Scalability of Control Planes for Software Defined Networks:Modeling and Evaluation

Jie Hu*, Chuang Lin*, Xiangyang Li*†, Jiwei Huang*
*Department of Computer Science and Technology, Tsinghua University
†Department of Computer Science, Illinois Institute of Technology
Email: jiehu1990@gmail.com, chlin@tsinghua.edu.cn, xli@cs.iit.edu, huangjw05@gmail.com

***Abstract*—With the increasing popularity of Software defined network (SDN), designing a scalable SDN control plane becomes a critical problem. An effective approach to improving the scalability is to design distributed architecture of SDN control plane. However, how to evaluate the scalability of SDN control planes remains unexplored. In this paper, we propose a metric of scalability for SDN control planes, and study three typical SDN control plane structures, including centralized, decentralized and hierarchical architectures. We build performance models for response time, based on which we evaluate the scalability of these three structures. Furthermore, the comparison between different architectures are analyzed by mathematical methods. Numerical evaluations are also conducted to validate the conclusions drawn in this paper.**

I. Introduction

With the development of the Internet, the traditional network is approaching a bottleneck in network control. To tackle the challenge, a new network architecture named SDN is proposed which moves the control plane out of data plane and reduces the management complexity significantly. Taking advantage of centralized control, the controller for SDN is customized, programmable, and flexible. However, the centralized control faces some challenges in large scale networks. The first SDN controller (NOX) [1] can serve only 30,000 flow initiation requests per second while keeping the response time less than 10ms. It is challenge to serve more flows while keeping the response time within a reasonable small duration. Thus, understanding and quantifying the scalability of the SDN controller is a critical problem for successful adoption of SDN for large scale networks or networks with many flows. A SDN control plane is *scalable* if the control plane architecture will adapt to handle more network event requests with the increasing complexity and scale of network while satisfying the quality of service.

In order to improve the scalability of SDN control planes, two approaches were proposed [1]. The first is to move some control function of the controller to switches, reducing the event requests submitted to the controller significantly. Thus, the scalability of the controller is improved. However, the switches may need highly specialized application-specific integrated circuits(ASICs) to satisfy these requirements, and general-purpose CPUs may also be required to supply some control function. Therefore, this approach needs to be embraced by major manufacturers and thus increases the difficulty to deploy the SDN. The other one is designing a distributed control plane architecture which can distribute the load among multiple controllers. These controllers behave just like a distributed computing system, handling the incoming network requests together via communication and cooperation. And this architecture does not have special requirements of switches. Thus, in this work we will focus on understanding and quantifying the scalability of the distributed control plane for SDN.

978-1-4799-0913- 1/14/$31.00 © 2014 IEEE

To the best of our knowledge, previously these is no work providing a scalability metric for SDN control plane. And the performance of SDN control plane is mostly studied by experiments without mathematical models. Therefore, in this paper, we focus on the modeling of the scalability of SDN control planes, and a scalability metric is also proposed which may provide some insights into the construction of the SDN control plane.

Generally, the architecture of SDN control planes can be classified into three categories: *centralized*, *decentralized* (peer to peer) and *hierarchical structures* [2]. Decentralized structures have two strategies: local view strategy and global view strategy. Decentralized and hierarchical structures are also called *distributed* structures.We use stochastic modeling to analyze these four types of SDN control planes (here we treat two strategies in decentralized controllers as two different types). Firstly, the scalability of the four types of control planes are compared. We conclude that the hierarchical and the decentralized structures with local view have nearly the same scalability when the number of network hosts is far greater than the number of controllers. And they have the best scalability among these four types of SDN control planes. Secondly, we discuss how the average distance of the controller affects the scalability of SDN and conclude that as the distance increases, the scalability of control plane degrades.

