Title	Towards middleware security framework for next generation data centers connectivity
Authors(s)	Talpur, Samar Raza; Abdalla, Sameh; Kechadi, Tahar
Publication date	2015-07-30
Conference details	2015 Science and Information Conference (SAI), London, United Kingdom, 28 - 30 July 2015
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7417
Publisher's statement	© © 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/SAI.2015.7237308

Towards Middleware Security Framework for Next Generation Data Centers Connectivity

Samar Raza Talpur
School of Computer Science
and informatics
University College, Dublin
Dublin-4, Ireland
Email: samar.talpur@ucdconnect.ie

Sameh Abdalla
School of Computer Science
and informatics
University College, Dublin
Dublin-4, Ireland
Email: sameh@ucd.ie

Tahar Kechadi
School of Computer Science
and informatics
University College, Dublin
Dublin-4, Ireland
Email: tahar.kechadi@ucd.ie

***Abstract*—Data Center as a Service (DCaaS) facilitates to clients as alternate outsourced physical data center, the expectations of business community to fully automate these data centers to run smoothly. Geographically distributed data centers and its connectivity has major role in next generation data centers. In order to deploy the reliable connections between distributed data centers the SDN based security and logical firewalls are attractive and enviable. We present the middleware security framework for software defined data centers inter-connectivity, the proposed security framework will be based on some learning processes, which will reduce the complexity and manage very large number of secure connections in real-world data centers. In this paper we will focus on two main objectives; (1) proposing simple and yet scalable techniques for security and analysis, (2) Implementing and evaluating these techniques on real-world data centers.**

***Keywords*—DCI (Data Center Inter-connectivity), DCaaS, SDDC, SDN, Virtual Networking, Distributed Firewall, OpenFlow.**

I. Introduction

In continuous with the growth of data centers. The big number of data center service providers have thousands of servers and other equipment installed, similarly expanding by twice all these equipment every 18 months and proving the prediction of Moores law [25]. Once we look at running cost, millions of dollars are spending on diversified hardware, complex workload and thousands of various applications. Despite of all the conventional data centers neither providing proper access for public trace nor real time monitoring system for researchers.

According to the research study "Growth in data center electricity use 2005 to 2010" around 1.3% of whole world's electricity were consumed in data centers in 2010, while 2% of total electricity consumed by data centers in USA. [23]. The financial firm report (USA 2011), the annual cost of 1.80 billion dollars were spent on data centers, the power consumption in USA in 2006 for data centers were 61 billion kilowatt per hour at the cost of 4.5 billion dollars. It is increasing by yearly from 4 to 8% and expected 100 billions in upcoming years [36].

DCell and BCube [15], [14] are the prominent Server-Centric Network architectural model for data centers [36], [29]. The server centric routing structure has mesh of servers which acts like intermediate node (mini switch) with neighbor node. In the routing structure of server centric architecture, mesh of servers are act like intermediate node (mini switch) with neighboring node, in the event of node failure the other nodes will immediately switch and continue the communication properly. But the disadvantage of this structure is all the dependencies of network relies on single server, while increase the volume of network will cause lower throughput and packet delay. Along with massive cost it has very complex cabling system [2].

While PortLand and VL2 [11], [28] uses the architectural model of Switch-Centric routing structure, which controls the communication by using network switches for routing, the same anatomy used in three-tier (i. access ii. aggregate iii. core) and fat-tree. This type of architectures are largely used in conventional data centers physical topology. But the three-tier topology schemes are very large, complicated and heavy looking for price and power [2].

The Helios (Hybrid Electrical and Optical structure) [7] combines pod switches with core switches, the architecture propose the reductions of switching elements, cabling, cost, and power consumption. While cThrough [37] architecture by combining the optical and electrical technology, the optical segment routing performs one hop exchange of communication while the electrical segment works like routing in tree, although optical solution has better performance in power saving but rarely used in data centers due to high-priced cost of switches and complex configuration.

