SDN: Development, Adoption and Research Trends

(A Survey of Research Issues in SDN)

Lav Gupta, lavgupta at wustl.edu (A paper written under the guidance of Prof. Raj Jain) Download 

Abstract

Controlling and managing networks has become a highly complex and specialized activity. Network operators are struggling to cope with integration of different types of networks, while meeting the challenges of increasing traffic. The traditional network tends to be rigid. Once the forwarding policy has been defined, the only way to change it is by changing the configuration of all the affected devices. This is time consuming and puts a limit on scalability and meeting challenges of mobility and big data. In this context software defined networking (SDN) is being looked upon as a promising paradigm that has the power to change the way networking is done. By centralizing control, and making forwarding nodes simple, SDN offers flexible control over the traffic flows and the policies networks use to manage these flows. Along with the excitement, there have been apprehensions regarding SDN. The perceived risks associated with SDN have prevented faster adoption so far. There have been a number of trial deployments but very few production networks. In this paper we would discuss the important implementation issues in SDN, the requirements it would fulfill in two fast growing areas â€“ wireless networks and clouds and the research that is taking place in these areas. We discuss, in some detail, the enhancements required in the SDN framework as far as security is concerned. Finally, we explore the future directions that SDN research is expected to take

This paper does not aim to be a comprehensive of all aspects of SDN but after starting with a little background, it discusses two areas of SDN deployment: clouds and mobile and wireless networks. It then elaborates on the lack of availability of standards on security issues and discusses steps that could be taken to make SDN implementations secure. It ends with a mention of important ongoing research work in this area.

Keywords

SDN, Software Defined Networking, Openflow, Security, Cloud Computing, Traffic Engineering, Controller, Switches, measurements, Information Centric Networking

Table of Contents

1 Introduction

Networks of the twenty first century offer immense flexibility to the business and individual users, but at the cost of higher complexity. Controlling and managing such networks have become highly complex and specialized activities. In this context Software defined networking (SDN) is being looked upon as a promising paradigm that has the power of changing the networking world. Through SDN, network administrators would be able to get a better control over the traffic flows. They would also be able to easily program and modify network policies to manage these flows according to the user requirements. This becomes possible because SDN transfers control and policy functions from a large number of distributed devices to one or more general-purpose servers [Sixto13]. There has been heightened interest in recent years in the academia, industry and the network operators in research and implementation of SDN. It now appears that the time for SDN has finally arrived..

Section 2 gives the implementation issues in SDN and discusses security, interoperability, performance and scalability issues in carrier grade networks. Section 3 takes up the integration of SDN with cloud computing. It looks into the shortcomings of the current design and how SDN can help build better clouds. In Section 4 explores why mobile operators are keen to adopt SDN as they look to make their networks more flexible and scalable to deal with the growing demand for data-intensive applications. The important issue of security in SDN is discussed in Section 5. Some of the important areas of research in SDN include distributed control and information centric networking. We discuss these in Section 6.

1.1 Brief History of SDN

Though the concepts that evolved into SDN have been around for over 20 years [Seszer13], the developments that are more directly attributable to SDN are relatively recent. The development of General Switch Management Protocol (GSMP) by Ipsilon, in 1996, Cambridge's The Tempest in 1998, IETF Forwarding and Control Element Separation (FORCES) in 2000 and IETF Path Computation Element (PCE) in 2004 are important landmarks. PCE is centralized element for computing path for the network nodes. Along with Openflow it is one of the main approaches towards SDN. Also important were the Princeton's Routing Control Platform in 2004 and 4D (decision, dissemination, discovery, and data) approach to separation of control logic from networking elements, in 2005. For many experts, SDN started evolving when the concepts of SDN were first explored in the Ethane project at Stanford University in 2007 [Stanford]. Standardization of Openflow as the first communication interface for SDN by the Open Network Foundation in 2009 was the game changer.

1.2 Initial adoption

The perceived risks associated with SDN have prevented large-scale adoption so far. The fears are not unfounded and need to be allayed by academic guarantee as well as industry commitment. In the past both, the forwarding and the implementation of routing policy, were done at the routers and switches. With SDN, control vests with a server, that is responsible for routing policies, traffic management and security policies. If the single server implementing the control plane became faulty, then the network cannot be provisioned and controlled. SDNs from major vendors do not inter-operate and therefore prevent organizations from having an integrated network wide management system. Notwithstanding these initial hiccups, many believe that SDNâ€™s now ready to take off.

1.3 SDN definition and description

The term SDN has been coined in recent years. It is not implementation specific but is the general term for a framework. Open Network Foundation [ONF] definition of SDN is as follows:

â€œIn the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.â€

The key ideas include separation and centralization of control, open interfaces between controllers and forwarding elements and programmability of the network by external applications. Figure 1 presents these concepts diagramatically. A more elaborate definition would include monitoring, optimizing and managing FCAPS in a multi-tenant environment [Jain13]

