International Journal of Future Generation Communication and Networking
Vol.10, No.5 (2017), pp.59-66
http://dx.doi.org/10.14257/ijfgcn.2017.10.5.06

Challenges and Research Issues of Decision Maker for Solving Complex Problems Having Multiple Subjective and Conflicting Choices for SDN Controller

Shailendra Mishra[1] and Mayank Singh[2]

[1]Al Majmaah University, Saudi Arabia
[2]Krishna Engineering College, India

Abstract

This paper focuses on challenges and research issues of decision maker for solving complex problems having multiple subjective and conflicting choices for SDN Controller, which select multiple properties of the controller for better efficiency of network equipment eventually increasing network throughput and having less delays and cost saving. Making the network migration toward SDN in seamless and giving better results for the users. Throughput for the network can be increased by having centralized controller and no processing at network elements like switches and better traffic management thereby decreasing the latency as less as possible. SDN will helps to improve centralized visibility as all the underlying open flow switches are connected to controller, all switches can be configured from SDN controller without accessing individual switches. This research will help how we select the best possible SDN controller which in result will help reduce the complexity of a network, price of implementation and maintenance of the network, empower scalability, less hardware and software requirements, less technical resources requirements, centralized visibility, hassle free traffic engineering, high availability of network.

***Keywords**: SDN, NFV, NOX, POX*

1. Introduction

In the recent years, the hot top in network administration has been software defined networking (SDN) and SDN controllers. There is, nonetheless, remarkable confusion among endeavor IT industry with respect to these points. There are numerous sources of that confusion, including the sheer number of organizations and regulatory bodies who have solutions that take care of various issues utilizing various models and developments, every one of whom claim to offer SDN or potentially SDN controllers.

Software-defined networking (SDN) is an extension of network function virtualization (NFV) wherein software can perform dynamic reconfiguration of an operator's network topology to adjust the load and demand. In the past few years SDN attracted many industries due to lot of advantages mainly due to network virtualization.

Software Defined Networking has come up with the solution of traditional network issues which includes separate control plane for each network device in traditional devices including switches and routers, so that not only the decision is made on each device but also the processing takes place at each device. With traditional Network, there is no centralized visibility of network devices and there needs to be separate NMS Network monitoring system to be installed for full visibility. In order to resolve these issue, SDN comes up with centralized approach with separate data and control plane for the purpose of forwarding, visibility of all the network devices, programmable interfaces, scalable, speedy network, OS compatibility as open interface, source code access [1], so that the decision is made on each device and it takes processing at each device. Secondly, VLAN is scalable up to 4096 [2], as VLAN bit in Ethernet frame is 12bit which comprises of

4096 combinations. Therefore, there can be as many as 4096 networks in a network which minimize the scalability of growth in a network. Traditional network restricts the scalability of network expansion. Access to each device separately, configurations on each device separately makes harder for work to configure each device.

Today, numerous SDN technologies are truly at the point for facilitating cloud computerization arrangements. At the point when SDN initially showed up on the innovation sight, there were more inflexible thoughts of how SDN structures ought to be composed and what characterized a SDN arrangement. Today, clients take a more extensive perspective of what sort of SDN arrangement is correct for them. As the essential utilize case for SDN has developed toward cloud mechanization, clients consider what they're looking for in a policy-based mechanization arrangement rather than simply the specifics of the fundamental SDN innovation. SDN is additionally really an open innovation.

In the event that a system is consistent with the privilege SDN benchmarks, it could be controlled by numerous SDN controller applications. This is superior to every system having its own administration reassure and changes that administration even more overwhelming. Today, different SDN models are advancing in diverse ranges, and effective SDN techniques will consistently be in the market, between operable multi-dimensional communities with key open source innovations or institutionalized solutions.

Alongside the development to SDN, there are various innovation patterns that are influencing the design also, outline of advanced server plane and venture systems that must be calculated into SDN innovation prerequisites. In many associations, the server plane is moving far from conventional client server structures to models in which altogether more information is being exchanged between servers inside the server plane (regularly called east-west activity). This requires more system versatility and more refined strategies for asset distribution.

Also, numerous IT offices are demonstrating incredible enthusiasm for moving to open, private, or cross breed cloud situations. SDN, actually, is being taken a glance at as a key giver to expanding IT nimbleness and enhancing self-service IT offerings. Undertakings are additionally putting resources into huge information applications to encourage better business basic leadership.