II. Problem Formulation and Preliminary

A. SDN and its scalability problems

SDN moves the control plane out of data plane and uses only one controller in the beginning phase. But as the size of the network scales up, the centralized architecture can not meet the demand. There are two directions to improve the scalability of SDN. One is to design a distributed architecture of SDN control plane, such as Hyperflow [3], Onix [4], and Kandoo

[5]. Hyperflow is a decentralized structure.The controllers are organized in flat style. Every controller in Hyperflow has a global network view. Different from Hyperflow, the network view is distributed among multiple controller instances in Onix. And, the lower tier controller and its managed network will be aggregated as a logical node in upper tiers in a hierarchy of Onix controllers. Kandoo is a hierarchical structure. There are two classes of controllers: the root controller and the local controller. The local controller has no knowledge of network-wide state. And the root controller has a global network view. The local controller manages one or a few of switches. They are the switch proxies for root controllers. And the root controller manages all the local controllers and can run the applications that need to access network-wide state.

The other direction is to offload partial workload of controller to switch, such as Diffane [6] and DevoFlow [7]. It can improve scalability to some extent. However, it needs to modify the switch hardware to meet the requirements. All of these proposals are evaluated through experiments without mathematical modeling and analysis. In this paper, we focus on modeling the scalability of the SDN control plane structures.

B. Scalability of SDN control plane

The SDN control plane is acturally a distributed system. Therefore, we can use the scalability metric of distributed system to quantify the scalability of SDN control plane. The scalability metric for distributed system is based on productivity [8]. The distributed system is scalable if the productivity is maintained as the system scale changes. In SDN control plane, the productivity $F(N)$ can be defined as follows:

$$F(k) = \varphi(N) \times \frac{T(N)}{C(N)}$$

Where, N is the number of network hosts; $\varphi(N)$ is the throughput of the control plane in processing network requests; $T(N)$ is the average response time of each request, $C(N)$ is the cost to deploy the control plane. Then,the scalability metric of SDN control plane is as follows:

Definition 1 (SDN Controller Scalability)**.** *The scalability metric for SDN control plane when the network scale varies from N_2 to N_1 is defined:*

$$\Psi(N_1, N_2) = \frac{F(N_2)}{F(N_1)} \quad (1)$$

In SDN, there are several types of network requests which the control plane needs to process, such as network view maintenance, network state update and failure recovery. However, the flow initiation is the main and basic function of SDN controller [1].Therefore, we focus on the scalability of SDN control plane in processing flow initiation in this paper. For the cost to deploy the control plane, we mainly consider the controller device purchase cost.

C. System model and architecture

1) Data plane model: We consider a network with N hosts. Let $Tf_{x,y}$ represent the time interval of two consecutive flow requests from host x to host y, where the time of flow request means the time instance when the flow sends its first packet. We assume these random variables are independent identically distributed. Each random variable is subject to exponential distribution with average value of λ.

2) Flow initiation request arrival model: The process of flow initiation is illustrated in Figure 1.When the first packet of a flow arrives at the switch, and there is no matching flow entries, a flow initiation request is generated by the switch and is sent to the controller later. The controller needs to create flow entries for this flow and will send these flow entries to corresponding switches simultaneously. So each flow generates only one flow initiation request. $Ti_{x,y}$ is the interval time between two flow initiation requests with source host x and destination y. We assume that $Tr_{x,y}$ has the same distribution as $Tf_{x,y}$. Let $f_{x,y}$ represent the flows from host x to y. Therefore, the arrival of $f_{x,y}$ initiation requests at the control plane is subject to Poisson distribution.

