EasyChair Preprint

№ 9162

Software Defined Networking: Architectures, Research Issues and Security

Konstantinos Mavrommatis

October 26, 2022

TITLE

Software Defined Networking: Architectures, Research Issues and Security

Author: Konstantinos I. Mavrommatis

School of Engineering, Informatics Computer Engineering Department, University of West Attica, Greece, kmavrom@uniwa.gr

Abstract

Software-defined Networking (SDN) is a new network architecture that decouples the control plane from the data plane. Scalability of the control plane with respect to network size and update frequency is an important problem that has been addressed by previous studies from a variety of viewpoints. However, the solutions found in these studies may be only locally optimized solutions. To find a globally optimized solution, a broader viewpoint is required: one in which various SDN architectures can be evaluated and compared.

Additional Keywords and Phrases: SDN, NVF, NVFI, OVS, SDS, ARM

Introduction

"Software-Defined" is the ability to control and manage some or all of the functions of a system using exclusively software. Such a system, which will be determined by a software, assumes the following characteristics:

- Removal of Physical Resources: Provides a set of Application Programming Interfaces (APIs - Application Platform Interfaces) aimed at managing the system without using physical resources (eg CPU, Ram, disk).
- Automation of Actions/Controls: Includes applications that perform actions and controls depending on the operating conditions or of the software monitoring system.
- Reconfiguration of Resources: This is the ability to adapt the resources of the system according to its needs-requirements.

Network Functions Virtualization (NFV)

Network Functions Virtualization (NFV) is defined as the network architecture that uses virtualization technologies to simulate network node functions into building blocks that can be connected together to create telecommunications services. It is based on traditional computer virtualization techniques, but also has differences. A virtual network function (VNF) can include one or more virtual machines running different software and processes on shared servers, switches and storage devices, or even in a cloud environment, rather than needing specialized devices to every function of the network.

The NFV operating framework consists of 3 main parts:

- VNFs are software implementations of network functions. They are used as modules, which are building blocks of an NFV architecture.
- Network Operations Virtualization Infrastructure (NFVI) is the set of hardware and software components that create the environment in which NFV is deployed. Processing (virtual and physical), storage resources, and virtualization software are key parts of NFVI. NFV infrastructure can span more than one location, and the network that provides connectivity between those locations is considered part of the infrastructure.

- The management and orchestration framework of the virtualization of network functions (NFV MANO - NFV Management and Orchestration) is the set of all functional units, the data they use, the reference points and the interfaces through which they exchange information with for managing and orchestrating VNFs and NFVI.

Software-Defined Network (SDN)

SDN is based on the premise that by separating the control of network functions from the network devices themselves (switches, routers, firewalls, etc.), it can address several limitations related to today's integrated, closed and proprietary networking infrastructure. The adoption of virtualization technologies and the use of voice, video and data over an IP network created the need for such a change in networking standards. In Figure-1 we observe a typical network using SDN in a data center environment [1].

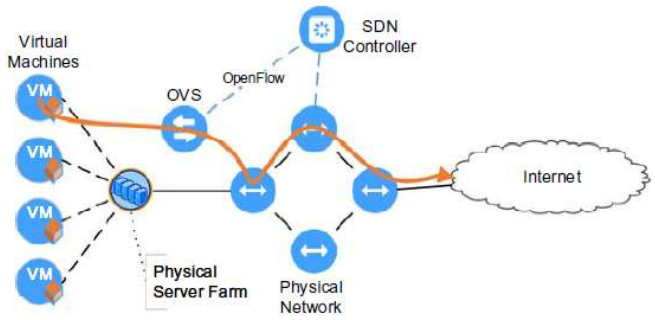

Figure 1: Typical network implemented using SDN (via sciencedirect.com)

Four users have 4 VMs on the same physical host, with each VM connected to the same OVS. The data frames coming from the VMs are tagged with a VLAN ID in order to separate them from the 4 users. OVS (see subsection 5.2.5), uses flow rules received from the SDN Controller to determine how to handle data movement. The separation of control and data results in switches becoming "dumb" forwarding devices and logical control being applied to a central controller. This allows the network administrator to have a more detailed analysis of traffic control, as well as respond to changing network demands in a more efficient manner. In Figure-2 we have a simplified view of the SDN architecture.