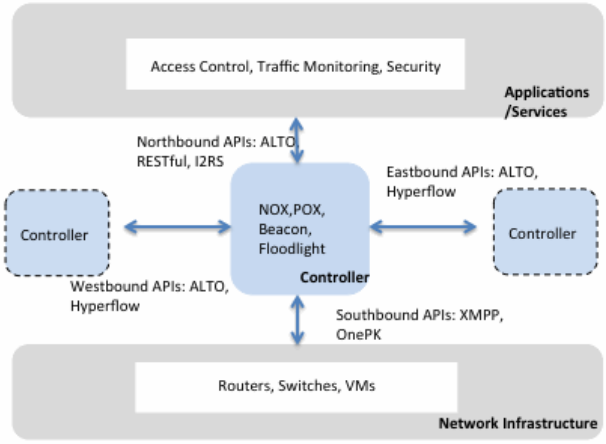

Figure 1. SDN Architecture

1.4 Why SDN?

In traditional networks control and data planes are tightly integrated in the network devices like switches and routers. Once the forwarding policy has been defined, the only way to change it is by changing the configuration of all the affected devices. This is time consuming and puts a limit on scalability and also meeting challenges of mobility and big data. The actual networks have by now become complicated and expensive to maintain. At the same time, revenues are globally declining. Operators are, therefore, looking for solutions that can unify network management and provisioning across multiple domains. SDN has been developed to take care of what is missing in the traditional networks. This has been done by moving control out of the network nodes and keeping it centralized in a server called controller, which has the complete view of the network. This controller can then use the complete knowledge of the network to optimize flow management and support service-user requirements of bandwidth, scalability and flexibility. The separation of the forwarding hardware from the control logic allows easier deployment of new protocols and applications, simplifies network visualization and management. Making applications aware of the network enables greatly improved use of resources and opens up the potential for new applications with the associated potential for revenue generation [Andreas 13].

1.5 Benefits of SDN

The controller is implemented in a server, separate from the switches that forward the traffic. The controller manages the network elements via APIs which make it easier to add new functionality. Advent of big data centers and the clouds, demands on the network administrators to be able to meet the demands for configuration and traffic changes are increasing. The centralized controller of SDN that can exploit the complete knowledge of the network to optimize flow management and support service-user requirements of scalability and flexibility comes to the rescue of the administrator. SDN reduces the capex and opex of the network by implementing system-wide management applications. As the control is software based, organizations can reduce the opex by selecting features that they need. Removing control from all the routers/switches and putting them on a single controller reduces the capex of the network. According to a report by Information Week [Marco â€˜13], some of the important benefits of SDN include ease of provisioning networks, holistic management, better security and efficiency. Despite a number of highpoints, there are some implementation issues that we will discuss in the next section.

Infrastructure as a Service (IaaS) takes the separation even lower and provides the developer with the physical infrastructure needed to provide a service. Thus, the service provider controls the physical resources (networking equipment, connectivity, and physical hardware), while the developer would have control of anything above those resources [NIST800-145]. The provider typically allows the developer to create virtual machines on the physical resources, giving the developer complete control of the system from the choice of operating system to the choice of web environment.

2 Implementation issues in SDN

Network operators and service providers are facing increasing network complexity, increasing cost of deployments, increased variety of service offerings but declining revenues. As already mentioned, they find SDN attractive in terms of network simplification and reduction of the capital and operational expenditure. They are, however, concerned about its success in large carrier grade deployments and are wary of taking risks. We discuss here a number of implementation issues that need to be resolved to make large-scale deployment of SDN a reality.

2.1 Carrier Grade Networks

Network operators have, for long, built their networks with network nodes that have control and data planes integrated. They have institutionalized ways to offer services that can be called carrier grade. The architectural requirements to meet operational expectations in carrier grade networks are scalability, reliability, quality of service (QoS), and service management [Seszer13]. A key challenge in SDN is maintaining carrier grade service within the framework of separation of control and data planes. There have been new developments in the traditional networks, like increasing stress on energy efficiency and reducing the carbon footprint, which SDN and Openflow will have to cope with right from the beginning. In order to have the maximum benefit, the controller would be expected to power up/down parts of the switch on demand. Openflow currently has limited support for the control of power management in the switches and would need extra messages for controlling individual ports. The controller should also be able to query the nodes and find out the available energy efficiency features.

Another feature that is important for Carrier grade networks is recovery from incidents without impacting the users. Enough resilience needs to to be able to recover with 50 ms sub-interval would be required. Building in restoration and protection mechanisms can do this. Having additional pre-defined paths, that are reserved in advance, can incorporate protection. For restoration, the recovery paths can be either preplanned or dynamically allocated, but resources are allocated on failure. Protection is a proactive strategy while restoration is a reactive strategy. In case of SDN, it would be important to have data plane recovery. One way to do this is to build into Openflow, support of a protocol like MPLS-TE, which has its own recovery mechanisms. Alternatively, it can be done by building resilience into Openflow, supporting the recovery of arbitrary flows, regardless of the type of traffic they are carrying [Sharafat11]. Straessens et al have suggested ways to handle both data plane and control plane failures [Straessens11].