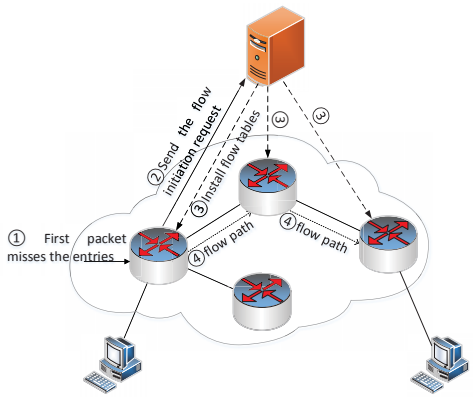


Fig. 1: The flow Initiation in SDN

3) Controller model: The load statistics on controller which include CPU, memory utilization, and network I/O rate are done by Dixit [9]. It concludes that the CPU is the throughput bottleneck. Thus, we focus on the time spent by CPU in processing the flow initiation requests. The processing time depends on the network topology, routing algorithms and controller computing power. The time complexity of routing algorithms is denoted by $g(V, E)$, where V represents the number of network nodes and E represents the number of network connections. The computing power is represented by K. Then we assume the processing time is subject to the exponential distribution with average value of $\frac{g(V,E)}{K}$. Dijkstra's algorithm is widely adopted in routing protocol such as OSPF and IS-IS [10]. Thus, Dijkstra's algorithm is assumed to be used by the controller to look for the best path from the source to destination. The complexity of Dijkstra' algorithm depends on its implementation. Generally, it has a complexity of $O(V^2)$ in the general graph, and the complexity can be reduced to $o(V^2/\ln(V))$ in the sparse graph or further reduced using advanced data structures. In this work, for simplicity we use $g(V, E) = V^2$ for illustrating our computation and comparison. Since the arrival of initiation

requests of $f_{x,y}$ is subject to Poisson distribution, and that the sum of independent Poisson distributed random variables is also a Poisson distribution, the aggregated arrival of flow initiation requests at the controller has a Poisson distribution. Thus, each controller can be modeled as a $M/M/1$ queue.

4) Controller plane structures: There are three categories of SDN control plane structures: centralized, decentralized and hierarchical structures. The centralized structure has only one controller. The controller has a global network view and processes all flow initiation requests. The decentralized structure and hierarchical structure have multiple controllers. In decentralized structure, the controllers are organized in a flat style and the controllers have a peer relationship. We study two strategies in the decentralized structure. In the first strategy, every controller has a global network topology. In the other one, every controller has only the topology of its local network and each of its neighboring local networks are abstracted as a logical node. For the hierarchical controller structure, controllers and switches are organized in a tree structure, and controllers are divided into different layers. Switches are leaf nodes in the tree. There are two different controllers in this architecture: *leaf controllers* and *internal controllers*. A leaf controller does not have other controller in its subtree and it will manage all the switches in its subtree. An internal controller will manage all its children controllers only. In this paper, we focus on the hierarchical structures with two layers.

In the local view decentralized structure and the hierarchical structure, each flow initiation request may be processed by multiple controllers. Thus, we could use the *average controller distance* to compute how many controllers needed to compute a flow initiation request on average. The average controller distance controller is a structure parameter. In the decentralized structure with the local view, each controller and its local network can be abstracted as a logical node. If two local physical networks are connected directly, a edge is added into corresponding logical nodes. Thus, we can get a logical graph. Then, the average controller distance is defined as the average distance between two nodes in the logical graph. The average leaf controller distance in the hierarchical structure can be defined in the same way.

III. Performance analysis of flow initiation in different structures

A. Centralized structure

In the centralized structure, the average arrival rate of flow requests is $\lambda_C = N \times (N-1) \times \lambda$. It has a Poisson distribution. The processing time of controller has an exponential distribution. And the average processing rate $\mu_C(N)$ is inversely proportional to the scale of the network N. Then, $\mu_C = \frac{K}{g(N)}$. Thus, the average response time of flow initiation is

$$E\{T_C(N)\} = \frac{1}{\mu_C - \lambda_C}$$

B. Decentralized structure

The decentralized structures have multiple controllers. The number of controllers is assumed to be m_D. Each controller manages a local network. And there are two strategies for decentralized structures.

