distribution (λ), called class S.
- Packet from controller, called class F.

There are two types of output for the data panel switches, it follows a geometric distribution with average $1/\beta$:

- Packet forward to the output port with probability β.
- Packet which forwarding method is not found in the flow table then sent back to the controller with a probability β, called class C.

The forwarding time of a switch follows an exponential distribution with parameter μ_1, the processing time of controller follows an exponential distribution with parameter μ_2.

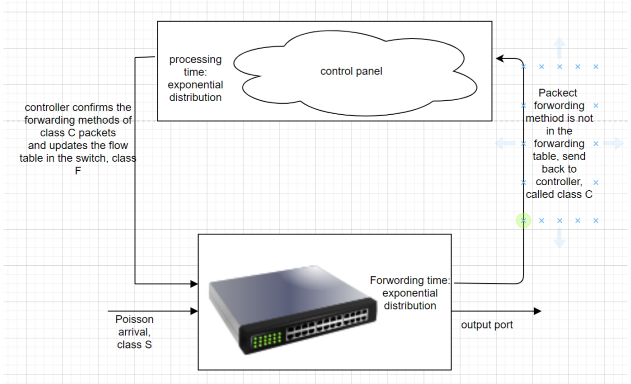

Figure 1: Queuing Model

ANALYSIS

Let $N_C(t)$, $N_F(t)$, $N_S(t)$ separately denote class C, F, S packet numbers, and meet the following condition that the input packet number has a upper bound limit.

$$0 \leqslant N_F(t) + N_S(t) \leqslant K_1, 0 \leqslant N_C(t)$$

This paper considers the stochastic process $(N_C(t), N_F(t), N_S(t)); t \geqslant 0$ with state space F is a continuous-time Markov chain.

$$F = (i, i, k) | i \in Z_+, (j, k) \in 0, 1, ..., K_1 * 0, 1, ..., K_1 - j$$

Q denotes the transition rate matrix of the Markov chain, $N_C(t) \in Z_+$ as the level variable, taking $N_F(t)$ as the second level variable, A_0, A_1, A_2 and B_1 have similar structures. The following is Q:

$$\begin{pmatrix} \mathrm{B}_1 & \mathrm{A}_0 & O & O & O & ... \\ \mathrm{A}_2 & \mathrm{A}_1 & \mathrm{A}_0 & O & O & ... \\ ... & \mathrm{A}_2 & \mathrm{A}_1 & \mathrm{A}_0 & O & ... \\ ... & O & \mathrm{A}_2 & \mathrm{A}_1 & \mathrm{A}_0 & ... \end{pmatrix}$$

Packets from controller have priority over packets from external when being processed and forwarded by the switch, so $N_C(t)$ increases by 1 then $N_F(t)$ is 0, then $N_S(t)$ decreases by 1. If $N_C(t)$ decreases by 1 and $N_F(t)$ increases by 1, $N_S(t)$ does not change; If $N_C(t)$ does not change, $N_F(t)$ does not change or decreases by 1. According to these different situation derive the state transitions that $N_C(t)$ and $N_F(t)$ remain unchanged and that $N_S(t)$ changes. The Markov chain is irreducible. Assume it is positive recurrent, so have the unique state distribution vector.

$$\pi = (\pi_{i,j,k})_{i,j,k \in F}$$

As the transition rate matrix Q is block tridiagonal, according to the Markov chain is the quasi-birth-and-death process, get the solution.

This paper considers three performance measures:

- packet loss probabilities in external input data

$$P_{loss} = 1 - \frac{\mu_1 \sum_{K_1}^{k=1} \sum_{K_2}^{i=0} \pi_{i,0,k}}{\lambda}$$

- packet loss probabilities in control panel input data

$$P_{loss} = 1 - \frac{\mu_1 \sum_{K_1}^{j=1} \sum_{K_1-j}^{k=0} \sum_{K_2}^{i=0} \pi_{i,j,k}}{\mu_2 \sum_{K_2}^{i=1} \sum_{K_1}^{k=0} \sum_{K_1-k}^{j=0} \pi_{i,j,k}}$$

- average packet transfer delay through the system

$$E[W] = \frac{\sum_{\infty}^{i=0} \sum_{K_1}^{k=0} \sum_{K_1-k}^{j=0}}{(1-\beta)\mu_1 \sum_{\infty}^{i=0} \sum_{K_1}^{k=1} \pi_{i,0,k} + \mu_1 \sum_{\infty}^{i=0} \sum_{K_1}^{j=1} \sum_{K_1-j}^{k=0} \pi_{i,j,k}}$$

NUMERICAL EXAMPLE & VALIDATION

(1) Effect of external packet arrival rate:

- A higher packet arrival rate induces a higher packet loss probability
- The average packet transfer delay decreases with increasing flow size

(2) Effect of flow size:

packet loss probability slowly decreasing as a function of increasing flow size. When average flow size gets larger while keeping external arrival rate unchanged, the arrival rate of packets from controller becomes smaller, and then the packet loss probability decreases and gradually gets less sensitive to the average flow size.

For long flows, the average packet transfer delay is insensitive to flow sizes, since the number of packets transferred to the controller is small, which alleviates queuing delays at both the switch and controller.

Conclusion

This paper proposed a queuing model to analyze an OpenFlow–based software defined network, this paper considers the different types of packets and derives average packet transfer delay in the whole system and packet loss probabilities for the input from external and controller.