Among many ideas for architectural model of data centers, BCube has an innovative idea for high performance server centric network architecture. The major goal for designing this shipping and container based modular data centers (MDC) to provide make available much higher network capacity and enable more efficient network utilization of infrastructure. But while synchronization it needs more laborious efforts for multipath routing and one to all communication. Putting into practice of Centralized Data Centers, the routing control, flow demand estimation, and efficient cum fast scheduling heuristics. The BCube is well designed model and fully support for short term deployment of data centers. Beside higher system with low power, less cooling and manufacturing cost in MDC, it is ultimate and facilitate for all the traffic patterns (i.e. One-To-One, One-To-Several, One-To-All, OR All-To-All) [1]. Dynamic Load Balancing Multi-

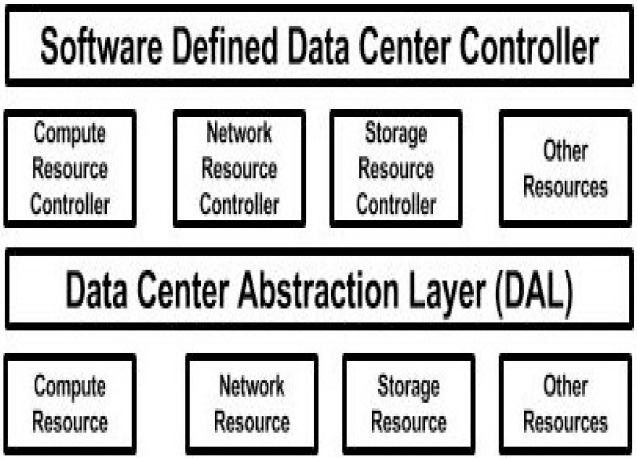

Fig. 1: Abstraction Layer of SDDC

Pathing in Data Center Ethernet (DLBMP) is an alternative solution of STP (Spanning Tree Protocol), DLBMP propose the solution to overcome the proper bandwidth utilization on data link layer (L2) by using Dijkstra algorithm. Since STP has problem of unexpected blockage for links and ports. In DLBMP redundant physical links have deployed to overcome the failure of physical links, it has more performance and can handle 300% more bandwidth capacity while compare with STP. The communication between nodes and traffic are dynamically adjustable, the load balancing are feasible and ease to achieve their efficient link with proper bandwidth utilization [38]. The research study has proved that 30% of total power consumption is required for network equipment and remaining 70% power consumed for servers machines and cooling system inside data centers, for reducing the power cost of network equipment without impact on overall network performance. The innovative idea for VM planner virtualized the network and servers with the proper placement of virtual machines. Dynamically swapping of virtual machines the flow based routing may have appropriate results for saving energy, and the network performance can also be optimized [6]. DGET (Data Grid Environment and Tools) is also peer to peer based middle tool for sharing the resources across different boundaries of network. The predefined access control policies are runtime discovered and access the appropriate resources by exchanging the pre-defined signatures. The hybrid approach for authentication can save the bandwidth utilization and the computational resources at security entity level [5], [19].

II. Motivation

These data centers are playing the different roles of services, and providing diversified infrastructure and platforms, even XaaS-Anything-as-a-Service. i.e. (a) Infrastructure as a Service (IaaS), (b) Platform as a service (PaaS) and (c) Software as a service (SaaS) or may be combination of any three (a,b,c). However these Data Centers are expecting non-blocking, higher speed and reliable connectivity. Including all above they also require with rapid response and higher availability. While most of the storage server connectivity and other server to server connectivity still ranging from 1 to 10 Gbps, although users expectations are much higher for critical data access elasticity from 100 to 1000 Gbps.

The Software defined networking with OpenFlow (by open networking foundation) [34] started the new era of software defined networking. The network has flown out of box and segregated with Control and Management Plane. The logically centralized control plane working with APIs rather than protocols, the open standard software based applications and tools have better flexibility and agility. Traditional security devices installed in conventional data centers (i.e. firewalls) are very expensive to buy and are very laborious to configure. Thousands of rules to implement carefully while configure, taking hours to days at the time of upgrading and hazardous as well. Configuring conventional and large scale data centers require too many cables and paths for connectivity, for example if a data center requires thousands of physical servers to install and each servers cluster needed twenty times more virtual machines for accommodating. Furthermore if necessitate to deploy full any to any communication with proper placement of VMs anywhere anytime. Really if this type of scenario it would be very exhausted to synchronized thousands of physical servers with extensive devices with each entities. Logically the existing network devices are neither fully pace-up with IP forwarding and control planes, nor do the existing routing protocols fulfill the requirement of scalability, portability and security.

The above described scenario can easily be accomplished with Software Defined Data Centers (SDDC). Virtualizing the datacenters we can reduce the storage cost by 60%, network cost by 63%, and Server cost with 41%, Power by 25% and also room space by 33% [18] . The Software Defined Data Centers (SDDC) are mainly divided in four layers (a) Software Defined control Layer (b) Resource control layer (c) Data Center Abstraction layer and (d) Physical Resource layer. All the devices and resources are abstracted and interconnected with each other, these layers are synchronized with adjacent layer to communicate each other. SDDC Controller has connection with Data Abstraction Layer (DAL), where virtualized resources are connected with its adjacent resources (i.e. Showing in Fig:1 Architecture of SDDCs). Each component (i.e. Network Controller to Network Resource) are abstracted with its appropriate layer, these layer and can be monitored and configured easily.