In the **first** strategy (also called *local view strategy*), each controller has its local network view and each of its neighboring local networks is abstracted as a logical node. The flows can be divided into two types: global flows and local flows. If the flow's source host and destination host are managed by the same controller, it is the local flow. Otherwise, it is the global flow. The initiation request of a local flow is processed by only one controller, and the initiation request of a global flows is processed by multiple controllers. When a controller receive a global flow initiation request, the global flow initiation request will be split into two initiation requests. One is the local flow initiation request for the controller, and the other is a local or global flow initiation request for one of its neighboring controllers. The global flow initiation request is processed as above until there are no global flow initiation requests. If the average controller distance is d_H, each global flow initiation request will be split into $d_H + 1$ local requests on average. And the response time for the global flow initiation request is the sum of the response time for these local requests.

Among the $N \times (N-1)$ flows, there are $\frac{N^2}{m_D} - N$ local flows and $N^2 - \frac{N^2}{m_D}$ global flows.Then, we can conclude that the arrival rate of flow initiation request at each controller is:

$$\lambda_{D,l} = \lambda \times \frac{(N^2 - \frac{N^2}{m_D}) \times (d_D + 1) + (\frac{N^2}{m_D} - N)}{m_D}$$

Each controller needs to mange a topology with less than $\frac{N}{m_D} + m_D - 1$ nodes. Therefore, The average service rate of the controller is assumed to be:

$$\mu_{D,l} = \frac{K}{g(\frac{N}{m_D} + m_D - 1)}$$

According to the Little's law, the queue length for each controller is:

$$L_{D,l} = \frac{\lambda_{D,l}}{\mu_{D,l} - \lambda_{D,l}}$$

Based on the theory of Jackson network,the average response time of each request is:

$$E\{T_{D,l}\} = \frac{L_{D,l} \times m_D}{N \times (N-1) \times \lambda} = \frac{1 + \frac{N \times (m_D - 1)}{(N-1) \times m_D} \times d_D}{\mu_{D,l} - \lambda_{D,l}}$$

In the **second** strategy (also called *global view strategy*), each controller has a global view of the network. Thus, each controller can process all the flow initiation requests generated by its local networks. The average service rate of the controller is $\mu_{D,g} = \frac{K}{g(N)}$. And the average arrival rate of initiation request at each controller is $\lambda_{D,g} = \frac{\lambda \times N \times (N-1)}{m_D}$. Hence, the average response time of flow initiation requests is

$$E\{T_{D,g}(N)\} = \frac{1}{\mu_{D,g} - \lambda_{D,g}}$$

C. Hierachical structure

Hierarchical structures usually have two layers [2] [5] . We focus on the hierarchical structures with two layers in this paper and hierarchial structures with more than two layers can be analyzed in a similar way.

In the structure, controllers are organized in a tree structure, and controllers are divided into two different types: one root controller and multiple leaf controllers. The leaf controllers manage the data plane directly. And each leaf controller only has the view of its local network and is managed by the root controller. A leaf controller and its local network are abstracted as a logical node by the root controller. The root controller has the topology of these logical nodes. The flow can be categorized into two types: local flows and global flows. The source host and destination host of the local flow are managed by the same leaf controller. And global flows are just the opposite. Each local flow initiation request will be processed by only one leaf controller. And each global flow initiation request will be processed by the root controller first, and then will be split into multiple local requests which will be processed by corresponding leaf controllers.

We assume there are m_H leaf controllers and the average leaf controller distance is d_H. Therefore the root controller mange a logical graph with m_H nodes. So the average service rate of the root controller is $\mu_{H,r} = \frac{K}{g(m_H)}$. Each leaf controller manage $\frac{N}{m_H}$ hosts. Then, the average service rate of each leaf controller is $\mu_{H,l} = \frac{K}{g(\frac{N}{m_H})}$.