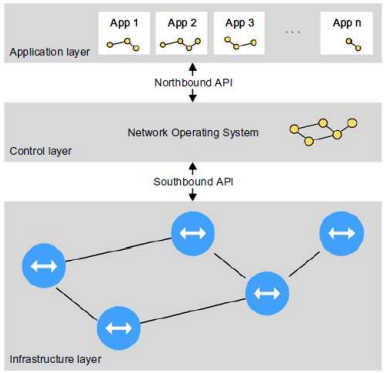

Figure 2: SDN Architecture (via sciencedirect.com)

We notice that the functional architecture of SDN is divided into three layers: Application layer (application layer), Control layer (control layer) and Data layer (infrastructure layer).

SDN Advantages

The use of SDN has increased in modern times due to the following advantages it offers:

- Central management: the management of the network is based on a central point, the controller, which knows the topology of the network and has a close relationship with the software, with the property of the unique equipment.
- Flexible technical support: it is allowed to add new applications and functions, as well as upgrades, using open source tools regardless of its manufacturer. Also, the central e-control allows error prediction and its timely correction.
- Capital saving: the cost of a company for the provision of information equipment is significantly reduced, since the needs are now limited to the controller.
- Saving of human resources: likewise, the need for human resources is reduced due to a reduction in the tasks required.
- Increased reliability: due to the automation of tasks, SDN networks become more reliable and functional.
- Innovation: it is possible to develop and test new services without affecting the operation of the network.
- Network virtualization: by reducing the complexity of the hardware, it becomes more affordable to switch to virtual equipment without large a-requirements.

Control Layer - SDN Data

The control layer of an SDN installation consists of one or more SDN controllers, which control the underlying vSwitches or forwarding devices. They also monitor the network environment and traffic in real time. The controllers interact with the rest of the SDN infrastructure using three communication interfaces, named south, north, and east/west. The division in their functions is as follows [2]:

- Southbound interface: allows the controller to communicate, interact and manage forwarding elements. The most common implementation of the southbound interface is OpenFlow.
- Northbound interface: allows applications at the Application level to program the controllers.
- East/west interface (east/westbound interface): intended for communication between groups of controllers.

The data layer in the SDN architecture is intended to enable the transfer of data from the sender to the receiver(s).

OpenFlow

OpenFlow is a protocol between the Control and Forwarding layers in the SDN architecture and is the most widespread for implementing such a network. A basic OpenFlow architecture consists of hosts, controllers and switches with OpenFlow capabilities. An OpenFlow switch is not limited to being a layer-2 device as it is in common hubs. Also, the controller communicates with the above switches with an OpenFlow API (Application Programming Interface).

SDN Controller

The controller is the brain of SDN operation. It is located between the Data layer and the Application layer. It takes responsibility for creating each flow in the network by installing flow entries on switches [3].

Flow entries can be added to a Data layer device, either in proactive mode where flow rules are sent to devices until the controller recognizes them, or in reactive mode where the controller sends flow entries only when required.
Some popular SDN controllers based on OpenFlow are:

- NOX, was one of the first controllers. There are many variants to date: NOX-MT, QNOX, Fort-NOX and POX, each with its own characteristics.
- OpenDaylight (ODL), is open source and used mainly in industry.
- OpenContrail, works efficiently with virtual routers and has an analysis engine.
- Beacon, based on the JAVA programming language.
- Floodlight, similarly based on JAVA and supports a number of OpenFLow switches. Its use is widespread by large companies such as Intel, Cisco and IBM.
- Ryu, is an open-source SDN controller based on Python.
- FlowVisor, is special purpose that supports a decentralized controller. It has motion shaping functions.