III. Some advantages for Software Defined Data Centers are discussed here;

Apart from Agility, reduction of infrastructure and IT staff , some more reasons are below to switch on Next Generation Data Centers.

1) Packet travels from upper to lower or lower to upper by crossing three layer topology (core to aggregation and aggregation to edge), latency can be minimized by reducing these hops, by removing physical layered devices and replacing with virtual abstraction.
2) Migration of all virtualized devices and VMs can easily be movable and deployable within domain and outside domain of data centers.
3) Traditional architecture of sub-netting can easily be replaced with layer-2 MAC addressing for routing and IP addressing. Therefore the packet travel speed enormously increases from one switch to another. The proper utilization of bandwidth is guaranteed.

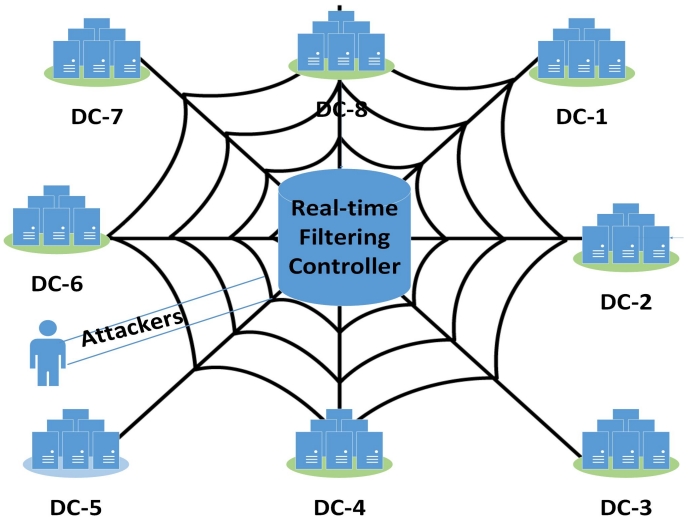

Fig. 2: Geographically Distributed Data Centers

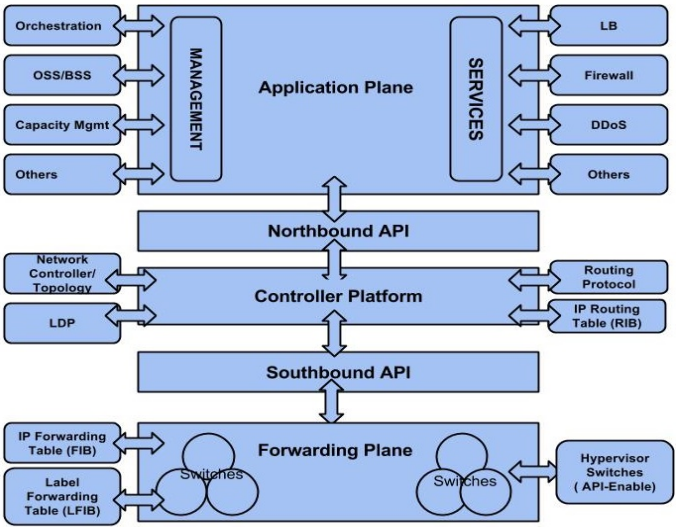

Fig. 3: Management Service Layer Architecture of SDN

4) In case of failure, the redundancy of network connections are always available. The whole process for recovering easily approachable in milliseconds rather than days and weeks. Consequently the higher availability is always there.
5) By using minor coding, implementation of uniform policy and flexible devices can be configured as per desire of administrators. Network scalability made easy and self-dependent. No need to wait for vendors updates, releases and patches.
6) The method for proper utilization and resources can be very efficient, balance between over utilized and un-utilized hardware and applications can be managed.
7) Lower cost for operations, expenses associated with programmer may be day to day basis, no hassle while minor changes occurs.
8) Much increase of revenue for sure. Better services provided during a certain time of period with less expenses and exchange of assets.

IV. Distributed Data Centers' Connectivity

Generally, geographically distributed data centers have three layered connectivity (a) layer-2 and LAN over fiber, (b) Layer-3 with WAN over dark fiber and (c) storage extension. However the distributed data centers connectivity expect the compatibility and fasten support within all network vendors and all technologies. At present DCI uses layer 2 - 3 Virtual Private Networks with Multi Label Switching (VPN-MPLS), Secure Socket Layer with Virtual Private Networks (SSL-VPN), and some other bundles of secure protocols. Various other protocols i.e. IPSEC-VPNs and VxLAN (virtual private networks and bundle of virtual LANs) for secure connections are used.

