# A Management Model for SDN-based Data Center Networks

Yifei Xu
China Telecom
Beijing, China
yifei.xu@net.pku.edu.cn

Yue Yan
Peking University
Beijing, China
yanyue@pku.edu.cn

Zhuyun Dai
Peking University
Beijing, China
daizhuyun@pku.edu.cn

Xiaolin Wang
Peking University
Beijing, China
wxl@pku.edu.cn

***Abstract*—Current data centers often employ SDNs (Software-Defined Networks) that are layered in most cases. However, information overwhelming has been a limitation to SDN deployment. We present a management model for data center networks. In this approach, regional networks on lower layers will be aggregated and viewed as single switches to upper layers. Management information will be divided into three parts, which can be seen by network managers, regional controllers and tenants, respectively.**

***Index Terms*—SDN, data center, cloud network.**

## I. Introduction

SDN (Software-Defined Networks) were designed for the purpose that smart open-source programmable controllers control dumb low-cost switches. However, controllability and flexibility of SDN are restricted by the lack of information organization and division. As a result, controller logics are often far more complex than they are supposed to be.

In this poster, we discuss a new way of organizing topologies and information of OpenFlow [1] networks in data centers, and present a management model that divides network views and information orthogonally and orderly to reduce management complexity. In our approach, regional networks on lower layers will be aggregated and viewed as single switches to upper layers. Information of management will be divided into three parts, which are, respectively, managed by network managers, regional controllers and tenants. To achieve this, we established a mechanism determining which parts a massage should be sent to. In data center networks, aggregation of networks means that a regional network will decide its own inside logic and be seen as a single switch whose ports are edge ports of the regional network to upper layers.

Our contributions are two-fold. First, we present an isolation mechanism enabling regional customization and overcoming flatness of OpenFlow networks. Second, by dividing information into several sorts, complication of management is significantly reduced, while controllability is improved. As a result, a reasonable and decent management model of collaboration will replace the previous one which stacks all tasks to one controller.

## II. Network Aggregation

Regional network aggregation is easy to achieve in layered data center networks. Consider the general network topologies in data centers [2]. Most approaches divide networks along with management system structures into many layers including but not limited to hosts, racks, clusters, zones, data centers and clouds. In most usual cases, a switch on upper layers will be connected to every switch on next layer in its region.

Originally, OpenFlow networks are flat and all elements in a network are connected to a central controller. If there are several controllers in a network, all controllers must share a whole network view [3]. This could be a waste of storage and computation resources since controllers cannot handle messages from elements unconnected to them. For the sake of performance and controllability, we assume full-intra-connection of regional networks, which means switches are connected with each other if they are connected to the same controller on an upper layer. What is more, for controllers of upper networks, they have to be connected to controllers of lower regional networks. Controllers will perform all functions of OpenFlow Switches. To avoid redundant network view information, connections between regional switches will not be seen by upper observer. Upper controllers will view regional networks as single switches and deploy rules to their controllers. Through this approach, upper or global controllers can focus on inter-regional-network communication, and spare computation and storage to lower or regional controllers. Figure 1 shows an example of inherited and aggregated networks.

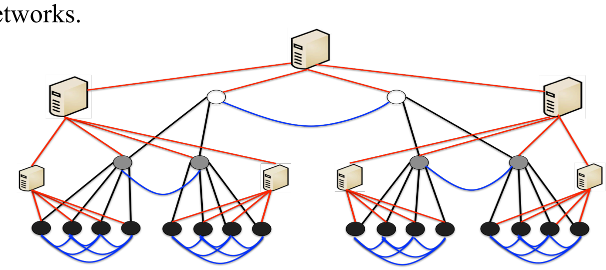

Fig. 1. Topology of aggregated networks. Controllers are connected to all regional switches and will be seen as switches to upper controllers.

## III. Information Division

OpenFlow has a message mechanism for information exchanging and event alerting between controllers and switches. There are basically three kinds of messages:

1) Status Exchange. Switches and controllers report their status to whomever it concerns. This kind of messages will be sent when (1) status changes (connection up, connection down and configuration changed, etc.) or (2) periodical status report (such as echo) happens.

2) Rules Query and Configuration. This typically happens when a packet missed all entries on flow table. As a result, it is sent to controller and controller will probably deploy a flow table entry on the switch correspondingly.

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

3) Statistics Query and Report. This happens when a controller needs to query statistics on connected switches. Switches only send statistics reports after receiving a query.

In our approach, since inside information should be hidden, we have to decide whether a certain message should go to regional controller or upper controller. Our strategy is simple:

1) For a status exchange message, if the message reports a status change that will cause an edge view change, regional controller will report a status change to upper controller.