4.2.5 Open Virtual Switch (OVS)

OVS is an open source, multi-tiered virtual distributor. Its implementations consist primarily of flow tables, with each flow entry having matching conditions and associated actions. It communicates with the controller with a secure channel and uses the OpenFlow protocol. It has also been integrated into major cloud orchestration systems such as OpenStack and CloudStack. The architecture of OVS is shown in Figure-3:

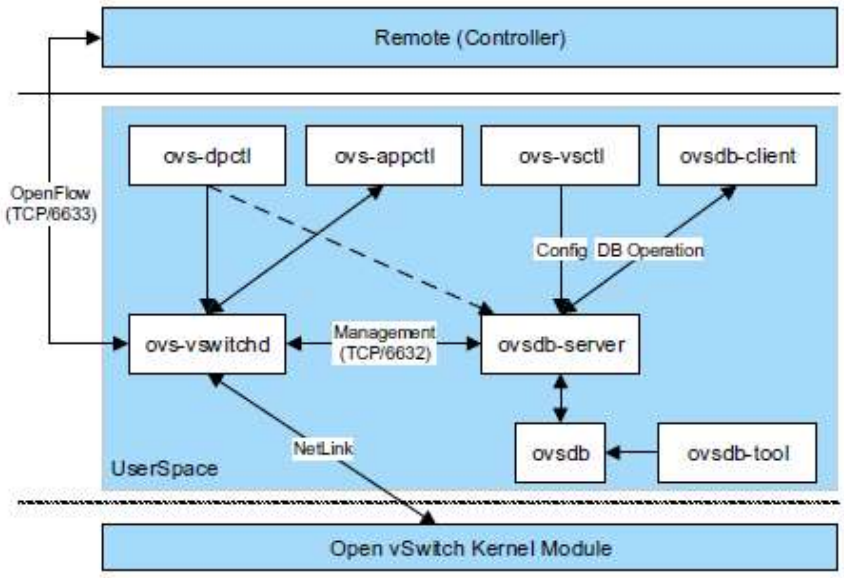

Figure 3: Open Virtual Switch architecture (via sciencedirect.com)

Routing in SDN

In classical networking, topology information, i.e. network connectivity and hierarchy, is forwarded to the network for routing calculation. This results in high bandwidth, high process utilization, increased storage usage, and complex and non-scalable networking. In an SDN environment there is no single device that is defined exclusively as a router. Connection states are periodically checked by the controller. Routing is done at the SDN Control layer.

SDN Security

Network Functions Virtualization (NFV) has emerged as a technology that provides a virtual implementation of hardware such as firewalls, routers and intrusion detection systems (IDS). The implementation is done with virtual machines (VMs) running on top of the physical server of the cloud infrastructure.

SDN acts as an enabling technology for NFV. Despite the benefits that both offer, security, privacy and trust management are issues that need to be addressed. We will address the security challenges faced by SDN networks.

SDN is widely applied in cloud environments and data-center networks. The adoption of its technology will likely have a benefit in the area of cloud security as well. The way SDN is designed attracts several attackers with several attacks, such as distributed denial of service (DDoS) attacks, against the controller. The SDN architecture has three layers, each of which can accept multiple attack vectors (methods of entering a computing system) as well as the interfaces between them. Below are some of the attack vectors:

- Application Layer: applications developed for SDN functions may have security vulnerabilities. Security issues that may exist in typical web applications apply here as well and may allow network-wide propagation.
- Control Layer: consists of one or more controllers, e-applications and plugins to handle different kinds of protocols. The attacker can create fake IP addresses and send a huge amount of traffic to the controller. Thus, the communication between switch and controller may become saturated and "crash" the system.
- Data layer: attacks here are done with infected packets. Attackers observe delay in communication between control layer and data layer applications using special packets. So they will be able to identify functions and type of the controller. Attack y-there is a case for the switches to accept as well. The switch is responsible for the updates of data flow rules and contains memory, which is limited and can overflow by creating a large number of rules.
- Communication Channels: the communication channels between switches with controllers (Southbound API) and controllers with Application level (North-bound API) are likely to receive attacks, such as interception/covert modification of traffic between hosts.