Beside all the lack of trust and confidence of end users, towards service providers always remain questionable. For user point of view (a) trust exploitation, (b) breach of confidential information, (c) data theft and alteration are the major threats are always there. However the most common security threats for Data Centers are denial of service attack (DoS) or distributed denial of service attack (DDoS) to servers. The unauthorized use for compute resources and session hijacking are also crucial. Beside the above mentioned protocols, expensive devices and complicated architecture the deployment of security is neither flexible nor portable.

Firewall has major role in network security, full written rules and policies having initial boundaries point to access any network. The role of any firewall is not only to secure and protect the internal to external network from unauthorized access, but also protect from internal to internal network. The primary function of firewall is to accept the desired packets and discard or drop the unnecessary packets, the traditional firewalls usually concerned with transport layer (TCP, UDP and ICMP protocols). However the next generation firewalls must deal, not only with permit and deny the port addresses but can accommodate additional features of security controls (i.e. deploy intelligent security tags with own decision and web URL filtering, looking after application layer gateways, intrusion detection and prevention etc).

V. Software Defined Network Security Framework for Data Centers and Basic Security Measures for Controller

However it is necessary to deploy the software defined security engine for software defined data centers, the basic security life cycle standard ADDM (Asses, Design, Deploy and Maintain) followed for keeping alive the process. To overcome the upcoming weaknesses and vulnerabilities, the domain specific distributed decision point (controller) needs to be well structured and tightly couple. All the components are fully dependable upon controller. Openflow controller performs a number of important functions and manage all the activities whose performed inside and outside of domain, although controller has default instruction set of commands to handle with network activities (i.e. match, add, modify, translate, forward and drop). However administrators' make

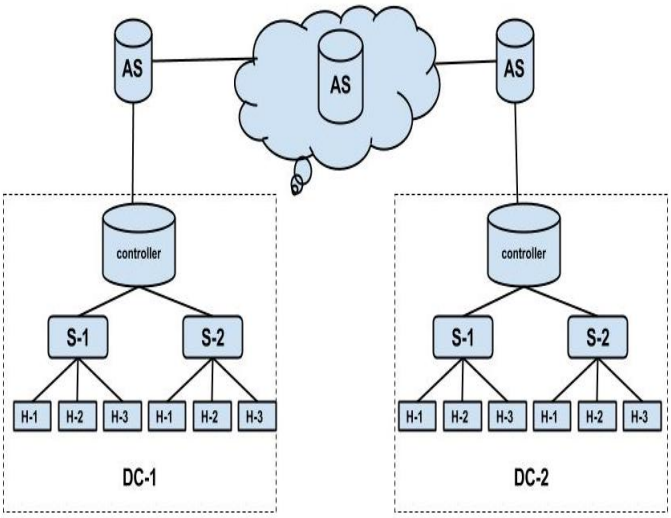

Fig. 4: Authentication Server (AS) for SDDCs

sure with its security, availability and continuous maintenance while necessary. Maintaining two way trust between all dependable component by sharing the private and public keys for authorization and authentication the network. Continuously observe and analyze the logs for forensics and remediation for the future policies.

VI. Deploy SDN Controller with Kinetic

Following listed many open source software architectural tools for security are easily available and may be compatible with software defined networking i.e. (a) FRESCO-SEK (Framework for Enabling Security Control in Openflow Networks and Security Enforcement Kernel) [32], (b)snort [31], (ii)SANE [4], (c) BotHunter [12] (d) OpenSafe [10], (e) Nettle [35], (f) Procera) [21] and (g) Ethane [3]

While working with above listed programming tools, the experience was good but most of listed tools had different issues while working with. (i.e. (a) either complicated coding to understand, (b) lack of support while deploying the security policies OR (c) compatibility issues with openflow and mininet emulator. Taking advantage of Kinetic and Frenetic programming tool for network domain, we appreciate for specific friendly programming tool for SDN, the ease of expressing the dynamic policies in controller and trigger based programming can easily be implemented (especially for intrusion detection and bandwidth limitation).

In Kinetic the event based dynamic policies can be written as per the desire of network administrators. Inside the abstraction of Finite State Machine (FSM), the control of data usage, host authentications and intrusion detection are the major areas to dealt with. Also the Frenetic [9] high-level language for network has more capabilities of real time traffic analysis and packet logging on networks.