Let $c_{i,j}$ denote the jth controller at ith layer (the root controller in the first layer and leaf controllers in second layer). $C_{x,y}$ denotes the set of controllers which are required to process the initiation requests of $f_{x,y}$. Let $I_{x,y}(i,j)$ be a binary variable indicating whether $c_{i,j} \in C_{x,y}$. The value of $I_{x,y}(i,j)$ is defined as follows:

$$I_{x,y}(i,j) = \begin{cases} 1; \text{if } c_{i,j} \in C_{x,y} \\ 0; \text{otherwise} \end{cases}$$

Among the $N^2 - N$ flows, there are $\frac{N^2}{m_H} - N$ local flows and $N^2 - \frac{N^2}{m_H}$ global flows. Thus, we can get

$$\sum_{x=1}^{N} \sum_{y=1, y\neq x}^{N} I_{x,y}(1,1) = N^2 - \frac{N^2}{m_H}$$

$$\sum_{x=1}^{N} \sum_{y=1, y\neq x}^{N} \sum_{j=1}^{m_H} I_{x,y}(2,j) = (N^2 - \frac{N^2}{m_H}) \times (d_H + 1) + \frac{N^2}{m_H} - N$$

Therefore, the average arrival rate of flow initiation requests at root controller is

$$\lambda_{H,r} = \lambda \times \sum_{x=1}^{N} \sum_{y=1, y\neq x}^{N} I_{x,y}(1,1)$$

And the average arrival rate of requests at each leaf controller is:

$$\lambda_{H,l} = \lambda \times \sum_{x=1}^{N} \sum_{y=1, y\neq x}^{N} \sum_{j=1}^{m_H} \frac{I_{x,y}(2,j)}{m_H}$$

Let T_H denote the response time for flow initiation request. We can split T_H into two parts: the response time at the root controller denoted by Tr and the response time at the leaf controllers denoted by Tl. And it is obtained $E\{T_H\} = E\{Tr\} + E\{Tl\}$. Let $Tr_{x,y}$ denote the response time for initiation request of $f_{x,y}$ at the root controller (if $f_{x,y}$ is a local flow, $Tr_{x,y} = 0$). And let $T_{c_{i,j}}$ denote the response time for initiation request at $c_{i,j}$. Then, we can get:

$$\begin{aligned} E\{Tr\} &= \frac{\sum_{x=1}^{N} \sum_{y=1, y\neq x}^{N} E\{Tr_{x,y}\}}{N \times (N-1)} \\ &= \frac{\sum_{x=1}^{N} \sum_{y=1, y\neq x}^{N} E\{T_{c_{1,1}}\} \times I_{x,y}(1,1)}{N \times (N-1)} \\ &= \frac{N - \frac{N}{m_H}}{N-1} \times \frac{1}{\mu_{H,r} - \lambda_{H,r}} \end{aligned}$$

If $f_{x,y}$ is a global flow, the initiation request generated by $f_{x,y}$ will be split into $\sum_{j=1}^{m_H} I_{x,y}(2,j)$ local requests by the root controller. And these requests will be sent to the corresponding leaf controller in $C_{x,y}$ simultaneously. Let $Tl_{x,y}$ denote the response time for $f_{x,y}$ initiation request at leaf controllers. Therefore, $Tl_{x,y}$ is equal to the longest response time for these local requests. Thus, we have

$$Tl_{x,y} = Max_{j=1}^{m_H} I_{x,y}(2,j) \times T_{c_{2,j}}$$

If $f_{x,y}$ is a local flow, the expression of $Tl_{x,y}$ is the same. As $T_{c_{2,j}} (j = 1, 2, \cdots, m_H)$ are independent identically distributed, and have a negative exponential distribution with the average value of $\frac{1}{\mu_{H,l} - \lambda_{H,l}}$, we have