2) For a rule query message, if the packet concerning this message aims at or sources from switches outside of this regional network, upper controller must be notified to decide first and/or last switches on the edge of the regional network. Otherwise regional controller can make decision by itself.

3) For a statistics message, all kinds of statistics are also calculated and maintained on regional controllers (also as aggregated switches) according to switch-reported data, and will be reported to upper switches when queried.

In this division we assume that a regional network is assigned to a tenant (this is unreal but worth analogizing). The result of information division is presented in Table 1. Note that layers of controllers are relatively inherited. That means upper controllers and regional controllers technically have same abilities. They just gain information from different views.

TABLE I. INFORMATION DIVISION BY ROLES

| Role | Messages describe |
|---|---|
| Network Managers (Upper Controllers) | ***Inter regional network rule decisions. Statistics of lower regional networks. Status changes effecting global network view.*** |
| Regional controllers | ***Intra rule decisions. Statistics of from lower controllers or switches. Status changes effecting regional network view.*** |
| Tenants | ***Statistics of regional networks.*** |

## IV. IMPLEMENT AND EVALUATION

Our special-purposed controller Hanoi is implemented based on POX [4]. We added an aggregation module that made the controller behave like a switch to upper controller. Information division and translation is enforced by our system. Figure 2 shows the simple workflow of our system.

There is a very fundamental trade-off between time taken to compute paths and the probability that some of the paths are relatively long. As a result, our evaluation focuses on comparing performances of one controller mode and our mode.

In our evaluation, we simulated a network with tree-like topology in Mininet and measured the overhead caused by network view division. We used Iperf as a benchmark. Single controller mode and flat multiple controller mode were used as comparison. The bandwidth performance of Iperf test is presented in Figure 3. As the result shows, the overhead of our system is very small. When the network is large enough, Hanoi controllers benefit from the broad network view that weighs out the overhead cost by database operation and inter-controller communication. Its performance is even better than flat multi-controller network.

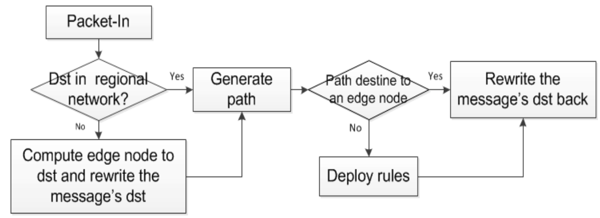


Fig. 2. Workflow of information intercepting and rewriting on Hanoi.

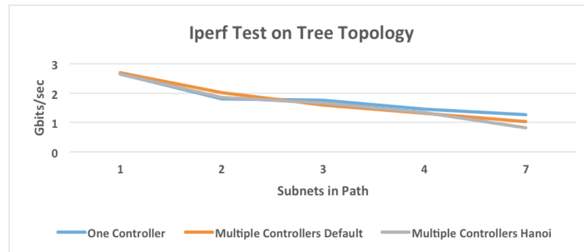


Fig. 3. Iperf test of bandwidth.

## V. CONCLUSION AND FUTURE WORK

In this poster, we proposed a management model of layered OpenFlow networks. We believe that through our network aggregation and information division mechanism, complexity of management can be reduced significantly. Evaluation also showed that overhead of our system is endurable.

In the future we plan to test our system in a real virtualized data center. Also, we are developing a suit of API according to our roles design to make this system usable for tenants and network managers. Current data center network management tools mainly lay in cloud management systems such as OpenStack and CloudStack. In such cloud systems, network management models are defined as plugins or resources. It is easy to adapt our model into these systems by developing a plugin, which is exactly what we plan to do.

## VI. ACKNOWLEDGEMENT

This work is supported by Tencent Inc., the National Natural Science Foundation of China (Grant No. 61232008, 61170055, 61272158, 61328201), the National High Technology Research 863 Program of China (2012AA010905), the Research Fund for the Doctoral Program of Higher Education of China (20110001110101), and the Special Foundation of Industry Development for Biology, Internet, New Energy and New Material of Shenzhen (JC201104210107A).

## REFERENCES

[1] N. McKeown et al., "Openflow: enabling innovation in campus networks," in ACM SIGCOMM, 2008.
[2] A. Greenberg et al., “VL2: A Scalable and Flexible Data Center Network,” in ACM SIGCOMM, 2009.
[3] N. Gude et al., “NOX: Towards an Operating System for Networks,” in ACM Computer Communication Review. pp. 105-110. July 2008.
[4] T. Koponen et al., “Onix: a distributed control platform for large-scale production networks,” in OSDI, 2010.