The well defined security policies in Frenetic family [9] having NuSMV [17] generated bunch of logic for verification, its more than helpful for real time filtering and security implementation in Software Defined Data Centers.

Rather to deploying, combining other tools and writing code from scratch, we are using the same tool with little bit change in coding. The key features of this tool having better ability to perform Network intrusion prevention system (NIPS) and network intrusion detection system (NIDS), also handle the real time traffic analysis and packet logging very efficiently.

The proposed framework will have following step to perform.

VII. Steps for architecture and deployment

- By referring the Fig. 4, we simulate couple of data centers on couple of virtual machines, we chose Mininet emulator for data center environment and Openflow based controller for (a) Customized Topologies and Traffic Generator i.e. three tier topology with core, edge and hosts servers (b) Openflow written rules (c) VM handler for connecting the other topology of data centers or cloud (d) virtual network services and network flow for different types of traffic for load (e) web platform for traffic monitoring (d) GUI based open source tool for traffic monitoring.
- To implement the security for Software Defined Data Centers, we implement the middleware architecture frame work with Kinetic using Pyretic coding. Initially the AS layers of gateways are connected to the controller, and the controller synchronized with data centers for real time filtering. AS (Authentication Server) receives the the authorized traffic from local domain controller (i.e. DC-1) through S1 and S2 whiling communicating the different machines, from H-1 of S-1 to H-3 of S-2. The packets are forwarded from Controller to centralized Authentication Server for decision. On the other-hand if the H1 of DC-1 wants to communicate with H1 the traffic will go like this pattern From H1 >> S1 >> Controller-1 >> AS local-1 >> AS Central >> AS Local-2 >> Controller-2 >> S1 >> H1. On the other hand the same process will be repeated if any host wants to talk from DC-2 to DC-1, even the same process from host-host inside of of DC-1 OR DC-2. Keeping in view of security the authentication process is rely on Switch >> Controller >> and all Authentication Servers (AS)(as shown in Fig. 4). Subsequently the authorized traffic goes to adjacent gateway from source to destination data centers and rest of packets are either blocked or dropped.
- The Pyretic has provision to deal with several kind of policies to be monitored (including forwarding behavior, byte counts, packet counts and drop the raw packets). The python based controller interface having high level flow rules for security directives i.e. block, deny allow, redirect, undo, constrain, quarantine, info the architecture can be extended. The API based programmable controller is wholly solely responsible to force module permissions, packet privilege permissions for out and all other modifications which are implemented on switch.
- Conflict analyzer tool has basic three command (a) Add, (b) Modify (c) Delete commands. While using

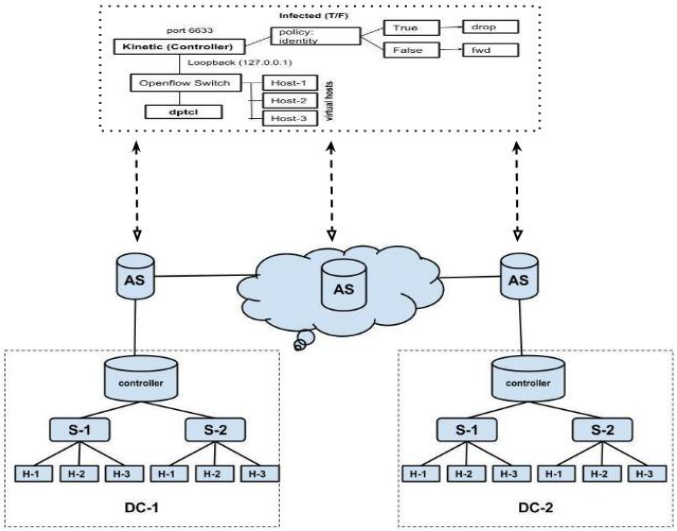

Fig. 5: FrameWork for middle security tool for Data Centers

module design the five tools can be working for module design for communication i.e. input, output, parameter, event and action . The KINETIC with openflow, is the easily deploy-able framework and it will work like real firewall, the scanning and detection packets, Distributed Denial of Service attack, intrusion detection and prevention are the major features for the security of data centers.

- Registration of APIs: Can create trust based distributed policy for synchronization, exchanging digital signature duly signed by all data centers (assign by Network Administrators). The random asymmetric key generator generate the key and export the same key for authorization to all participants for handshaking. The database table checks if the source input is not matched with the given criteria and key, than it detects and block the unwanted malicious entries and blacklist the MAC or IP addresses.
- Multiple network policies can be written in one go either sequential or parallel, the source to destination network can be targeted by predefined policies and triggered at run-time level.

