Performance Evaluation for SDN Deployment: an Approach Based on Stochastic Network Calculus

LIN Changting[1], WU Chunming[1*], HUANG Min[1], WEN Zhenyu[2], ZHENG Qiuhua[3]

[1]Next Generation Network Laboratory, College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, Zhejiang Province, China

[2]Centre for Intelligent Systems and their Applications, University of Edinburgh, U.K.

[3]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang Province, China

Abstract: The OpenFlow implementations (SDNs) have been deployed increasingly on varieties of networks in research institutions as well as commercial institutions. To develop an OpenFlow implementation, it is required to understand the performance of the network. A few benchmark tools (e.g., Cbench and OFlops) can be used to measure the network performance, while these tools take considerable time to simulate traffic behaviors and generate the required results, therefore extending the development time. In this paper, we present an analytical model, which is based on stochastic network calculus theory, for evaluating the performance of switch to controller. The previous studies show that stochastic network calculus can provide realistic emulation of real network traffic behaviors. Our model is evaluated by using both simulation tool and realistic testbed. The results show the stochastic network calculus based analysis model can realistically measure the network performance of the end-to-end properties between controller and switch.

Keywords: software-defined networking; end-to-end; performance; stochastic network calculus

I. INTRODUCTION

The traditional networks switch functions are assembled in one equipment: the control and data planes are bundled together. Moreover, it lacks real-time reconfigurability and adaptivity. Software-Defined networking (SDN) [1][2] is an emerging network architecture which breaks the vertical integration by separating control and data planes. The main argument for this approach is that it provides a more structured software environment for developing network-wide abstractions while potentially simplifying the data plane.

OpenFlow is the first successful protocol [3] used in SDN, which can get access to the network-forwarding plane and provide common and simple APIs for network device control. The majority of OpenFlow and SDN frameworks have been studied in the recent years, and there are several SDNs from hardware and software vendors being deployed in different areas, such as enterprise, research community, data centers and so on. Further, more and more SDNs will be required for the next generation network in future, thereby requiring a growing number of performance experiments over SDN-enabled networks. Most existing researches focus on the SDN frameworks, however, few researches investigate the SDN performance. In practical terms, understanding the performance of SDN concept is one of key requirements for designing an SDN deployment.

There are some benchmarks or simulation

tools, such as OFLOPS [4], Cbench [5], can measure the performance of the network system to provide useful information for designing SDNs. By understanding the performance of devices, the network architects can make better design for building a robust network. However, these tools may take a considerable time and resources to generate the required results. Therefore, more convenient and faster performance measure fashion is desired for network architects, especially, who use SDN for designing a network.

To this end, analytical model can be an alternative way that can simplify a deployment and quickly predict performance by analytically modelling the behavior of SDN system. Based on queuing theory for the forwarding speed and blocking probability of current OpenFlow switches, analytical model [6] has been used to estimate the performance of the network. However, this model is only concerned with the average value under steady state. Further, a deterministic Network calculus-based model [7] is proposed to evaluate the performance of an SDN deployment, but this approach is inadequate to deal with the stochastic burstiness in real network traffic.

In this work we utilize stochastic network calculus as a mathematical method to analyse the behavior of an SDN system performance. The proposed model in this paper evinces the interaction between OpenFlow switches and SDN controller. We highlight the contributions of this paper as follows:

An analytical model to measure performance of SDN deployment, which is based on stochastic network calculus.

The evaluation of our model, using testbed, emulation and numerical calculus to verify this model.

The remainder of the paper is organized as follows. In Section II, we review the related studies. The following is the network calculus theory for adapting to our model. In Section IV, we introduce an analytical model that describes the delay bound of switch-controller end-to-end delay in SDN deployment. Before drawing a conclusion, the evaluation of the proposed model is discussed in Section V

II. Related Work

Regarding to the performance of the SDN and OpenFlow, there are very limited literatures focusing on it. The most common method of measure or evaluate the performance of SDN system is using benchmark tools.