2.2 Securing the network

Security is one area on which there has been limited industry and research community discussion. A greater focus on security is, therefore, required if SDN is going to be accepted in broader deployment. At the controller-application level, questions have been raised around authentication and authorization mechanisms, in a multi-tenant setting, that would allow protection of resources of multiple organizations accessing the network [Kern â€˜11]. A security model must be evolved to take care of varying privilege requirements of applications. The controllers are a particularly attractive target for attack in the SDN architecture, open to unauthorized access and exploitation. If the controller is not secured then an attacker may steal its identity, masquerade as one and carry out malicious activities. A security technology such as transport layer security (TLS) with mutual authentication between the controllers and their switches can mitigate these threats. This security feature is optional in Openflow and the standard of TLS is not specified. A full security specification for the controller-switch interface must be defined to secure the connection and protect data transmitted across it. When there are multiple controllers in the network, potential for unauthorized access to nodes and alteration of its configuration and traffic rerouting may take place. It could lead to Denial of Service (DoS), which could have crippling effect on the network. SDNâ€™s strength in open interfaces and known protocols become a boon for attackers. However, the SDN architecture supports a highly reactive security monitoring, analysis, and response system. We shall discuss security aspects in more detail in Section 5.

2.3 Interoperability

As is the case with any large-scale migration, transition to SDN requires coexistence of SDN and the legacy equipment. More developmental work is required to achieve a hybrid SDN infrastructure in which SDN enabled and traditional integrated nodes inter-operate. Such interoperability requires the support of an appropriate protocol that both introduces the requirements for SDN communication interfaces and provides backward compatibility with existing IP routing and multiprotocol label switching (MPLS) control plane technologies. Such a solution would reduce the cost, risk, and disruption for enterprise and carrier networks transitioning to SDN. Different industry groups like IETF, ETSI and ONF are developing standards for SDN. Their work needs to be harmonized for developing the most effective standards to support migration from the traditional network model to SDN.

2.4 Performance

It has to be ensured that the performance of SDN in terms of throughput and latency is commensurate with the traditional networks if not better. These networks should be flexible enough so that new features and capabilities can be programmed. Introduction of new protocols, applications and security features are examples of the changes that might be called for. Use of general-purpose processors, in the controllers, provides high flexibility. High-level programming languages and design tools enable the highest design abstraction and rapid development of complex packet processing functions. The limitation of centralized general-purpose processor implementation, however, is its performance and power dissipation. Looking at the tradeoff it is clear that only a hybrid approach will provide an effective technology solution for SDN. Use of technologies aimed at improving power dissipation, cost and scalability would give the best combination of performance and programmability.

2.5 Scalability

The question of scalability of the controller becomes important in large deployments. Operators fear increased latency with increasing number of network elements. If the number of controllers is increased then the issue of inter-controller communication using east and westbound APIs needs to be resolved. Another related issue is the size and operation of the controller back-end database. A distributed control plane architecture spreads the load. HyperFlow is an example of such a solution for Openflow, that sits on NOX and allows the network operators to have as many controllers as required. All the controllers share the same consistent network-wide view. This concept of providing the network view by distributing the state over multiple controllers is highlighted in [IETF NWG]. To achieve full scalability in SDN, it is being studied whether it would help if the CPU within the network node handles some work locally. A hybrid architecture can reduce the communication between the nodes and the controller and consequently the load on the controller. The backend database size of the controller and its storage requirements are simultaneously reduced [Sezer13].

3 SDN and cloud computing

Cloud computing has emerged as a widely accepted computing paradigm built around concepts such as elimination of up-front investment, reduction of operational expenses, on-demand computing resources, elastic scaling and establishing a pay-per-usage business model for information technology and

computing services. Applications may need to be rewritten or reconfigured before deployment in the cloud to address several network related limitations. It is said that networks create cloud and therefore managing the interplay between networks and clouds is key to efficiency of clouds and in turn their success. SDN is increasingly accepted as the path to design of cloud inter-networks for use of cloud computing on a massive scale [Yeganeh13].

3.1 Cloud inter-network and SDN

TTwo models of SDN have emerged: the "overlay model" and the "network model." In the overlay model, software creates a virtual network. In the network model, network devices create those virtual networks. Overlay SDNs, such as VMware's recently acquired Nicira technology, use software to partition IP or Ethernet addresses into multiple virtual subnetworks. Network APIs allow applications to access these subnetworks as though they were IP or Ethernet networks. The software keeps the traffic of multiple subnetworks secure and isolated. Network-hosted SDNs are built from network devices; therefore, they manage SDN traffic directly. Both the overlay and network models and the SDN clash in the WAN. Overlay SDN depends on software to create virtual networks and getting all the users to have the required software is difficult. The network devices have to be software based so that they can be easily updated and therefore cannot use overlay virtualization. Therefore, the network-hosted SDN becomes kind of unavoidable if the cloud virtual networks are to be extended to the user. Additionally, adequate QoS cannot be assured with an overlay SDN. A network-hosted SDN can manage traffic and ensure QoS.

3.2 Challenges of cloud computing

The prevalent cloud networking architectures reflect the common denominator in meeting different and varied requirements of a cloud. As the many of the important design parameters like the network topology, forwarding protocols, and security policies try to cater to all the requirements, it ends up preventing the optimal usage and proper management of the network. Customers shifting from physical data centers would like to have performance close to it, even though they may be incurring lesser cost. For example, they would expect that the cloud would allow them to specify bandwidth requirements for applications hosted in the cloud as they are otherwise used to [Nolle13]. In many cases the customers sign up for stringent service level agreements (SLAs) according to the requirement of their applications. These need to be met by the cloud resources.