VIII. Management Actions

The background code of Pyretic is written in Python, usually the policies are defined with .py extension, theoretically it takes an input of packet and return set of packets. List of policies i.e. match, drop, modify, forward, and flood are in frequent use. While deploying, following few commands are given for defining the policies. Logic implementation and finite state machines(FSM) stated when and where to communicate basis with specific MAC or IP addresses. if listed matched packet recognized than forwarded otherwise the same packet will be dropped.

Few examples while Defining the policies, logic and others :

redirectToGardenWall [17]

```
def redirectToGardenWall():
        client_ips =
        [ip('10.0.0.1'), ip('10.0.0.2')]
        rewrite_policy =
        rewriteDstIPAndMAC
        (client_ips, '10.0.0.3')
        return rewrite_policy
```

Defining Logic:

If Infected than drop otherwise forward [17]

```
def policy(self):
        self.case(test_and_true
        (V('exempt'),V('infected')),
        C(redirectToGardenWall()))
        self.case(is_true(V('infected')),
        C(drop))
        self.default(C(identity))
```

Describing Finite State Machine (FSM) [17]

```
self.fsm_def = FSMDef(
        infected=FSMVar(type=BoolType(),
        init=False,
        trans=infected),
        exempt=FSMVar(type=BoolType(),
        init=False,
        trans=exempt),
        policy=FSMVar(type=Type(Policy,
        {drop, identity,
        redirectToGardenWall()}),
        init=identity,
        trans=policy))
```

IX. Related Work

There have been a recent flood of new research on the security of virtualized networking as well as on Software Defined Data Centers. However very few of the researchers are looking towards the security of next generation of data centers.

Security Enforcement Kernel for OpenFlow Networks. The FortNox is new security kernel of openflow controller, this model implemented for the security for the flow rules with predefined policies and the performance measure for the conflict of UDP packets, also the distributed security policies are synchronized with barrier messages. [29]

FRESCO [32] is well defined modular and composable security service for software defined networking, this application service address to firewall, DDoS detection and scanning the behavior of network. The security applications and policies are easily deployable and reduced code is used. This research is very influence-able to my research.

OpenFlow: A Security Analysis [22] Analyzed the security model for the openflow, The Microsoft: STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privileges) based model used for analysis the security behavior in openflow based controller.

Towards a Secure Controller Platform for OpenFlow [24] The permission based system for flow is introduced. The third party tool PermOF is used for different type of 18 permission are deployed and analyzed the security.

AVANT-GUARD [33]: Scalable and Vigilant Switch Flow Management in Software-Defined Networks: new tool is proposed as a guard of Openflow controller, specially designed the denial of service attack and activity. The trigger based tool for existing packets flow in Software defined Networking.

Splendid Isolation: A Slice Abstraction for Software-Defined Networks [16] The mechanism for isolation and slicing of network is proposed using openflow, the programmable network on abstraction layer is created. The proposed algorithm segregated each network with other network, for security and privacy the traffic had made with its associated slices and nodes.

Machine-Verified Network Controllers: [13] OpenFlow based model for reasoning the network, Coq proof assistant and machine verified complier which work on runtime ssyetm for high level network programming. The main purpose to obtain the result in the specific framework time by implemented logics on controller. The controller correctness will reduce the proof obligation liveness properties.

Languages for Software-Defined Networks: [8] A well written code for compiler controller in Frentic language for Software Defined Networking. The controller can take decision on runtime and triggered the policies . The paper trying to proof for controlling, management and monitoring policies.

Abstractions for Network Update: [30] The paper presented for the re-usability of code and abstraction updation in software defined networking. The guaranteed updates for network on real time and preserve well defined behavior while transitioning and configuration. The optimized code will reduce the overall work for programing.

Logic Programming for Software-Defined Networks: [20] Flog: An SDN Logic Programming Language, special purpose new language designed for event based logic in software defined networking. The code can accommodate three main features predicate, action and priority.

Frenetic: A Network Programming Language: [9]. Its high-level language for network has more capabilities of real time traffic analysis and packet logging on networks.

The Software defined programming high level language is designed for logic and control , special purpose new language designed for event based logic in software defined networking. The code for Network Operating system can accommodate to Openflow based controller in Python programming. A run-time system will handle all of the details related to installing and un-installing the rules in low level programming.

Composing Software-Defined Networks: [27] Pyretic is a new language, this allows to software defined networking and the programmers can build large, sophisticated controller in openflow. The network operator can code as per their desire without complication and complexities.