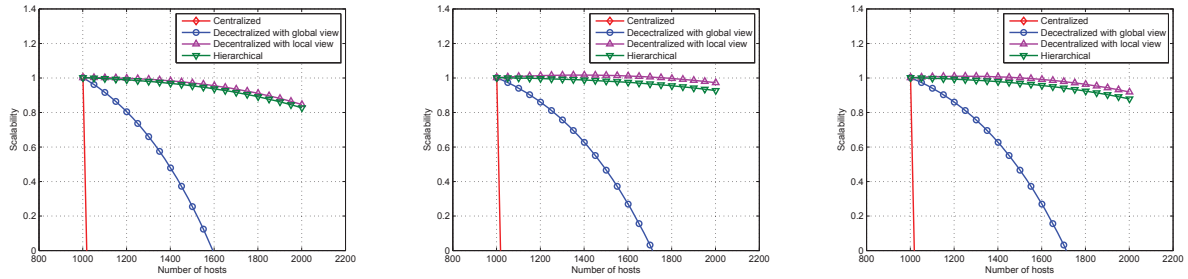
$$\begin{aligned} P\{Tl_{x,y} < t\} &= \Pi_{j=1}^{m_H} P\{I_{x,y}(2,j) \times T_{c_{2,j}} < t\} \\ &= (1 - e^{(\lambda_{H,l} - \mu_{H,l}) \times t})^{\sum_{j=1}^{m_H} I_{x,y}(2,j)} \end{aligned}$$

Let $d_{x,y} = \sum_{j=1}^{m_H} I_{x,y}(2,j)$. The probability density function of $Tl_{x,y}$ is

$$\begin{aligned} f_{Tl_{x,y}}(t) &= d_{x,y} \times (1 - e^{(\lambda_{H,l} - \mu_{H,l}) \times t})^{d_{x,y} - 1} \\ &\quad \times (\lambda_{H,l} - \mu_{H,l}) \times e^{(\lambda_{H,l} - \mu_{H,l}) \times t} \end{aligned}$$

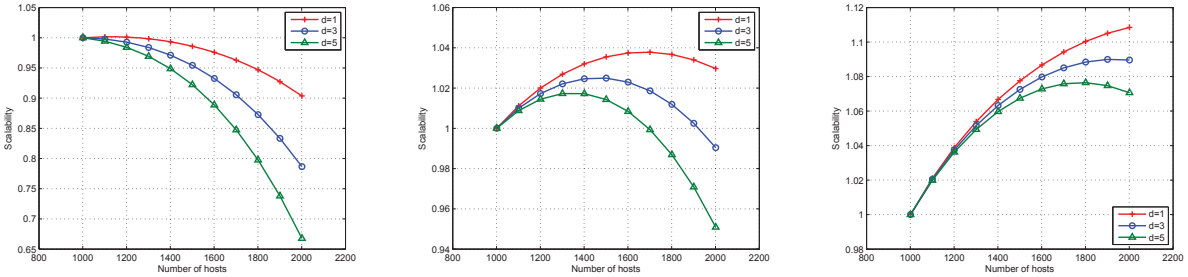
Then, the average value of $Tl_{x,y}$ is

$$\begin{aligned} E\{Tl_{x,y}\} &= \int_0^{\infty} f_{Tl_{x,y}}(t) \times t \ \mathrm{d}t \\ &= \frac{d_{x,y}}{\lambda_{H,l} - \mu_{H,l}} \times \sum_{i=0}^{d_{x,y}-1} \binom{d_{x,y} - 1}{i} \times \frac{(-1)^i}{d_{x,y}^2} \\ &\leq \frac{\ln d_{x,y} + 1}{\lambda_{H,l} - \mu_{H,l}} \end{aligned}$$



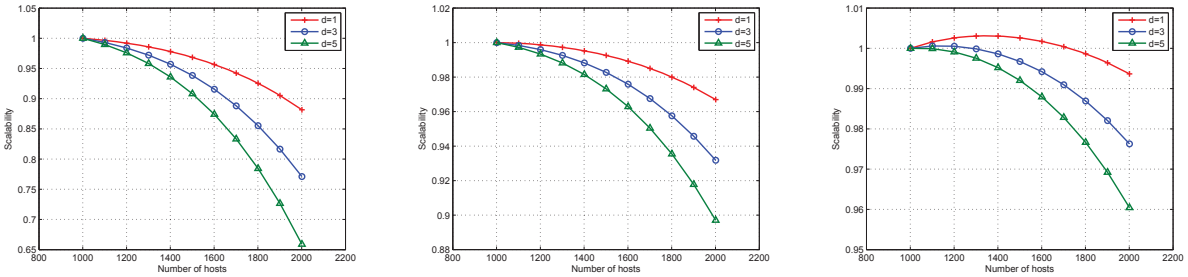
(a) $m = 6$ $d = 2$ (b) $m = 8$ $d = 2$ (c) $m = 8$ $d = 4$