These sorts of uses require enormous parallel handling transverse over hundreds or a huge number of servers. The request to handle enormous information sets is putting more noteworthy weight on the system and driving the requirement for more prominent limit and computerization. These components assume a huge part in the interest for more effective, dexterous, and higher performing corporate system situations. SDN is planned to meet those requests. The top level advantages of a SDN technique gather to all ranges of the association. You will get an upper hand since your framework will accomplish more.

The speed of your business will expand, the aggregate cost of proprietorship will go down also, and dangers will be diminished due to the more noteworthy security. Organizing associations are under expanding weight to be more productive and lithe than is conceivable with the conventional way to deal with systems administration. One wellspring of that weight comes about because of the across the board appropriation of server virtualization. As a component of server virtualization, virtual machines (VMs) are progressively moved between servers in a matter of seconds or minutes. Be that as it may, if the development of a VM crosses a Layer 3 limit, it can take days or weeks to reconfigure the system to bolster the VM in its new area. It can here and there be hard to characterize precisely what it implies for a system to be light-footed.

All things considered, on the off chance that it takes weeks to reconfigure the system to bolster the development of a VM, that system isn't light-footed. The primary concern is that a conventional system develops gradually; is constrained in usefulness by what is given by the sellers of the ASICs and the merchants of the system apparatuses; has a generally abnormal state of OPEX and is moderately static in nature. SDN holds the guarantee of overcoming those confinements.

Protocols running on devices as proprietary from companies which is not giving permission to change on source code[3].No open programmable interface for devices as in traditional network, there is no option for that [4]. Processing of packets at each device for decision making, such as routing or switching, so it takes CPU usage, delay in overall traffic and adding extra time for reachability of packets. With traditional Network there is no centralized visibility of network devices and there needs to be separate Network monitoring system (NMS) to be installed for full visibility.

In order to resolve these issue, SDN comes up with centralized approach with separate control plane and data plane for forwarding, visibility of all network devices, programmable interfaces, scalable, speedy network, OS compatibility as open interface, source code access [5].Control Plane and data plane are inside the controller but they are decoupled for their work. Data plane is used for carrying user traffic and control plane is used for communication between controller and network application for decision for forwarding the decisions and actions.

Many SDN Controller platforms has come into existence in the past few years like Beacon OpenFlow controller [6.], NOX [7,8],POX [9], Nettle [10], OpenDay Light [11]], FloodLight [12], Ryu [13], *etc.* By using these platform researchers develop many applications such as load balancing [14], network virtualization [15],energy efficient networking [16],dynamic access control in enterprise network [17],virtual machine mobility [18] *etc.* To use the controller in network there must be some criteria like size of network, bandwidth, delay, user requirements, GUI support, documentation, year of maturity, cost, reliability *etc.* for selecting the suitable controller which can cope up the requirements of the current setup [19].

To solve these issues *i.e.* multiple located branch networks, cost, technical resources at each location, expertise, separate control plane for configurations, decentralized visibility of network devices, separate VLANs for each branch, complex traffic engineering, limited physical access of branches in the context of working hours, bandwidth bottleneck at each branch, we surveyed literatures and websites for the existing SDN controllers like NOX [7,8], POX [9], Nettle[10], OpenDay Light[11], Flood Light[12], Ryu[13].

All these controllers are based on Open Flow protocol. NOX is used for management functions for organizations and home networks. It provides a high-level programmatic interface on which network management and control applications can be constructed. POX is inherited from NOX. It can used to route different applications like switches, hubs, load balancers, and/or firewall *etc.* Ryu is a component based SDN framework. It is the most verified framework with Open Flow protocol. Flood Light is open-source core of big network controller. It is programmed over apache server. OpenDay Light is specially design to generate robust code that covers major components of SDN architecture.

Software defined networking is one of the most discussed topic these days. This technology is being considered one of the favorable technologies for isolation of control plane and data plane and logical placement of centralized control from SDN controller. Data plane and control plane are decoupled for various purposes.

2. Literature Review

The review of literature has been carried out in the following contexts:

a. Software defined networking

b. SDN protocol: OpenFlow

c. SDN research challenges

SDN contains replacement of the control plane on the switch with a programmatic interface, migration of the control planes of the switches to a controller, establishment of programmatic interfaces into the controller. There is also a rapidly rising consensus that an SDN must emphasize not just on Layer 2 and Layer 3' s functionality, but also Layer 4 till Layer 7 functionality and that is must support both physical and virtual equipment. [20, 21].