OFLOPS [4] is a benchmark tool on the controller side that is used to measure the performance of OpenFlow-enabled software and hardware switches. A modular framework was implemented in the tool for adding and running implementation-agnostic tests to quantify the performance of a switch.

Cbench [5] is a program which generates for testing OpenFlow controllers by generating packets in events for new flows. Cbench emulates a bunch of switches that connect to a controller, sends *packet_in* messages, and watches for *flow_mod* to get pushed down. However, Cbench is a single-threaded tool, therefore implementing multiple jobs needs a multi-core CPUs. In addition, Cbench only builds connections between one controller and a large scale of switches, which means that aggregated statistics are gathered for all switches instead of each individual switch.

OFCBenchmark [8] is the latest benchmark tool, which was designed to be able to create a set of message-generating virtual switches. Each switch can be configured independently from each other to simulate a specific scenario, at the same time keeping its own statistics.

As we have discussed previously, benchmark tools would take considerable time to simulate and generate results. Therefore a performance indicator is required, which can quickly provide potential scalability bottlenecks for an OpenFlow switch-controller system before detailed data is available, especially at the initial stage. To this end, some researchers proposed the analytical model, mathematical calculate the network performance.

Tootoonchian and Gorbunov [6] proposed an analytical model to estimate the performance of the network by using queue theory to simulate

the forwarding speed and blocking probability of current OpenFlow switches. In general, the queuing theory characterizes arrival process and service process from which mean queuing length and mean system delay can be derived. However, the queuing theory has so many limitations that it cannot capture all features of traffic. Because it only concerns the average quantities under the steady state. Moreover, it is hard to deduce concatenation property and leftover service in queuing theory. This partly explains why it is difficult to apply traditional queuing theory to modern network analysis.

Jiang and Liu [9] introduced a network calculus (deterministic network calculus and stochastic network calculus) method to simulate the nature of traffic arrival and service curve. This approach can undercover the worst case instead of average behavior, and easier to apply than benchmark to any SDN deployment.

The article [7] is the most related work. Although the Network calculus-based model is proposed to evaluate the performance of SDN deployment, the proposed method is not very realistic. According to the hypothetical arrival process and the service process, the analytical model can capture the delay and buffer sizing inside SDN switches. The model based on deterministic network calculus is not concerned with stochastic behavior of traffic and service. However, the behavior of real network traffic is stochastic burstiness.

Zhang and Liu [10] used stochastic network calculus to model software-defined networks. However, it concentrated on wireless virtual networks. The article proposed a simple model with a GPS Scheduler for describing the upper bound delay of a wireless virtual network in the context of SDN. It provided a new clue to evaluate the performance of SDN deployment.

To account for the stochastic network, we have developed a statistical multiplexing gains model to analyse the stochastic characteristics of traffic and service processes. Our method has been proofed more suitable for real network in [9], compared to deterministic network calculus.

III. Network Calculus Theory

Network calculus is a theory which deals with the queuing systems in computer networks. Moreover, this theory focuses on performance guarantees. The main idea of the theory is to transform complex network systems into analytically tractable systems by using alternate algebra. To simplify the analysis, another idea is to characterize traffic and service processes using various bounds. Deterministic network calculus is a simple method which is not concerned with the stochastic behavior of traffic and service. However, the real network traffic is stochastic burstiness, the deterministic network calculus is not able to describe this feature very well.

To put it differently, if an independent case analysis adopts stochastic network calculus, the results will be better than the results from the deterministic network calculus. Stochastic network calculus can derive the QoS performance bounds such as the delay bound by stochastic arrival curve and stochastic service curve. In this following part, we will give a brief background of Stochastic network calculus. A comprehensive overview and outlook of stochastic network calculus can be founded in [9] and [11].

3.1 Stochastic arrival curve

The stochastic arrival curve means that the amount of packets generated by a flow in a time interval in a system. A cumulative arrival process A(t) is a non-decreasing, which is valued with non-negative integer Z+ such that A(0)=0. $A(t)$ denotes the total number of arrivals (i.e., events) in time slots 1,2,...,t.