SDN Security Measures

In order to counter potential attacks on our network, some features/mechanisms must be designed to prevent them. Specifically:

- Replication: Replication of applications and controllers can help in case of failure due to high traffic or software vulnerability.
- Diversity: using only one type of software or operating system makes it easier for attackers to converge on a target. Diversity improves robustness and makes it less susceptible to intrusions.

• Automated Recovery: in the event of security attacks, which lead to service disruption, proactive and reactive security recovery mechanisms have the ability to contribute to the smooth continuation of service availability.
• Dynamic Device Association: the association between the controller and devices, such as the OpenFlow switch, should be dynamic in nature. If one controller fails, the switch should be able to dynamically associate with another backup in a fail-safe manner.
• Controller-Switch Trust: an establishment mechanism between controller and distributor is important to deal with cases of spurious flows introduced by malicious switches.
• Controller-App Plane Trust: software components change behavior due to a change in the environment. Older software can also introduce security vulnerabilities. Controller and Application layer components should use independent trust management mechanisms.
• Security Domains: security domains help segments the network into different levels of trust and limit threats.

Software-Defined Security (SDS)

Network security is defined as the set of actions to protect the operation and integrity of the network and data. It is not limited to the core/host of the network, but includes the hardware, software and human factors inside and outside an organization [4].

There are various tools and techniques to protect against malicious attacks, such as network firewall, web application firewall and intrusion detection system (IDS).

Software-Defined Data Centers (SDSs) have the ability to allocate compute, storage and networking resources dynamically. Although data centers in the modern age are achieved in a virtual way, we observe that they are lagging behind in the security of the virtual infrastructure. Current information security methods are too rigid and static to support the rapid evolution of technology. SDN provides next-generation security services that enforce proactive monitoring, analysis, detection and prevention of threats to the virtual infrastructure. The set of these services is called Software-Defined Security (SDS). The security infrastructure in SDS should adapt to changes in network infrastructure and application services. In the long term, the adaptive security infrastructure will be based on SD models, which will provide protection against emerging threats.

Some key features of SDS are:

- Site independent security for information and workloads.
- Matching security controls to the risk profiles they protect against.
- Enabling automated security orchestration and management policy.
- Removal of time- and error-prone human middleware through a higher level of automation.
- Enable information security professionals to focus on policies and detect advanced threats using programmable security components.
- Enable security at scale and protect dynamic cloud-based workloads.
- Adaptation of security controls at high speeds.

Distributed Security and Micro-segmentation

Distributed security is a method of breaking down the traditional data center and cloud network into logical components and managing each component separately. Such an architecture enables security at the sensitivity of each network, subnet, and application task. Micro-segmentation is one such approach to a decentralized security framework. Network

micro-segmentation provides abstraction in subnet traffic control, making security management architecture simpler in complex SDDCs with multiple applications and workloads. An additional benefit is preventing attacker lateral movement by introducing a zero-trust architecture.
Micro-segmentation is the method of creating secure zones in data centers and cloud deployments to isolate workloads from each other and secure them individually. It aims to make network security more granular. With micro-segmentation, policies are applied to individual workloads for greater attack resilience. The rise of SDN and NFV has paved the way for micro-segmentation. Micro-segmentation is designed to address two security issues:

- Identification of network traffic above the Transport layer, e.g. user, application, etc.
- Policy-based network traffic control.