There are some customer requirements that affect a large number of switches and routers. For instance, there are a number of network devices installed, including firewalls, load balancing and application acceleration. Traffic flow related to all of these need to be isolated. Additionally, they would require access control for the end users. Traffic isolation and access control to the end-users are among the multiple forwarding policies that should be enforced. These policies directly impact the configuration of each router and switch. Changing requirements, different protocols, different flavors of L2 spanning tree protocols (STP), along with vendor specific protocols, make it extremely challenging to build, operate and inter-connect a cloud network at scale. Connectivity between the data centers to provide the vision of â€œone cloudâ€ is another challenge.

3.3 Integration of SDN and cloud computing

The demand for virtualization and cloud services has been growing rapidly and attracting considerable interest from industry and academia. The challenges it presents include rapid provisioning, efficient resource management, and scalability that can be addressed using SDNâ€™s control model. A high level description of key building blocks for an SDN-based cloud federation includes: 1) an OpenFlow enabled cloud backbone edge nodes, which connect to the enterprise and cloud provider data center, 2) an OpenFlow enabled core nodes which efficiently switch traffic between these edge nodes, 3) an OpenFlow/SDN-based controller to configure the flow forwarding tables in the cloud backbone nodes and providing a WAN network virtualization application 4) a hybrid cloud operation and orchestration software to manage the enterprise and provider data canter federation, inter-cloud workflow, and resource management of compute/storage and inter-data center network management.

In clouds, offering Infrastructure as a Service (IaaS), users only get a logical view of the underlying network and have limited control. SDN technologies have the capabilities to facilitate delegation of network controls and provide some level of network abstractions to end-users to enable them to configure their slice of the network. Delegating more control to the end-users could raise security concerns for the providers [Azodolmolky13]. SDN-based clouds will allow enterprises to have multi-vendor networks to avoid vendor lock-in. They can access dynamic bandwidth for ad- hoc, timely inter-data center workload migration and processing; and eliminate the burden of underutilized, costly high-capacity fixed private leased lines. SDN-enabled bandwidth-on-demand services provide automated and intelligent service provisioning, driven by cloud service orchestration logic and customer requirements.

3.4 What SDN would bring

SDN provides a new, dynamic network architecture that transforms traditional network backbones into rich service-delivery platforms. By decoupling the network control and data planes, SDN-based architecture abstracts the underlying infrastructure from the applications that utilize it. This makes the networking infrastructure programmable and manageable at scale. SDN adoption can improve network manageability, scalability and dynamism in enterprise data center. The flexibility provided through SDN has allowed its users to efficiently route their flows in networks and conveniently build security applications on top of their networks. SDN-enabled core and edge nodes with a proper SDN controller and network application can be considered as a novel cloud federation mechanism. VLAN, VM-aware networking, vCDNI, VXLAN and Nicira NVP are technologies to provide virtual networks in cloud infrastructures. Nicira NVP, which utilizes MAC in IP encapsulation and external OpenFlow control plane, provides the efficient solution for virtual network implementation.

4. SDN in mobile and wireless networks

Mobile carrier networks follow a well defined architecture with standardized network elements and their interfaces, but developing new features and services in a working network becomes difficult provide. In recent years mobile operators have witnessed rapid growth in over-the-top mobile applications and a large increase in subscriber traffic. Because of its closed nature, groundbreaking network innovations have been few and far between. Carrier networks can benefit from the developments in SDN by incorporating new ways of managing and controlling the network.

4.1 Carrier mobile networks

Carrier networks rely on tightly integrated, proprietary and expensive equipment which is neither easy to configure optimally nor troubleshoot. Carrier equipment follows widely accepted network standards by bodies like ETSI, ITU and Third Generation Partnership Project (3GPP). However, typical implementations rely on vendor-specific hardware platforms. Operators often end up in vendor lock-ins as expansions and upgradations must be covered with equipment from the same vendor for easy inter-operability and flawless operation. Though there are new opportunities, like M2M communication, but standards take years to be firmed up. Enhancements like intelligent networks have been used with limited success in multi-operator networks. Mobile network operators are operating a mix of 2G, 3G and 4G networks. The cost of maintaining and upgrading these are increasing while the revenue is decreasing. Mobile operators are looking at SDN as a path to flexible network with lower cost and ease of giving new services.

4.2 SDN in mobile networks

SDN aims at a shift toward a flow-centric model that employs inexpensive hardware, a logically centralized network controller, and applications that utilize controller-exposed information to orchestrate service delivery in the network. SDN explicitly separates the control and data planes in a manner more versatile than other carrier-grade architectures, such as the 3GPP Evolved Packet Core (EPC). In the existing networks this can only be done on top of IP, as only vendors can modify the highly sophisticated but primarily hardware-based network elements. In SDN, each domain could develop its own network services, addressing specific user needs. SDN implements this through centralized control and programmable switches. SDN is therefore being taken up in important standardization bodies like ITU and ETSI. According to a survey report by research firm Telecoms and Media [Burt13], mobile carriers will be among the leading adopters of SDN technology as they look to make their networks more flexible and scalable to deal with the growing demand for data-intensive applications. Ninety-three percent of mobile carriers expect to implement SDN initiatives in their businesses within the next five years [Kempf12, Costanzo12].