A flow has a stochastic arrival curve $\alpha \in F$ with bounding function $f \in \overline{F}$, denoted by $A \sim <f,\alpha>$, for all $0 \leq s \leq t$ it holds:

$P(A(s,t)-\alpha(t-s)) \leq f(x)$.

3.2 Stochastic service curve

The stochastic service curve model is generalized from the deterministic service curve model which bases on its duality principle. A system

provides a stochastic service curve $\beta \in F$ with bounding function $g \in \overline{G}$, denoted by $S \sim <g,\beta>$.

3.3 Service guarantee

The service guarantees property means that the system performance bounds such as the delay bound and backlog bound can be derived under the given traffic model and server model. In this paper, our mainly concern is the delay bound. Figure 1 shows the backlog and delay based on network calculus can be presented intuitively.

Backlog

Consider a system with arrival process ($A(t)$), service process ($S(t)$), and departure process ($D(t)$). By definition, the backlog in the system at $[t,+\infty)$ is:

$$B(t) = \sup_{0 \le s \le t}(A(s,t) - S(s,t))$$

where sup(x) means supremum.

Delay

According to the definition, delay $W(t)$ in the system at $[t,+\infty)$ is presented as below:

$W(t)=inf(\tau: A(t) \le W(t+\tau))$.

which implies that, for any $x \ge 0$, if $W(t)>x$, there must be $A(t)>D(t+x)$ since otherwise if $A(t) \le D(t+x)$ and $W(t) \le x$, that would contradict the condition $W(t)>x$. So it also can be expressed:

$P(W(t)>x) \le P(A(t)>D(t+x))$.

Consider an arrival process $A \sim <f,\alpha>$ in a system and the system provides a stochastic service curve function $S \sim <g,\beta>$. Then, for all $t \ge 0$ and $x \ge 0$, the delay $W(t)$ is bounded by:

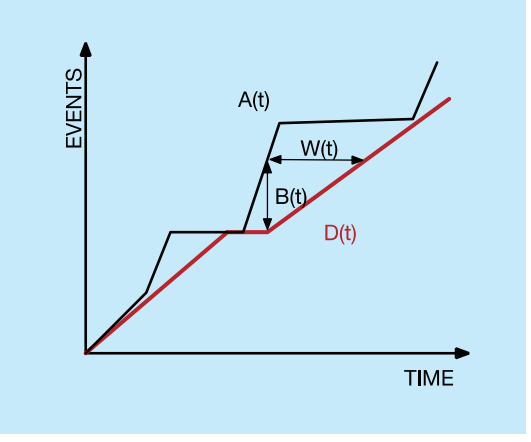

Fig. 1 *Backlog and delay features based on network calculus*

$P((W(t)>x)>h(\alpha+x,\beta)) \le f \otimes g(x)$,

where $h(\alpha+x,\beta)$ is the maximum horizontal distance between function $\alpha(t)+x$ and $\beta(t)$ for $x \ge 0$.

3.4 Leftover service

In order to provide QoS in a scalable manner, aggregate scheduling has been studied as an important approach. For such networks, to derive per-flow QoS bounds, it is desirable to study the per-flow service received by each flow in an aggregate, which is called leftover service.

Consider the case where there are two flows competing for resources in a system under aggregate scheduling. A system is fed with a flow A that an aggregation of two constituent flows A_1 and A_2. Suppose that the both service characterization from the server and traffic characterization from A_2 are given, however, we are interested in flow A_1, with which per-flow bounds for A_1 can then be easily obtained. Leftover service curves provide thus a worst-case description of service, and have the property that they are guaranteed by any work conserving scheduling mechanism.

3.5 Concatenation property

The concatenation property means that a series of servers in tandem can be considered as one single server and represented using the same server model. As the output characterization property, the concatenation of nodes can be treated as an equivalent node.