A basic tool for network security is the firewall. Firewall is a common terminology widely used in the security field. It is a key system security component to provide inspections of various network components. In addition to protecting against "malicious" North - Southbound traffic carried in and out of a trusted domain, firewalls have been used to filter West - Eastbound traffic into a trusted domain to prevent malicious traffic moving laterally to explore internal vulnerabilities. The level of sensitivity of protected networks can be at the level of a subnet, a VLAN, an interface, a contact or a data flow, which is usually implemented through a virtual networking approach, which as we mentioned above, is called micro-segmentation.
The classic form of a firewall (Figure-17) interposes two networks and filters traffic between them according to some non-security policy. Conventional firewalls depend on limiting the topology of networks. At the point of inspection, the firewall divides the networks into two parts, internal and external. The firewall cannot filter traffic on the internal network and considers all its hosts untrusted, while the opposite is true for the external network, where they are all untrusted.

In today's era, with the expansion of connectivity, high-speed lines, multiple entry points and telecommuting, firewalls face major challenges such as:

- Firewalls do not protect networks from insider attacks.
- High and wide internet connectivity makes them obsolete models.
- End-to-end encryption is a firewall threat.
- Some protocols are not detected by the firewall.
- Firewalls can become network bottlenecks.
- Unauthorized entry points exist, which can and do bypass firewall security.

In order to solve the above problems, a "distributed firewall" was created. A set of firewalls, located on the same host and configured and managed centrally, make up a distributed firewall (Figure-18). In this architecture the security policy is still defined centrally but is enforced at each endpoint. Central policy determines which connection is allowed or denied. It is then distributed to all endpoints, where required. Three components are needed for distributed firewalls:

- Security policy language.
- Policy distribution plan.
- Authentication and encryption mechanism.

It is worth mentioning the main advantages of using distributed firewalls:

- Topology independence: ability to provide protection to hosts across topology boundaries.

- Protection against internal attacks: for the host computer there is no difference between "internal" and "external" networks. The hosts are identified by their encrypted certificates, which also reduces possible identity spoofing.

Preventive Security

Proactive security refers to a new security management technique where the attack surface changes over time, as opposed to the classic philosophy of detecting, preventing, monitoring and remediating threats. This creates an asymmetric disadvantage for attackers. MTD is a term coined to generalize different preventive security mechanisms, such as introducing diversity into the network topology, software, operating system, or randomization of memory, so that intrusion becomes more difficult for the attacker. attacking.

Security Policy Management

A large organization may have different teams, each of which is responsible for managing a small part of the network. There is often a lot of interdependence between the access control policies of different groups. The management of security policies detects and corrects in an automated way the differences between security policies of each group.

Attack Representation Methods

The interaction between different applications and the impact on the attack surface of the dynamics of programs and the constantly changing workload in a cloud environment must be expressed in such a way that network security managers can make an informed decision and recognize the multitude of threats. Attack Representation Methods (ARMs), such as attack graphs and attack trees, help represent complex information in an easy and understandable format.

Conclusion

Software Defined Networking is a promising emerging architecture for many networking environments such as data centers, enterprise networks, campus networks, cloud networks, and WAN. The major advantages of SDN are its programmability and agility. However, the scalability issues in the control plane is one major problem in SDN that needs more research attention. In the paper, we have firstly given an overview of the SDN architecture and OpenFlow protocol along with its support mechanisms.

References

[1] Feamster N. The Road to SDN an intellectual history of programmable networks. Publisher ACM. 2013; 11(12):1–21.

[2] Caraguay ALV, Lopez LIB, Villalba LJG. Evolution and Challenges of Software Defined Networking. IEEE Communications Magazine; 2013. p. 1–7

[3] Xia W, Wen Y, Foh CH, Niyato D, Xie H. A Survey on Software-Defined Networking. IEEE communication surveys and tutorials. 2015; 17(1):27–51.

[4] Open Flow Switch Specification. Open Networking Foundation ONF. p. 106 Available from: https://wand.net. nz/~brad/papers/openflow-spec-v1.5.0.pdf. 19/12/2014