4.3 Expectations from SDN in mobile networks

Mobile operators are interested in mature limited-risk technologies that optimize use of the scarce and expensive wireless resources. As the number of customers and traffic increase, the network should be able to handle billions of users and onslaught of traffic from a variety of mobile devices. They are also looking at SDN to increase the flexibility of introducing new service bundles faster. A recent position paper on software-defined cellular networks [Malik13] suggests that SDN can simplify cellular networks and lower management costs. A top-level requirement for application of SDN in mobile networks is to provide maximum flexibility, openness, and programmability to future carriers without mandating any changes in user equipment. In this way, operators can innovate inside their domain without having to depend on either over-the-top (OTT) service providers or UE vendors to support their innovations. The SDN open interfaces and APIs foster service innovation, increasing the capability of the operator to roll out new network features while reducing time to market for new services [Pentikousis13].

4.4 SDN for Cellular Data Networks

Cellular data networks are ripe for the introduction of SDN, where the network equipment performs basic packet-processing functions at the behest of applications running on a logically centralized controller. SDN could give cellular operators greater control over their equipment, simplify network management, and introduce value-added services. SDN can enable carriers to distribute data-plane rules over multiple less expensive network switches, reducing the scalability demand on the centralized controller and enabling flexible handling of traffic. Mobility puts extra pressure on the network as the changes in the state of individual subscribers need to be forwarded to avoid disconnections. Traffic flows need to be kept track of to make sure that the subscribers stay within their usage caps. In addition, to be able to adapt to QoS policies, the network must adapt quickly to measurement data. The controller should be able to convert policies based on subscriber requirements to the rules based on which the switches will forward the traffic flow. Switches should perform some local processing to reduce the load on the controller. Cellular network may require message control protocols, deep-packet inspection or header compression. Virtualization of the base station can give different traffic classes the illusion of a dedicated base station.

4.5 Software defined wireless networks

An increasing number of enterprises working in the field of wireless communications are joining the SDN initiatives. Providers of such networks as WiMax, Wifi mesh, â€¦ believe that features provided by Software Defined Wireless Networks (SDWN) introduce evolvability in their networks, which would allow them to differentiate their service offerings from competitors. SDN promises to drastically reduce the complexity of network configuration and management. They, therefore, expect to improve network efficiency given that new, more efficient technical solutions can be easily deployed in existing equipment. In the wireless infrastructureless networking environments the separation of the network control and management functionality from the forwarding operations offers even greater possibilities. SDN resolves the problem of migrating nodes among networks by defining functionalities at the higher layer of the protocol stack through software and thus making them easily modifiable. One of the requirements from SDWN would be support for duty cycles for reducing energy consumption. SDWN must also support in-network aggregation of correlated data for conserving energy. Use of OpenFlow allows SDWN to support flexible definition of rules that consider the traditional TCP/IP header fields only. This allows a definition of switching and routing strategies which are much more flexible when compared to traditional switching/routing strategies.

5 SDN and Security

Security has been difficult to implement even in the traditional networks because of difficulty in enforcing the required policies in a continually changing environment. SDN provides new ways of dealing with this problem by enabling introduction of sophisticated network policies. An example is the Ethane project of Stanford University that describes an SDN architecture that allows managers to enforce fine-grained access control policies. In the remaining part of this section we will discuss the features that are available and the ones that need to be included in the SDN framework to make it secure and dependable.

5.1 Comparison of traditional network and SDN

Traditional networks were largely built through proprietary, integrated and closed solutions. The closed nature of network devices, their fairly static design, the heterogeneity of software, and the decentralized nature of the control plane provided a natural defense against the common threats. The source code of the software was a closely held secret of the OEMs and its nuances were a secret well hidden from malicious attackers. SDN systems are built on programmable architecture with much of details available in the public domain. The centralized programmable control plane architecture introduces new attack points and opens the doors for new threats that did not exist before or were harder to exploit. A common open standard like OpenFlow also increases the risk of common faults permeating the control and data plane equipment.

5.2 Security and dependability in SDN

According to [Sorensen13] security and dependability issues are serious concerns for the industry. They argue that security should be an inherent part of the design of the future SDN and not introduced later as an appendage. The two basic aspects of SDN â€“ software based control and centralization of intelligence that give it greater flexibility can become attractive targets for malicious users. Hackers can gain access to software-based controllers and control the entire network. On the contrary, if the SDN is properly designed and deployed, the resulting network environment will be more rugged and resilient. The security and dependability incorporated so far in the SDN controllers are limited to authenticated communication channels and replication of data among controller instances. There is also no technique implemented to assure data integrity and confidentiality in or between controllers [Kerner13]. The threat profiles that need to be taken care of are described in the next sub-section.

5.3 Threat profiles

The serious threats in the SDN environment can be profiled into different categories and work on finding solution for them can be pursued accordingly [Kreut13].