Assuming a system is tandem, and the system has two nodes which are with service curves $S_1(t)$ and $S_2(t)$. Moreover, the arrival curve is $A(t)$, so we can derive the departure of the first node is equal to the second node arrival curve. According to this method we can generalize the service net equation as below,

$S^{net}(t)=S^1 \otimes S^2 \otimes \cdots \otimes S^n$.

3.6 End-to-end property

The Internet is designed and established based on end-to-end arguments. This design supports

a unified service for any other types of data. End-to-end delay is an important parameter in network performance. High end-to-end latency adversely affects the performance of time-sensitive applications, such as SDN recovering.

The end-to-end stochastic delay bound is analysed based on the concatenation property which shows all the network service can be modelled as a whole as if there were a single service system. With the service guarantee property and the output characterization property, the end-to-end performance of a sequence of nodes in tandem can be investigated by using the node-by-node analysis approach.

Stochastic arrival curve is $A\sim<f,\alpha>$, and each node provides a leftover stochastic service curve $S^i\sim<\beta^i,\Gamma^n>$.

Furthermore, according to the concatenation property of the stochastic service curve, we have,

$\beta_{net}(t)=\beta^1\otimes\beta^2\otimes\cdots\otimes\beta^n$.

And

$f_{net}(t)=\Gamma^1\otimes\Gamma^2\otimes\cdots\otimes\Gamma^n$

According to the delay bound theorem, we have

$P(W>h(\alpha+x,\beta_{net}))\leq f\otimes f_{net}$.

So the cumulative distribution function can be derived easily and the stochastic network calculus method can get lower bounds.

IV. Analytical Model

In this section, we will introduce an analytical model which is based on stochastic network calculus and can provide the performance of switch-controller end-to-end delay. The system of SDN includes plenty of OpenFlow switches and a controller. These switches are centralized controlled by a controller.

An analytical model based on deterministic network calculus is presented in paper [7]. The interaction between a controller and an OpenFlow switch is depicted in Figure 2. These flows in the model, the command flow $C(t)$ and the packet flow are forwarded to controller $F(t)$ as background traffic. However, the model was only concerned with interaction between one controller and one switch. Moreover, an end-to-end model was not introduced in paper and even

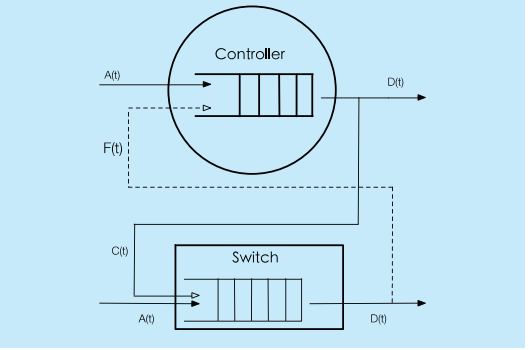

Fig.2 *A logical model of SDN controller and switch*

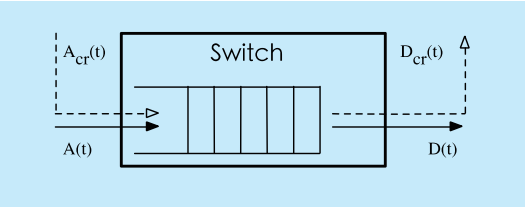

Fig.3 *A controller model with background traffic*

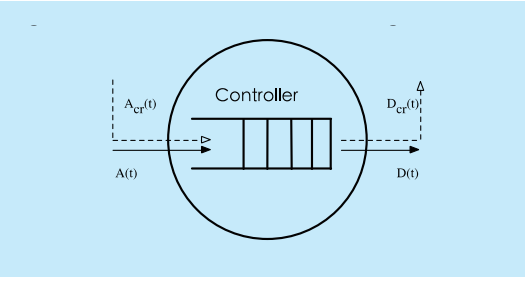

Fig.4 *A switch model with background traffic*

if the end-to-end switch-controller model was added on this model, it would be too complicated to analyse.