A Compiler and Run-time System for Network Programming Languages: [26] An innovative idea for inventing the new language called NetCore, for openflow for forwarding the policies.

X. Conclusions

This paper has described to design the software defined security for distributed data centers connectivity, especially software defined data centers. Since there may be many other middleware tools required for the security of virtualized networking, the network community is also struggling to work the different aspect securities. Our little effort may also contribute to step towards contribution for the security of SDDC and virtual networking.

For the security framework tool for Data Centers we use Kintetic and pyretic as a middleware centralized tool. We generated real world scenario for different types of topologies of Data Centers on different virtual machines. Also generate the different wanted and unwanted traffic for exceptions and rejection. This exercise done to judge the different behavior of network, for variation we just need minor coding in different APIs. To simulate couple of data centers on physical machine (a) we chose Mininet emulator for data center environment, (b) Openflow based controller for data centers topologies and (c) traffic generator. The response of security engine was fine after implementation of all the different rules on controller. Redeploying the security policies on security framework with different angle, it works properly as a security engine. For GUI based monitoring and traffic load, we chose Wireshark analyzer for packet capturing and analyzing. We also tried to experience for the migration of virtual machines to observe the behavior of post migration impact.

XI. Future work

We would like expand this research project with much larger scenario about data center virtualization and its security, specially to work on the portability and scalability. Also the impact on different domains of security while moving from one to another. In parallel we are also working on availability and reliability with test bed scenarios.

References

[1] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan, *Data center tcp (dctcp)*, ACM SIGCOMM computer communication review **41** (2011), no. 4, 63–74.

[2] Kashif Bilal, Samee U Khan, Limin Zhang, Hongxiang Li, Khizar Hayat, Sajjad A Madani, Nasro Min-Allah, Lizhe Wang, Dan Chen, Majid Iqbal, et al., *Quantitative comparisons of the state-of-the-art data center architectures*, Concurrency and Computation: Practice and Experience **25** (2013), no. 12, 1771–1783.

[3] Martin Casado, Michael J Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker, *Ethane: Taking control of the enterprise*, ACM SIGCOMM Computer Communication Review **37** (2007), no. 4, 1–12.

[4] Martin Casado, Tal Garfinkel, Aditya Akella, Michael J Freedman, Dan Boneh, Nick McKeown, and Scott Shenker, *Sane: A protection architecture for enterprise networks.*, Usenix Security, 2006.

[5] Tariq N Ellahi, Benoit Hudzia, Liam Mcdermott, and T Kechadi, *Security framework for p2p based grid systems*, Parallel and Distributed Computing, 2006. ISPDC'06. The Fifth International Symposium on, IEEE, 2006, pp. 230–237.

[6] Weiwei Fang, Xiangmin Liang, Shengxin Li, Luca Chiaraviglio, and Naixue Xiong, *Vmplanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers*, Computer Networks **57** (2013), no. 1, 179–196.

[7] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat, *Helios: a hybrid electrical/optical switch architecture for modular data centers*, ACM SIGCOMM Computer Communication Review **41** (2011), no. 4, 339–350.

[8] Nate Foster, Arjun Guha, Mark Reitblatt, Alec Story, Michael J Freedman, Naga Praveen Katta, Christopher Monsanto, Joshua Reich, Jennifer Rexford, Cole Schlesinger, et al., *Languages for software-defined networks*, Communications Magazine, IEEE **51** (2013), no. 2, 128–134.

[9] Nate Foster, Rob Harrison, Michael J Freedman, Christopher Monsanto, Jennifer Rexford, Alec Story, and David Walker, *Frenetic: A network programming language*, ACM SIGPLAN Notices, vol. 46, ACM, 2011, pp. 279–291.

[10] Aaron Gember, Jeffrey R Ballard, Brian Kroth, and Aditya Akella, *Opensafe: Hardware-based network monitoring using software control*.

[11] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, and Sudipta Sengupta, *Vl2: a scalable and flexible data center network*, Communications of the ACM **54** (2011), no. 3, 95–104.

[12] Guofei Gu, Phillip A Porras, Vinod Yegneswaran, Martin W Fong, and Wenke Lee, *Bothunter: Detecting malware infection through ids-driven dialog correlation.*, USENIX Security, vol. 7, 2007, pp. 1–16.

[13] Arjun Guha, Mark Reitblatt, and Nate Foster, *Machine-verified network controllers*, ACM SIGPLAN Notices, vol. 48, ACM, 2013, pp. 483–494.