Fig. 2: Scalability comparison among different structures



(a) $m = 6$ (b) $m = 9$ (c) $m = 12$

Fig. 3: Scalability of the local view decentralized structure with different values of d



(a) $m = 6$ (b) $m = 9$ (c) $m = 12$

Fig. 4: Scalability of the hierarchical structure with different values of d

Therefore, it is obtained

$$\begin{aligned} E\{Tl\} &= \frac{\sum_{x=1}^{N}\sum_{y=1,y\neq x}^{N} E\{Tl_{x,y}\}}{N\times(N-1)} \\ &< \frac{\sum_{x=1}^{N}\sum_{y=1,y\neq x}^{N}(\ln d_{x,y}+1)}{N\times(N-1)\times(\mu_{H,l}-\lambda_{H,l})} \\ &< \frac{\ln(\frac{\sum_{x=1}^{N}\sum_{y=1,y\neq x}^{N} d_{x,y}}{N\times(N-1)})+1}{\mu_{H,l}-\lambda_{H,l}} \\ &= \frac{\ln((\frac{N-\frac{N}{m_H}}{N-1})\times d_H+1)+1}{\mu_{H,l}-\lambda_{H,l}} \end{aligned}$$

Thus, the average response time in hierarchical structure is

$$E\{T_H\} = \frac{\frac{N-\frac{N}{m_H}}{N-1}}{\mu_{H,r}-\lambda_{H,r}} + \frac{\ln(\frac{N-\frac{N}{m_H}}{N-1}\times d_H+1)+1}{\mu_{H,l}-\lambda_{H,l}}$$

IV. Discussions on Scalability of Different Structures

In this section, we compare the scalability of four types of control plane. Then, in the local view decentralized structure and hierarchical structure, the relationship between the scalability and its average controller distance is discussed.

For simplicity, $\Psi_C(N_1,N_2)$, $\Psi_{D,g}(N_1,N_2)$, $\Psi_{D,l}(N_1,N_2)$, $\Psi_H(N_1,N_2)$ are used to denote the scalability of centralized structure, global view decentralized structure, local view decentralized structure and hierarchical structure when the host number varies from N_1 to N_2. For all structures, we get $\varphi(N) = N\times(N-1)\times\lambda$. $C(N)$ can be replaced by the number of controllers when the price of each controller is the same. Therefore, the scalability can be computed by Eq. (1). In SDN, the number of hosts is considerably larger than the number of controllers. For simplicity and without loss of generality,

we assume $N_2 > N_1 >> \max\{m_D, m_H\}$. Thus, we have Theorem 1.

Theorem 1. *If $N_2 > N_1 >> \max\{m_D, m_H\}$, the scalability of four types of SDN control planes are as follows.*

$$\Psi_C(N_1, N_2) \approx \frac{K - N_1^4\lambda}{K - N_2^4\lambda}$$

$$\Psi_{D,g}(N_1, N_2) \approx \frac{Km_D - N_1^4\lambda}{Km_D - N_2^4\lambda}$$

$$\Psi_{D,l}(N_1, N_2) \approx \frac{Km_D^4 - N_1^4(d_D m_D - d_D + m_D)\lambda}{Km_D^4 - N_2^4(d_D m_D - d_D + m_D)\lambda}$$

$$\Psi_H(N_1, N_2) \approx \frac{Km_H^4 - N_1^4(d_H m_H - d_H + m_H)\lambda}{Km_H^4 - N_2^4(d_H m_H - d_H + m_H)\lambda}$$

According to Theorem 1, the comparison of scalability among the four types of control plane can be easily obtained. The conclusion is shown in Theorem 2.