Thus, we simplify this model as controller/switch model with through flow and background flow. An SDN controller and a switch model can be depicted in Figure 3 and Figure 4 respectively. In an ideal SDN system, controller/switch has a through flow $A(t)$ and a background flow $A_{cr}(t)$. The $A(t)$ denotes the cumulative number of arrival by time t, and $A_{cr}(t)$ is the cumulative number of arrival background flow; $D(t)$ denotes the cumulative number of departures by time t. $D_{cr}(t)$ is the cumulative number of departure background traffic. In addition, the through flow is closely related to the background flow.

There are a lot of benchmark tools supporting evaluations of the performance of SDN deploy-

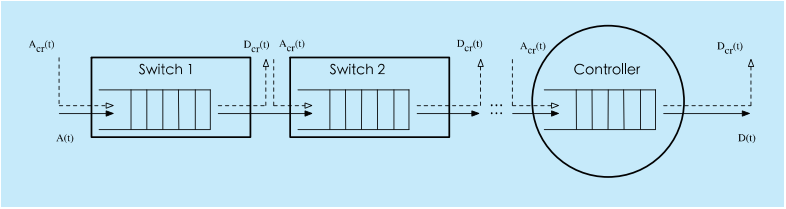

Fig.5 *End-to-End switch-controller model*

ment. However, the experiment method to measure performance is hard to implement. The analytical model method would be faster than benchmark tools and the SDN designers and operators can reference the lower bound to get a quick view of the overall SDN network deployment performance. Therefore, we introduce an end-to-end analytical model to analyse switch-controller performance.

4.1 Traffic characteristic

Nowadays, not only the traffic in network is high volume, but their characteristics are changeable. There is no single model that can be used effectively for modelling traffic in networks. Types of network under study and the traffic characteristics strictly influence the choice of the traffic model used for analysis. The traffic models are always underestimated or overestimated for real network traffic so that they cannot totally capture or describe the statistical characteristics of the actual traffic on the network.

Benson [13] collected and analysed SNMP statistics, topology, and packet-level traces, then observed, which shows that the traffic in data center that the data center traffic characteristic is small size. Besides, these are ON/OFF in nature with properties that fit Exponential and Lognormal distributions.

In this paper, we assume that the interval time between each two successive events is characterized by Exponential and Lognormal distribution; and the event is happening frequently per second. We derive the two stochastic arrival processes, respectively.

Exponential Distribution: consider the interval time obeys exponential distribution, the packet size is fixed, the packets are aggregated and these packets are independent identically distributed, so the arrival process is Poisson process. Suppose the Poisson process with mean rate λ. Then, in any time interval in $(s,s+t)$, for any $x\geq 0$, the stochastic arrival curve is:

$$P(A(s,s+t)-\lambda t>x)\leq \sum_{k=x+\lambda t}^{\infty}\frac{e^{-\lambda t}(\lambda t)^{k}}{k!}.$$

Lognormal Distribution: consider a flow with fixed unit packet size, the μ is the mean rate and σ is standard deviation. Suppose the all packets are independent identically distributed, and these lognormal packets are aggregated. An aggregate lognormal process is obtained. Then, in any time interval in $(s,s+t)$, $A(s,s+t)$ satisfies, for any $x\geq 0$, the stochastic arrival curve is:

$$P(A(s,s+t)-\rho t>x)\leq \frac{1}{2}-\frac{1}{2}\Phi(\frac{\ln(x+\rho t)-n\mu t}{\sqrt{2n}\sigma t})$$

Furthermore,

$$P(A(s,s+t)-\rho t>x)\leq \frac{1}{2}-\frac{1}{2}\Phi(\frac{\ln(x+\rho t)-n\mu t}{\sqrt{2n}\sigma t})$$

4.2 Analyze the performance based on the model

In this subsection, we use stochastic network calculus to derive the system end-to-end performance which provides the delay for a flow in SDN deployment. Based on these measurements of end-to-end delay, we characterize the behavior of controller to switch.