1. Spurious Traffic flows: These can affect switches and controllers alike. An attacker, with malicious intent, can build up the infructuous flows to such an extent as to seriously overload the switches and the controller and create a DoS attack. A possible direction that could be pursued is the use of intrusion detection systems with support for runtime root-cause analysis to identify abnormal flows.
2. Attacks on vulnerabilities in switches: One single switch could be used to drop or slow down packets in the network, clone or deviate network traffic for data theft or with the intension of loading. A possible way to address this is by use of mechanisms of software attestation, such as autonomic trust management solutions for software components.
3. Attacks on control plane communications: This can be used to generate DoS attacks or for data theft. The TLS/SSL model is not enough. One solution that can be examined is securing communication with threshold cryptography across controller replicas.
4. Attacks on and vulnerabilities in controllers: In a very serious threat, a faulty or malicious controller could compromise an entire network. The use of a common intrusion detection system may not be enough. Replication, diversity and recovery can be used for mitigation.
5. Attacks on and vulnerabilities in administrative stations: As in traditional networks, these can be used in SDNs to access the network controller. The use of protocols requiring double credential verification may prove to be helpful.
6. Lack of trusted resources for forensics and remediation: This could prevent identification of the cause of a detected problem and proceed to a fast and secure mode recovery. In order to investigate and establish facts about an incident, we need reliable information from all components and domains of the network. Logging and tracing are the common mechanisms in use, but they should be available for both data and control planes and should be guaranteed to be immutable and secure.

5.4 Secure control platform

Controller is the nerve center of the network. If the nerve center is compromised, all the communication can go awry. Some of the techniques that can be applied to secure the control platform in SDN are as follows:

1. Replication: Devices and applications should be replicated. This improves the dependability of the system. The mixed approach ensures tolerance to both hardware and software faults, accidental or malicious.
2. Diversity: The same management application can be run on different controllers. Diversity improves robustness and avoids common-mode faults.
3. Self-healing mechanisms: the system can be kept working by replacing the compromised component through an auto healing process.
4. Dynamic device association: A single controller becomes single point of failure. Once the controller fails, the control operation of the switch fails and the switch will need to associate with another controller. Dynamic association with another controller would provide enhanced security.

Security and dependability of SDN still is a field almost unexplored, presenting many challenges and opportunities. Continuous effort would be required as SDN evolves to be one step ahead of the malicious attackers.

6. Future Works

As has been mentioned above, a number of issues relating to SDN are still under active research in academia and industry. While it scores on flexibility and versatility, SDN raises significant scalability, performance, robustness, and security challenges. Setting the tone of the research, one of the projects at Stanford University, FLARE [Diego13] describes a new network node model focusing on â€œdeeply programmable networksâ€ that provides programmability for the data plane, the control plane, as well as the interface between them. We shall focus here on some of the important research efforts focusing on the issues relating to the controller(s), forwarding elements and their interaction among themselves and the applications though protocols and APIs [Mendonca12].

6.1 Distributed controller design

An important research area is the scalability of the controller design. A recent study shows that one single controller can handle up to 6 million flows/s. It is however being increasingly recognized that for scalability and reliability, control should be physically distributed, even though it may be logically centralized. Onix, Kando, and HyperFlow use this approach to achieve robust and scalable control plane. The authors in [Nakao12] discuss important aspects in controller design including hierarchical control, data model, scalability, and extensibility. Vesting some control with the switches, so that they are able to do some tasks locally to relieve the centralized controller, is also an important area of study. In the scalable flow-based networking with DIFANE, as proposed by researchers at Princeton and AT&T Labs, flow entries are proactively pushed to switches in an attempt to reduce the number of requests to the controller. Similar concept has been proposed by A Curtis el al. in Devoflow [Curtis11] where â€œshort-livedâ€ flows are handled in switches and â€œlong-livedâ€ flows in the controller to mitigate flow setup delay and controller overhead.

As fallout of the logically centralized control model of SDN, much of the current work has focused on solutions that have single administrative domain. In use cases where the administration is decentralized e.g. the Internet, centralized control plane may not be the suitable option. Control plane would need to be logically distributed. If this becomes possible to implement effectively then the various autonomous system would have their own independent logically centralized and physically distributed controller. According to [Mendonca12] the idea of a Software-Defined Internet has not been explored. They propose a software-defined Internet architecture that borrows from MPLS, the distinction between network edge and core to split tasks between inter-domain and intra-domain components. Another approach to inter-AS routing uses NOX and OpenFlow to implement BGP-like functionality.

6.2 Definition of controller interfaces

A very important aspect of SDN is availability of APIs in various directions for effective operation and control. While southbound APIs for interaction between controllers and switches have been elaborately defined by ONF, the northbound APIs still lack adequate standards. If we think of the controller as a â€œnetwork operating systemâ€, then there should be a clearly defined interface by which applications can access the underlying hardware, co-exist and interact with other applications, and utilize system services, without requiring the application developer to know the implementation details of the controller. Several controllers have been developed, however, their application interfaces are still in the early stages and their interactions not defined. To avoid ad-hoc development of network applications and promote applications that are flexible and portable, needs clear definition of northbound interfaces.

6.3 Dynamic and customized measurements

Another area that would need attention is measurement. Measurement in real-time is a necessity to be able to have proper control over the network. Although control and measurement are two important components of network management, attention has so far gone to developing APIs for control with measurement being the neglected cousin. It is important for measurement techniques to be non-intrusive and have low cost in terms of resources required. The SDN environment should provide customized and application specific dynamic measurement and accurate data collection based on the users requirements [Yu13].