Ryu is a component-based SDN framework that supplies an appropriate platform for SDN applications to run on the top of Ryu controller. It is an open source tool programmed in Python that offers well defined APIs and packet libraries and supports all versions of OFP. It is well verified with various OpenFlow switches and is properly compatible on Open vSwitch [22]. Network can be controlled by programming interface called APIs. Hardware independent technology is likely to be achieved by this technique. [23].

In SDN, segregation of control plane and data plane is like that router and switches, and they are only working for forwarding the packets, and all decisions are made by the controller which uses network applications inside controller that decides which actions should be taken for certain flow. Using SDN, network devices are configured centrally without have to configure individually [24]. Openflow is an ascending protocol used in SDN Networking, and used as Application programming interface (APIs).

It can run without dependency for any application/services. Overall expense for network is reduced by using SDN with OpenFlow Protocol [25]. SDN OpenFlow design enables accessibility of all network devices using APIs. These APIs can get information about all the network working services like routing, QoS etc.

In traditional network, network devices needs to be provisioned by accessing it, however all the network devices are being provisioned from centralized location [26]. There are two types of OpenFlow switches, OpenFlow only switches- it uses only OpenFlow Operations and OpenFlow Hybrid Switch – it uses both OpenFlow and normal Ethernet operations. TCP port being used by Openflow is 6633 and IANA has assigned port number 6653[27]. In infrastructure network more than one controller is required for high-availability. Communication is done between controllers for network status sharing and any forwarding decision being made on upper layer [29].

NOX doesn't provide service insertion, it also not support load balancing, policy enforcement, dynamic network traps, support only partial network monitoring as compared to SDN controllers like Ryu, FloodLight, OpenDayLight [30]. Selecting controller using single property is insignificant. Therefore, select multiple properties of the controller to find out their impact in the efficiency and effectiveness in network, selection process is called Multi-Criteria Decision Making [MCDM] [31].

Analytic Hierarchy Process (AHP) is used in helping decision maker facing complex problems having multiple subjective and conflicting choices [32]. AHP allows decision makers to measure the relative significance of selected object [33]

As compared with author research there is missing element of user input, real time data, bandwidth requirement, delay requirement which is base of future research. The analyzed papers in this review describe software defined networking on mathematical and technological basis. These detailes makes it very hard for organizations to assess if this technology have any specific business impact on existing network deploynent. However, the increase of users and higher number of user accomodation requirements leave service providers with the need to rethink the use of existing deployed networks for their businesses.

3. Modernization Using SDN-Based Network Applications

SDN has been a way for system advancement lately, in light of the fact that it takes into account more adaptability and extensibility than existing system models. New elements or administrations that are added to a SDN-based system frequently require just a reasonable number of control components to be updated. Redesigns on transmitting components are regularly unnecessary relying upon the southbound interface. Therefore, the segregation of the control and data plane permits both planes to advance all the more autonomously of each other to make same with other system structures. Managing devices and administrations can be strongly included in the type of system applications, which encourages their organization. Numerous system applications have as of now been proposed and assessed by the SDN research group. They can be assembled in the ranges of system administration and activity building, application server stack adjusting and to get to control, SDN security, arrange virtualization and between space directing.

3.1. SDN Network Management and Traffic Engineering

The most surely understood SDN in industry is Google's OpenFlow-based WAN, which is utilized as a group management. As introduced at the Open Networking Summit 2012 and as far as they can tell at SIGCOMM 2013. Google depicts their inner worldwide system that interconnects their server planes around the world. They talk about the genuine execution of their system and show how activity construction can be upheld by steering choices. Their activity designing server plans operations relying upon the accessible data transfer capacity in the system. They indicate critical impacts in asset use that can be accomplished with SDN and OpenFlow with regards to server synchronization and correspondence. This is conceivable in the server environment of Google, however is not for the mostly physical for other developments.

SDN facilitates the presentation of new approach in the system. In [34], IP multicast is executed in the control plane utilizing OpenFlow by taking care of multicast subscribe asks for inside the control programming. The control programming introduces the transfer segments in the changes as indicated by the multicast application. This is accomplished without change of the trasfer tools, on the grounds that their OpenFlow switches upheld the required sending operations. Be that as it may, the OpenFlow information plane must be updated if another convention requires different operations than those offered by the OpenFlow purpose being used. For SDN arrangements that support a programmable information plane, *e.g.*, FLARE [35] or programmable systems, the control and information plane can develop all the more freely from each other. A few specialists consider organize redesigns and the results of botched upgrades.