Theorem 2. *If $N_2 > N_1 >> m_D = m_H$ and $d_D = d_H$, then $\Psi_H(N_1, N_2) \approx \Psi_{D,l}(N_1, N_2) > \Psi_{D,g}(N_1, N_2) > \Psi_C(N_1, N_2)$.*

Let $\Psi_{D,l}^d(N_1, N_2)$, $\Psi_H^d(N_1, N_2)$ denote the scalability of local view decentralized structure and hierarchical structure with the average controller distance of d when the host number varies from N_1 to N_2. We can prove both $\Psi_{D,l}^d(N_1, N_2)$ and $\Psi_H^d(N_1, N_2)$ become worse when d increases. So we have Theorem 3.

Theorem 3. *If $d_1 < d_2$, then $\Psi_{D,l}^{d_1}(N_1, N_2) > \Psi_{D,l}^{d_2}(N_1, N_2)$ and $\Psi_H^{d_1}(N_1, N_2) > \Psi_H^{d_2}(N_1, N_2)$.*

V. NUMERICAL EVALUATION

In this section, we give numerical results which will provide some intuitive conclusions and verify the propositions above.

In the numerical experiments, according to the network traffic statistics in data center [11], the average number of flow request λ from one host to another is assumed to be 0.001 per second. K is set to be 2^{30} (We have experimentally obtained that the PC which has an Intel(R) Core(TM) i3-2120 processor and 4.00 GB RAM can perform about 2^{30} additions per second). N_1 is set to be 1000, and N_2 ranges from 1000 to 2000. Let $m_D = m_H = m$ and $d_D = d_H = d$.

The scalability comparison among four types is shown in Figure 2. In order to verify Theorem 1, we set different m and d. We can see that, the scalability of the hierarchical and the decentralized structure with local view strategy nearly the same. Both of them are better than the other two types. And the centralized structure has the worst scalability. All of these conclusions from Figure 2 are in consistent with Theorem 1.

In Figure 3 and Figure 4, we can see,in both hierarchical structure and decentralized structure with local view, the scalability becomes worse as d increases.

VI. CONCLUSION

In this paper, we define what is the scalability of SDN control plane, and propose a mathematical metric to quantify the scalability. We analyze the scalability of SDN control plane in networks in which hosts have the similar traffic pattern. Three typical structures of control plane, namely centralized, decentralized(peer-to-peer) and hierarchical structures are studied and analyzed with detailed calculation, and some useful rules and conclusions have been proposed to quantify the scalability comparison of these structures. Furthermore, numerical evaluation was done to verify our propositions. We believe that this work shed some lights into the design and construction of SDN control plane with higher scalability.

In the future, we will build a more practical model for the SDN control plane and will study the effects of the traffic burstiness, the signaling overhead to obtain the network view and the cost to keep distributed controllers synchronized on the scalability of SDN control plane.

ACKNOWLEDGMENT

The research of authors is partially supported by National Basic Research Program of China (973 Program) under Grant No. 2010CB328105, Tsinghua EMC Visiting Chair Professorship, NSF CNS-1035894, NSF ECCS-1247944, NSF ECCS-1343306, National Natural Science Foundation of China under Grant No. 61170216, No. 61228202.

REFERENCES

[1] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 136–141, 2013.

[2] S. Schmid and J. Suomela, "Exploiting locality in distributed sdn control," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 121–126.

[3] A. Tootoonchian and Y. Ganjali, "Hyperflow: a distributed control plane for openflow," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*. USENIX Association, 2010, pp. 3–3.

[4] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama *et al.*, "Onix: A distributed control platform for large-scale production networks." in *OSDI*, vol. 10, 2010, pp. 1–6.

[5] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 19–24.

[6] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with difane," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 351–362, 2010.

[7] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: scaling flow management for high-performance networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 254–265.

[8] P. Jogalekar and M. Woodside, "Evaluating the scalability of distributed systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 11, no. 6, pp. 589–603, 2000.

[9] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed sdn controller," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 7–12.

[10] A. S. Tanenbaum, "Computer networks 4th edition," 2003.

[11] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 267–280.