6.4 Information Centric Networking and SDN

An active area of research in SDN is Information-Centric Networking (ICN). It aims to improve the efficiency of the future Internet in content delivery and content availability. There have been some architectural proposals that further this concept e.g., Content- Centric Networking (CCN), also known as the Named Data Networking (NDN). The separation between information processing and forwarding in ICN is aligned with the decoupling of the data plane and control plane in SDN. A number of projects that have dealt with combining ICN with SDN towards â€œSoftware-Defined Information-Centric Networksâ€ have proposed using SDN concepts to realize ICNs [Raghavan12, Veltri12, Blefari12].

7. Summary

SDN is a framework that increases the flexibility of the network through separation of control and data planes. This separation makes the network switching and routing devices simpler and less expensive. The control plane could be implemented in a general purpose commodity server. This server as a centralized controller takes care of routing and other policies according to which the network devices function.

SDN has many application in enterprise and carrier environments. In the carrier deployments it promises to obliterate the problem arising out of proprietary, closed and rigid networks. SDN reduces the capex and opex of the operators and provides flexibility of introducing new service easily. This paper discusses three important areas where SDN can be of great benefit: Cloud computing, cellular mobile networks and Information centric networking.

While SDN has been around for a few years, the initial uptake has been restricted. It is believed that though the operators like the benefits of SDN, they are wary of the risks the new technology carries. The most important of these are security, interoperability, performance and scalability. This paper discusses all these issues. Security arguably being one of the most important but neglected concern, the discussions are aimed at suggesting ways to characterize the threat profile and finding solutions for each of these.

Covering all the aspects of an elaborate framework like SDN in one single paper is a daunting task. This paper aims to provide an overview of the common issues of SDN implementation and how recent research work aims to mitigate these. Current research work and future directions have been discussed specially in the context of carrier networks including the Internet.

References

1. [Sixto13] Sixto Ortiz Jr., â€œSoftware defined networking-On the verge of breakthrough?â€ IEEE Computer Society, 2013 http://ieeexplore.ieee.org.libproxy.wustl.edu/stamp/stamp.jsp?tp=&arnumber=6576757