The making of easy to use interfaces for SDNs is talked about in [36] The authors exhibit a remote system administration interface to OpenFlow systems, called OMNI. It facilitates the administration of OpenFlow-based systems, which can be troublesome, because of the quantity of observing factors and various system setup choices. OMNI screens and arranges the dynamic creation of flow. Keeping that in mind, it gathers information properties of the system devices and gives a straightforward system wide perception of the present system state.

3.2. Load Balancing for Application Servers

Different SDN-based applications have been proposed for big commercial systems. A typical assignment is to load balancing online management to specific administration imitations. This is normally actualized by utilizing a devoted centerlied device that manage the load balancing. Receiving packets are specifically sent towards support counterfeits. An OpenFlow switch consequently disseminates the activity to various servers [37].

3.3. Security and Network Access Control

Conventional system security depends on securing the hosts and the utilization of resources, for example, firewalls, interruption discovery frameworks and system address interpreters. But, resources are frequently put at the edges of the system, because of the absence of system management control. Organize work virtualization (NFV) [38] is a profoundly related subject and is institutionalized at the European Telecommunications

Standards Institute (ETSI). Organize capacities, which are frequently given by particular equipment, can be virtualized with NFV. At that point, virtual system capacities can live as virtual machines on universally useful servers inside the system. Likewise with resources, SDN can be utilized to control flow to virtualized resourceses and control capacities. SFC is utilized to join different system administrations.

In [39] the authours propose the framework "Reverberation" that gives dynamically control approaches. Their answer depends on OpenManaged to realize dynamically control in system components. Because of the adaptability and the probability of fine-grained channels, they can accomplish receptive and dynamic system without the utilization of particular resourceses. They similarly highlight the weaknesses and constraint of their current VLAN-based engineering and show how the SDN approach defeats those confinements and issues.

Yao *et al.* [40] represent address validation in view of SDN so address spoofing can be productively denied. The approach depends on specific establishment of flow of packets that either allow or drop packets. There is chance that a packet arrives that does not coordinate any defined path, the controller is examined. At that point, the controller checks if the given sender is approved to send the packets. At the point when imitating is distinguished, direct drop actions are introduced in the changes to counteract support activities by a harmful intruder.

Like this, the host works on a similar IP address, while from the external view, it appears to from time to time change its IP address. This technique successfully keeps checking of the host from outside, and additionally inside attackers.

The identification and security against Denial of Service (DoS) and distributed DoS (DDoS) assaults have been discovered a few times on the evidence of SDNs. Braga *et al.* [41] identify DDoS attacks by measuring the recurrence of the managed of a flow. On the off chance that the flow recurrence surpasses a predetermined edge, they expect that a DoS assault is going on. In such a case, the controller educates the changes to drop the packets that have a place with the vindictive flow to moderate the impacts of the assault. Another argument proposed by Braga *et al.* depends on self-arranging maps to group activity designs. At the point when activity is named noxious, the controller responsively programs the system to drop the vindictive managed.

The control plane in SDN is more brought together, intense and open than in existing system models. Thusly, the control plane is clearly imperative and must not be bargained. Accordingly, NOX which is an expansion of the POX OpenFlow controller. It can be utilized as a part of OpenFlow-based SDN systems to enhance the security of basic control segments by giving part based approval of system applications and security requirement authorization inside the controller. This is accomplished by a contention identification motor that intervenes flow decide inclusion that can be activated by different system applications.

4. Research Issues

- SDN controller should be act like active-standby and in the case of one controller is down all the flow tables should go to backup controller, so that traffic operations are not disturbed.
- Dynamic multipath load-balancing, in case of congestion load can be shared.
- Handling huge amount of data in the network requires high CPU and memory at controller side so that all the requests are handled without being delayed.
- Dependency of how the number of flows SDN controller can handle from open flow process which setup the flow process each time new flow is started.
- Openflow hardware switches sends flow for decision making to SDN controller and in this way thousands of flow requests can come, which can make latency
- Controller Placement is very important as it can cause delays if not proper placed.