An SDN controller and an OpenFlow switch model are depicted in Figure 3 and Figure 4, respectively. According to concatenation property, we set up an end-to-end switch-controller model which is depicted in Figure 5. In this model, two input flows, and the through flow is what we concerned, each node has its own background flow and we can analyse this model by stochastic network calculus. The cumulative arrival process $A(t)$ represents the packet inputs from other SDN enabled switches, which are controlled by a centralized controller. Similarly, another input flow $A_{cr}(t)$ is not concerned, despite that $A_{cr}(t)$ flow is related closely to $A(t)$. $D(t)$ and $D_{cr}(t)$ are all departure flows.

In order to derive the end-to-end delay in SDN deployment, firstly we should consider an SDN controller/switch model with a stochastic service curve $S\sim<\rho,e^{-x}>$. Furthermore, we assume that

the flow arrival time interval obeys exponential distribution. According to the end-to-end delay theorem, we can derive the probability of end-to-end delay bound at time $(s,s+t)$. We have:

$$P(W > h(\alpha + x, \beta_{net})) \le \sum_{k=x+\lambda t}^{\infty} \frac{e^{-\lambda t}(\lambda t)^k}{k!} \otimes e^{-x} \otimes \ldots \otimes e^{-x}$$

When the input flow's arrival time interval obeys lognormal distribution. Also we have:

$$P(A(s,s+t) - \rho t > x) \le \frac{1}{2} - \frac{1}{2}\Phi\left(\frac{\ln(x+\rho t) - n\mu t}{\sqrt{2n}\sigma t}\right) \otimes e^{-x} \otimes \ldots \otimes e^{-x}$$

According to these formulas, we can easily derive the lower bounds of cumulative distribution function, respectively.

V. Evaluation

In order to evaluate our end-to-end delay model, our experiments were set up by using a simulation tool OMNET+ [14] and realistic testbed. Both simulation tool and realistic testbed are based on the same topology (as shown in Figure 6), and the packets were transmitted between different networks based on the OpenFlow protocol.

Figure 6 shows that our realistic testbed is composed of a set of PCs and NetFPGA-1Gs. One of NetFPGA-1G is used as a packet generator for sending out packets, and other NetFPGA-1Gs are used as OpenFlow vSwitches. Furthermore, the system capacity 1 Gbps. The aim of our model is to truthfully describe the stochastic network in SDN deployment. To verify the correctness of our model, we observed numerical results and compared them with the emulation and testbed results.

Opendaylight represents OpenFlow controller. In real network deployment, the transmission delay in testbed is inevitable, so we set the transmission delay between two nodes as 10us in the emulation. Each node's service has exponential rate burstiness and there are four nodes on the route. The size of packet was fixed and small; the interval between two successive events obeys Exponential and Lognormal distribution, respectively.

Two types of distribution were used to simulate the packet delay of switch-controller end-to-end in SDN deployment. They are exponential distribution and lognormal distribution corresponding to Figure 7 and Figure 8 respectively.

In the first experiment, we assume the inputted flows' average arrival rate is 0.03 million packets per second (mpps) for the end-to-end model. The emulation results show that the delay is longer than 65us, and aa huge part of the delay time is concentrated on 65-95us.