2. [Sezer13] Sezer, S. et al, â€œAre We Ready For SDN? Implementation Challenges for Software-Defined Networks,â€ IEEE Communications Magazine, Volume: 51, Issue: 7, 2013 , Page(s): 36- 43 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6553676
3. [Jain13] Raj Jain, Lecture notes on Introduction to Software Defined Networking (SDN), http://www.cse.wustl.edu/~jain/cse570-13/
4. [Stanford] Ethane: A Security Management Architecture, http://yuba.stanford.edu/ethane
5. [ONF] Open Network Foundation, https://www.opennetworking.org/
6. [Andreas â€˜13] Andreas Antonopoulos, â€œWhy Software Defined Networking is Becoming a Reality,â€ http://searchsdn.techtarget.com/tip/Why-software-defined-networking-is-becoming-a-reality
7. [Marco13] Kurt Marko, â€œSDN-Business Benefits,â€ Feb 2013 http://reports.informationweek.com/ abstract/19/ 10517/ Network-Infrastructure/5-SDN-Business-Benefits.html"
8. [Sharafat11] Sharafat, A. R. et al, â€œMPLS-TE and MPLS VPNs with Openflow,â€ SIGCOMM 2011 http://yuba.stanford.edu/~nickm/papers/mpls-sigcomm11.pdf
9. [Staessens11] Staessens, D.; Sharma, S.; Colle, D.; Pickavet, M.; Demeester, P, â€œSoftware defined networking: Meeting carrier grade requirementsâ€, 18th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN), Oct. 2011http://www.fp7-sparc.eu/assets/publications/12-SDN-LANMAN-2011.pdf
10. [Kern11] Kern, A. et al, â€œMPLS-Openflow based access/aggregation network,â€ iPOP, 2011http://www.fp7-sparc.eu/assets/publications/05-SPARC_MPLS_OpenFlow_Poster-A.pdf(poster)
11. [IETF NWG] IETF Network WG â€œSecurity Requirements in the Software Defined Networking Model,â€ Internet draft, https://datatracker.ietf.org/doc/draft-hartman-sdnsec-requirements/
12. [Yeganeh13] Yeganeh, S.; Tootoonchian, A.; Ganjali, Y., â€œOn Scalability of Software-Defined Networking,â€ IEEE Communications Magazine, Volume: 51, No. 2, February 2013, Page(s): 136â€“141 http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6461198&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F35%2F6461169%2F06461198.pdf%3Farnumber%3D6461198
13. [Nolle13] Tom Nolle, â€œThe role of software-defined networks in cloud computing" http://searchcloudcomputing.techtarget.com
14. [Azodolmolky13] Siamak Azodolmolky, Philipp Wieder, Ramin Yahyapour, â€œSDN-Based Cloud Computing Networkingâ€ 15th International Conference on Transparent Optical Networks (ICTON), 2013,Page(s): 1- 4http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6602678&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel7%2F6589037%2F6602671%2F06602678.pdf%3Farnumber%3D6602678
15. [Malik13] Malik, M.S. et al, â€œTowards SDN enabled network control delegation in clouds, (DSN),â€ 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2013http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6575320&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6575320
16. [Kempf12] Kempf, J. et al, â€œMoving the Mobile Evolved Packet Core to the Cloud,â€ Proceedings WiMob, Barcelona, Spain, 2012 http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=6379165
17. [Pentikousis13] Kostas Pentikousis et al., "MobileFlow: Toward Software-Defined Mobile Networks,â€ IEEE Communications Magazine, July 2013, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6553677
18. [Costanzo12] Salvatore Costanzo; Laura Galluccio; Giacomo Morabito; Sergio Palazzo, â€œSoftware Defined Wireless Networks: Unbridling SDNs,â€ European Workshop on Software Defined Networking, 2012, http://www.diit.unict.it/users/gmorabi/sdwnWebSite/sdwn.pdf
19. [Burt13] Jeffrey Burt, â€œSDN Adoption to Grow Rapidly Among Carriers,â€ Informa, 2013 http://www.eweek.com/networking/sdn-adoption-to-grow-rapidly-among-carriers-informa/#sthash.K7GxehLC.dpuf
20. [Kreut13] Diego Kreutz; Fernando M.V. Ramos; Paulo Verissimo, â€œTowards Secure and Dependable Software-Defined Networks,â€ Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, HotSDN '13, 2013 Pages 55-60 http://www.ietf.org/proceedings/87/slides/slides-87-sdnrg-2.pdf
21. [Sorensen13] Sorensen, S., â€œSecurity Implications of Software-Defined Networks,â€ 2012, http://goo.gl/BiXH2
22. [Kerner13] Kerner, S. M., â€œIs SDN Secure?,â€ 2013, http://goo.gl/lPn2V
23. [Diego13] Diego Kreutz et al., â€œTowards, Secure and Dependable Software-Defined Networks,â€ HotSDN 2013, ACM 2013 http://www.ietf.org/proceedings/87/slides/slides-87-sdnrg-2.pdf
24. [Mendonca12] Marc Mendonca et al., â€œA Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks,â€ (in submission in June 2013)http://hal.inria.fr/docs/00/82/50/87/PDF/SDN_survey.pdf
25. [Nakao12] Nakao, Flare, A., â€œOpen Deeply Programmable Network Node Architecture,â€ http://netseminar.stanford.edu/seminars/10_18_12.pdf
26. [Curtis11] Curtis, A.; Mogul, J.; Tourrilhes, J.; Yalagandula, P.; Sharma, P.; Banerjee, S., â€œDevoflow: Scaling flow management for high-performance networks,â€ ACM SIGCOMM, 2011 http://www.cmlab.csie.ntu.edu.tw/~kenneth/qing2011/paper/6.pdf
27. [Yu13] Minlan Yu, Lavanya Jose, Rui Miao, â€œSoftware Defined Traffic Measurement with OpenSketch,â€ Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, NSDI, 2013, Pages 29-42 http://www-bcf.usc.edu/~minlanyu/writeup/nsdi13-opensketch.pdf
28. [Raghavan12] Raghavan, B.; Casado, M.; Koponen, T.; Ratnasamy, S.; Ghodsi, A.; Shenker, S., â€œSoftware-Defined Internet Architecture: Decoupling Architecture from Infrastructure,â€ Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, 2012, Page(s): 43â€“48 http://www1.icsi.berkeley.edu/~barath/papers/sdia-hotnets12.pdf
29. [Veltri12] Veltri, L.; Morabito, G.; Salsano, S.; Blefari-Melazzi, N.; Detti, A., â€œSupporting Information-Centric Functionality in Software Defined Networks,â€ IEEE ICC Workshop on Software Defined Networks, June 2012, http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6364916&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6364916
30. [Blefari12] Blefari-Melazzi, N.; Detti, A.; Mazza, G.; Morabito,; Salsano, S.; Veltri, L., â€œAn Openflow-Based Testbed for Information Centric Networking,â€ Future Network & Mobile Summit, 2012, Page(s) 4â€“6, http://www.techrepublic.com/resource-library/whitepapers/an-openflow-based-testbed-for-information-centric-networking/

List of Acronyms

3GPP 3rd Generation Partnership Project

APIs	Application Programming Interface
BGP	Border Gateway Protocol
DWDM	Dense Wavelength Division Multiplexing
EPC	Enhanced Packet Core
ETSI	European Telecommunications Standard Institute
FORCES	Forwarding and Control Element Separation
GSMP	General Switch Management Protocol
IaaS	Infrastructure as a Service
ICN	Information Centric Networks
IEEE	Institution of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunications Union
M2M	Machine to Machine
MAC	Media Access Control
MPLS	Multi Protocol Label Switching
MPLS-TE	Multi-Protocol Label Switching-Traffic Engineering
ONF	Open Network Foundation
OTT	Over The Top
PCE	Pat Computation Element
PON	Passive Optical Network
QoS	Quality of Service
SCTP	Stream Control Transmission Protocol
SDN	Software Defined Networking
SDWN	Software Defined Wireless Network
SLA	Service Level Agreement
STP	Spanning Tree Protocol
TLS/SSL	Transport Layer Security/Secure Socket Layer
UE	User Equipment
vCDNI	vCloud Director Network Isolation
VLAN	Virtual Local Area Network
VM	Virtual Machine
VXLAN	Virtual Extensible Local Area Network
WAN	Wide Area Network

Last Modified: December 10, 2013
This and other papers on latest advances in computer networking are available on line at http://www.cse.wustl.edu/~jain/cse570-13/index.html
Back to Raj Jain's Home Page