- Interoperability issue in SDN is one of the major issues. There is need to have interface for communication between SDN and non-SDN control plane like MPLS, This will increase the scalability of network being run.
- From Security point of view, there should be some ACL Access control list.

5. Conclusion

Traditional Networks have some issues which need to be resolved and formulate a solution to form a method for scalability, granularity, resiliency, programmable and simplicity for design. Software Defined Networking (SDN) comes up with centralized approach with separate control plane and data plane for forwarding, visibility of all network devices, programmable interfaces, OS compatibility as open interface, source code access.

Selecting controller using single property is insignificant. Major issues which are being faced are increasing requirements from user side, bandwidth availability, hardware (switches requirement at every place), technical resources are required at remote site for configurations, scalability issues, cost, high level processing power at each device, traffic engineering, resiliency against failures, decentralized visibility of hardware devices *etc.*

SDN will helps to improve centralized visibility as all the underlying open flow switches are connected to controller, all switches can be configured from SDN controller without accessing individual switches. SDN controller which in result will help reduce the complexity of a network, price of implementation and maintenance of the network, empower scalability, less hardware and software requirements, less technical resources requirements, centralized visibility, hassle free traffic engineering, high availability of network.

References

[1] Software Defined Networking vs. Traditional Networking, https://globalconfig.net/software-defined-networking-vs-traditional/.

[2] Galvin, P. Traditional Network Infrastructure Model and Problems [cited 2015 27-Dec-2015]; http://www.pluribusnetworks.com/blog/detail/traditional-network-infrastructure-model-and-problems-associated-with-it/

[3] Traditional vs Software Defined Networking. 2012 [cited 2015 25-Dec-2015]; http://www.rfwireless-world.com/Terminology/traditional-networking-vs-software-defined-networking.html.].

[4] Feamster, N., The Past, Present, and Future of Software Defined Networking. University of Maryland. www.ietf.org/proceedings,www.ietf.org/proceedings

[5] SDN, NFV and OpenFlow for Skeptics, 2014,https://www.youtube.com/watch?v=khgf2U64v3c

[6] D. Erickson. The Beacon OpenFlow controller. In Proc. HotSDN, Aug. 2013.],

[7] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: Towards an operating system for networks. ACM SIGCOMM Computer Communication Review, 38(3):105–110, July 2008.

[8] The NOX Controller. 2015 [cited 2015 5-Dec-2015]; https://github.com/noxrepo/nox.

[9] POX Controller. 2015 [cited 2015 5-Dec-2015]; https://github.com/noxrepo/pox.

[10] Voellmy and P. Hudak. Nettle: Functional reactive programming of OpenFlow networks. In Proc. Workshop on Practical Aspects of Declarative Languages, pages 235–249, Jan. 2011.

[11] Project, L.F.C. Lithium. [cited 2015 7-Dec-2015]; https://www.opendaylight.org/lithium.

[12] Networks, B.S. Floodlight. 2015 [cited 2015 7-Dec-2015]; http://www.projectfloodlight.org/floodlight/.

[13] Community, R.S.F. Build SDN Agilely. 2014 [cited 2015 8-Dec-2015]; http://osrg.github.io/ryu/.

[14] R. Wang, D. Butnariu, and J. Rexford. OpenFlow-based server load balancing gone wild. In Hot-ICE, Mar. 2011.

[15] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar. Can the production network be the testbed? In Proc. 9th USENIX OSDI, Vancouver, Canada, Oct. 2010.

[16] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis,P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: Saving energy in data center networks. Apr. 2010.

[17] Nayak, A. Reimers, N. Feamster, and R. Clark. Resonance: Dynamic access control in enterprise networks. In Proc. Workshop: Research on Enterprise Networking, Barcelona, Spain, Aug. 2009.

[18] Erickson, A demonstration of virtual machine mobility in an OpenFlow network, Aug. 2008. Demo at ACM SIGCOMM.

[19] Khondoker, R., Zaalouk, A.; Marx, R. ; Bayarou, K., Feature-based comparison and selection of Software Defined Networking (SDN) controllers, in Computer Applications and Information Systems (WCCAIS). 2014, IEEE: Hammamet. p. 1-7.

[20] Manar Jammal, Abdallah Shami ,Rasool Asal b, Yiming, Software defined networking: State of the art and research challenges. ELSEVIER, 2014. 72: p. 74–98

[21] Ashton, M., and Associates Ten Things to Look for in an SDN Controller. 2013.