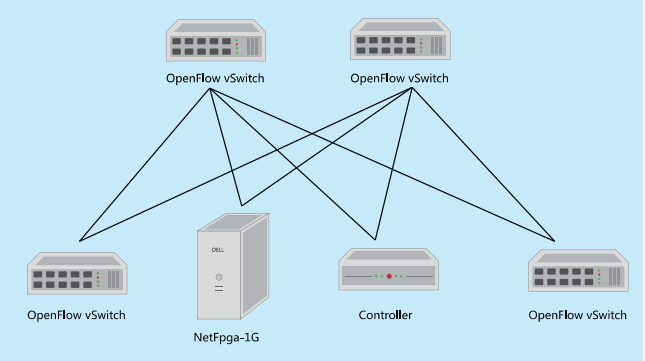

Fig.6 *The topology of the experiment*

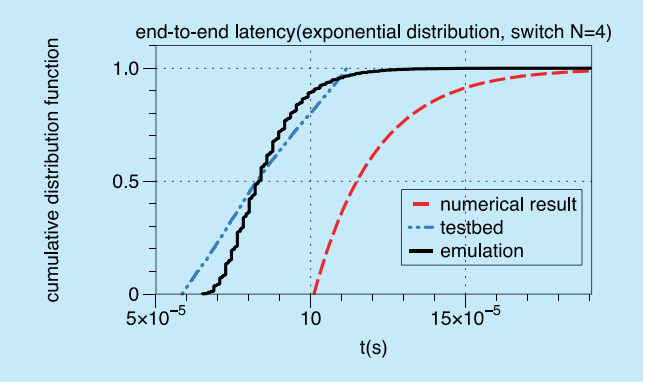

Fig.7 *End-to-End Delay with Exponential Distribution*

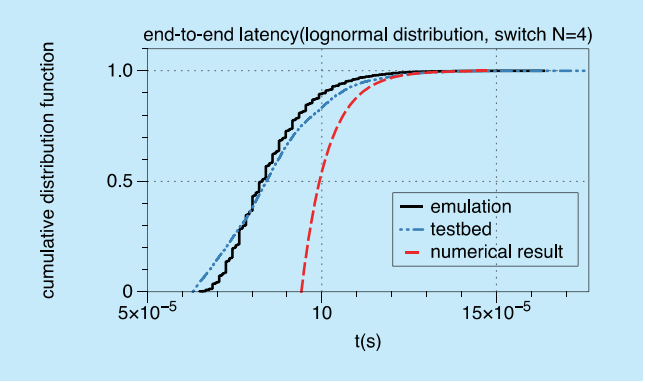

Fig.8 *End-to-End delay with lognormal distribution*

Similarly, the experiment results from testbed show that the delay time obeys uniformly distributed in the range of 50-110 us. As Figure 7 indicates, the results of numerical method, which was generated from our approach, always has more delay time for any possibility. Therefore, the numerical results can be the lower bounds of end-to-end delay. As for the second experiment, the spacing between each event follows lognormal distribution. As shown in Figure 8 the average arrival rate of the two flows is also 0.03mpps. The emulation results show that the delay is longer than 65us, and a huge part of delay time is concentrated on 65-90us. The experiment results from testbed show that the delays are distributed mainly within 62-110us. The numerical results are within 95-190us. The numerical results also show that values generated from stochastic network calculus method can be the lower bounds for target. In addition, the results from both the testbed and emulation are very close, which can verify the correctness of our implementations. Further, it proves that our model can quickly and correctly generate the lower bounds of switch-controller end-to-end packet delay in SDN deployment.

VI. Conclusion

Although there are some benchmark tools which can evaluate the performance of SDN deployment, most of them have limitations. For example, a majority of benchmark tools are very time consuming for generating results, while analytical models can be utilized to quickly predict the performance of SDN for any type of SDN deployments.

In this paper, we explore the stochastic network calculus method to analyse and evaluate the performance of SDN deployment. Compared with deterministic network calculus based methods, our work focuses on dealing with stochastic input flows which are more similar with real network traffic behaviors. In order to evaluate our method, SDN deployment is implemented on both simulated tool and real testbed. The correctness of the proposed SDN deployment and the end-to-end delay of switch to controller is verified by comparing the experiment results with the results from our model. The later was always guaranteed as the lower bound. Therefore, the model based on stochastic network calculus is vindicated, and the SDN designers and operators can reference the end-to-end model based on stochastic network calculus approach to get a quick view of the overall SDN network deployment performance.

Acknowledgment

This work is supported by the National Basic Research Program of China (2012CB315903), the Program for Key Science and Technology Innovation Team of Zhejiang Province (2011R50010-21,2013TD20), 863 Program of China (2015AA015602, 2015AA016103), and the National Natural Science Foundation of China (61379118), and the Research Fund of ZTE Corporation, and the Fundamental Research Funds for the Central Universities.