[14] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu, *Bcube: a high performance, server-centric network architecture for modular data centers*, ACM SIGCOMM Computer Communication Review **39** (2009), no. 4, 63–74.

[15] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu, *Dcell: a scalable and fault-tolerant network structure for data centers*, ACM SIGCOMM Computer Communication Review **38** (2008), no. 4, 75–86.

[16] Stephen Gutz, Alec Story, Cole Schlesinger, and Nate Foster, *Splendid isolation: A slice abstraction for software-defined networks*, Proceedings of the first workshop on Hot topics in software defined networks, ACM, 2012, pp. 79–84.

[17] http://kinetic.noise.gatech.edu/index.html.

[18] http://www.idc.com/.

[19] Benoit Hudzia, Liam McDermott, TN Illahi, and M Tahar Kechadi, *Entity based peer-to-peer in a data grid environment*, arXiv preprint cs/0608112 (2006).

[20] Naga Praveen Katta, Jennifer Rexford, and David Walker, *Logic programming for software-defined networks*, Workshop on Cross-Model Design and Validation (XLDI), 2012.

[21] Hyojoon Kim and Nick Feamster, *Improving network management with software defined networking*, Communications Magazine, IEEE **51** (2013), no. 2, 114–119.

[22] Rowan Klöti, *Openflow: A security analysis*, Proc. Wkshp on Secure Network Protocols (NPSec). IEEE (2013).

[23] Jonathan Koomey, *Growth in data center electricity use 2005 to 2010*, A report by Analytical Press, completed at the request of The New York Times (2011).

[24] Diego Kreutz, Fernando Ramos, and Paulo Verissimo, *Towards secure and dependable software-defined networks*, Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, ACM, 2013, pp. 55–60.

[25] Marcel Margulies, Michael Egholm, William E Altman, Said Attiya, Joel S Bader, Lisa A Bemben, Jan Berka, Michael S Braverman, Yi-Ju Chen, Zhoutao Chen, et al., *Genome sequencing in microfabricated high-density picolitre reactors*, Nature **437** (2005), no. 7057, 376–380.

[26] Christopher Monsanto, Nate Foster, Rob Harrison, and David Walker, *A compiler and run-time system for network programming languages*, ACM SIGPLAN Notices **47** (2012), no. 1, 217–230.

[27] Christopher Monsanto, Joshua Reich, Nate Foster, Jennifer Rexford, David Walker, et al., *Composing software defined networks.*, NSDI, 2013, pp. 1–13.

[28] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat, *Portland: a scalable fault-tolerant layer 2 data center network fabric*, ACM SIGCOMM Computer Communication Review, vol. 39, ACM, 2009, pp. 39–50.

[29] Philip Porras, Seungwon Shin, Vinod Yegneswaran, Martin Fong, Mabry Tyson, and Guofei Gu, *A security enforcement kernel for openflow networks*, Proceedings of the first workshop on Hot topics in software defined networks, ACM, 2012, pp. 121–126.

[30] Mark Reitblatt, Nate Foster, Jennifer Rexford, Cole Schlesinger, and David Walker, *Abstractions for network update*, Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, ACM, 2012, pp. 323–334.

[31] Martin Roesch et al., *Snort: Lightweight intrusion detection for networks.*, LISA, vol. 99, 1999, pp. 229–238.

[32] Seungwon Shin, Phillip A Porras, Vinod Yegneswaran, Martin W Fong, Guofei Gu, and Mabry Tyson, *Fresco: Modular composable security services for software-defined networks.*, 2013.

[33] Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei Gu, *Avant-guard: scalable and vigilant switch flow management in software-defined networks*, Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM, 2013, pp. 413–424.

[34] OpenFlow Switch Specification Version, *1.3. 2. open networking foundation*.

[35] Andreas Voellmy, Ashish Agarwal, and Paul Hudak, *Nettle: Functional reactive programming for openflow networks*, Tech. report, DTIC Document, 2010.

[36] Chao Wang, *Survey of recent research issues in data center networking*.

[37] Guohui Wang, David G Andersen, Michael Kaminsky, Konstantina Papagiannaki, TS Ng, Michael Kozuch, and Michael Ryan, *c-through: Part-time optics in data centers*, ACM SIGCOMM Computer Communication Review, vol. 40, ACM, 2010, pp. 327–338.

[38] Yang Yu, Khin Mi Mi Aung, Edmund Kheng Kiat Tong, and Chuan Heng Foh, *Dynamic load balancing multipathing in data center ethernet*, Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on, IEEE, 2010, pp. 403–406.