[22] S. Fang, Y.Y., C.H. Foh, K.M.M. Aung, A loss-free multipathing solution for data center network using software-defined networking approach, in APMRC, 2012. 2013, IEEE. p. 1-8

[23] Myung-Ki Shin, K.-H.N., Hyoung-Jun Kim Software-Defined Network (SDN):A Reference Architecture and Open APIs in ICT Convergence (ICTC). 2012, IEEE: Jeju Island. p. 360 - 361.

[24] Hyojoon Kim; Feamster, N., Improving Network Management with Software Defined Networking, in Communications Magazine, IEEE. 2013, IEEE. p. 114 - 119.

[25] Christian Esteve Rothenberg, M.R.N., Marcos Rogerio Salvador, Carlos Nilton Araujo Corrêa, Sidney Cunha de Lucena, Robert Raszuk. Revisiting routing control platforms with the eyes and muscles of software-defined networking. in First workshop on Hot topics in software defined networks. 2012. ACM.

[26] Bakshi, K., Considerations for Software Defined Networking (SDN): Approaches and use cases, in Aerospace Conference. 2013, IEEE: Big Sky, MT. p. 1-9.

[27] OpenFlow Switch Specification. 2012, Open Networking Foundation ONF. p. 106.

[28] Wenfeng Xia, Y.W., Chuan Heng Foh, Niyato, D., Haiyong Xie, A Survey on Software-Defined Networking, in Communications Surveys & Tutorials. 2014, IEEE. p. 27-51

[29] Haiyong Xie, T.T., Diego Lopez, Ron Sidi, Hongtao Yin, and Pedro, Software-defined Networking Efforts at IETF 84. IETF, 2012

[30] Sridhar Rao, SDN Series Part Eight: Comparison Of Open Source SDN Controllers, 2015 https://thenewstack.io/sdn-series-part-eight-comparison-of-open-source-sdn-controllers/]

[31] Matthias Ehrgott, Multiple criteria optimization: State of art annotated bibliographic surveys , Editor. 2003, Kluwer acedmic. Publisher: USA

[32] Labib, A.I.a.A., Analytic Hierarchy Process and Expert Choice: Benefits and limitations. OR Insight, 2009. 22: p. 201-220.

[33] John David, K.a.D.S., Decision Lens Inc., Use Analytic Hierarchy Process For Project Selection. 2007, WWW.ASQ.ORG.

[34] Kotani, D.; Suzuki, K.; Shimonishi, H. A Design and Implementation of OpenFlow Controller handling IP Multicast with Fast Tree Switching. In Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), Izmir, Turkey, 16–20 July 2012; pp. 60–.

[35] Nakao, A. FLARE: Open Deeply Programmable Network Node Architecture. Available online: http://netseminar.stanford.edu/seminars/10_18_12.pdf 2012. (Accessed on October 30, 2016

[36] Mattos, D.; Fernandes, N.; da Costa, V.; Cardoso, L.; Campista, M.; Costa, L.; Duarte, O. OMNI: OpenFlow MaNagement Infrastructure. In Proceedings of the International Conference on the Network of the Future (NOF), Paris, France, 28–30 November 2011; pp. 52–56.]

[37] Wang, R.; Butnariu, D.; Rexford, J. OpenFlow-Based Server Load Balancing Gone Wild. In Proceedings of the USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), Boston, MA, USA, 29 March 2011; pp. 12–12.

[38] ETSI—Network Functions Industry Specification Group. Network Functions Virtualisation (NFV). 2013. Available online: http://portal.etsi.org/NFV/NFV_White_Paper2.pdf (accessed on 30 October 2016

[39] YuHunag, C.; MinChi, T.; YaoTing, C.; YuChieh, C.; YanRen, C. A Novel Design for Future on-Demand Service and Security. In Proceedings of the International Conference on Communication Technology (ICCT), Nanjing, China, 11–14 November 2010; pp. 385–388

[40] Yao, G.; Bi, J.; Xiao, P. Source Address Validation Solution with OpenFlow/NOX Architecture. In Proceedings of the IEEE International Conference on Network Protocols (ICNP), Vancouver, BC, Canada, 17–20 October 2011; pp. 7–12.

[41] Braga, R.; Mota, E.; Passito, A. Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow. In Proceedings of the IEEE Conference on Local Computer Networks (LCN), Denver, CO, USA, 11–14 October 2010; pp. 408–415.