Reference

[1] Yeganeh S H, Tootoonchian A, Ganjali Y. On scalability of software-defined networking[J]. Communications Magazine, IEEE, 2013, 51(2): 136-141.

[2] Kim H, Feamster N. Improving network management with software defined networking[J]. Communications Magazine, IEEE, 2013, 51(2): 114-119.

[3] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[4] Rotsos C, Sarrar N, Uhlig S, et al. OFLOPS: An open framework for OpenFlow switch evaluation[C]// Passive and Active Measurement. Springer Berlin Heidelberg, 2012: 85-95.

[5] Tootoonchian A, Gorbunov S, Ganjali Y, et al. On controller performance in software-defined networks[C]//USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE). 2012, 54.

[6] Jarschel M, Oechsner S, Schlosser D, et al. Modeling and performance evaluation of an Open Flow architecture[C]//Proceedings of the 23rd international teletraffic congress. International Teletraffic Congress, 2011: 1-7.

[7] Azodolmolky S, Nejabati R, Pazouki M, et al. An analytical model for software defined networking:

A network calculus-based approach[C]//Global Communications Conference (GLOBECOM), 2013 IEEE. IEEE, 2013: 1397-1402.
[8] Jarschel M, Lehrieder F, Magyari Z, et al. A flexible OpenFlow-controller benchmark[C]//Software Defined Networking (EWSDN), 2012 European Workshop on. IEEE, 2012: 48-53.
[9] Jiang Y, Liu Y. Stochastic network calculus[M]. Heidelberg: Springer, 2008.
[10] [10] Zhang L, Liu J, Yang K. Virtualized Wireless SDNs: Modelling Delay Through the Use of Stochastic Network Calculus[J]. ZTE Communications, 2014 (2).
[11] Ciucu F. Scaling properties in the stochastic network calculus[D]. University of Virginia, 2007.
[12] Saltzer J H, Reed D P, Clark D D. End-to-end arguments in system design[J]. ACM Transactions on Computer Systems (TOCS), 1984, 2(4): 277-288.
[13] Benson T, Akella A, Maltz D A. Network traffic characteristics of data centers in the wild[C]//Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM, 2010: 267-280.
[14] Pongor G. Omnet: Objective modular network testbed[C]//Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems. Society for Computer Simulation International, 1993: 323-326.

Biographies

LIN Changting is currently a Ph.D. candidate at Zhejiang University, Zhejiang, China. He received the B.E. and M.S. degree in Communication Engineering from Zhejiang Gongshang University in 2009 and 2012, Zhejiang, China. His research interests include Software Defined Network, reconfigurable and network security.

WU Chunming received the Ph.D. degree in Computer Science from Zhejiang University in 1995, and currently is a Professor of College of Computer Science at Zhe jiang University. His research fields include network architecture, reconfigurable network, network virtualization, and network security. *The corresponding author. Email: wuchunming@zju.edu.cn

HUANG Min is a B.S. candidate at the College of Computer Science, Zhejiang University. His current research interests include software-defined networks and network security.

WEN Zhenyu received the B.E. degree in computer science and technology from Zhejiang Gongshang University, Zhejiang, China, in 2009, and the M.S and Ph.D. degree in computer science from Newcastle University, Newcastle Upon Tyne, U.K., in 2011 and 2015. He is currently a PostDoctoral Researcher with the School of Informatics, the University of Edinburgh, Edinburgh, U.K. He has authored a number of research papers in the field of cloud computing. His current research interests include Multi-objectives optimisation, Crowdsources, Artificial Intelligent and Cloud computing.

ZHENG Qiuhua received the Ph.D. degree from Zhejiang University in 2007, and currently is a lecturer College of Computer Science at Hangzhou Dianzi University. His research interests include cyber security, security protocols analysis and network management.