International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 11
November 2017

# A Survey of Software-Defined Networking

**Varshkumar Patel[1], Drashti Vashi[2]**
[1,2](U. and P. U. Patel Department of Computer engineering, CSPIT, CHARUSAT, Changa , Gujarat, India.

***Abstract****—The possibility of programmable systems has as of late recovered significant force because of the development of the Software-Defined Networking (SDN) worldview. SDN, regularly alluded to as a "radical new thought in systems administration", guarantees to significantly improve organize administration and empower in-novation through system programmability. This paper studies the cutting edge in programmable systems with an accentuation on SDN. We give a noteworthy point of view of programmable systems from early plans to late advancements. At that point we display the SDN engineering and the OpenFlow standard specifically, talk about flow choices for execution and testing of SDN-based conventions and administrations, look at ebb and flow and future SDN applications, and investigate promising exploration bearings in view of the SDN worldview.*

***Index Terms—Software-Defined Networking, programmable networks, survey, data plane, control plane, virtualization.***

## I. INTRODUCTION

Computer systems are commonly worked from countless gadgets, for example, switches, switches and various sorts of middleboxes (i.e., gadgets that control movement for purposes other than parcel sending, for example, a firewall) with numerous perplexing conventions executed on them. System administrators are in charge of arranging arrangements to react to an extensive variety of system occasions and applications. They need to physically change these abnormal state strategies into low-level design summons while adjusting to changing system conditions. What's more, regularly they have to achieve these exceptionally complex errands with access to extremely constrained instruments. Subsequently, organize administration and execution tuning is very testing and in this manner blunder inclined. The way that system gadgets are normally vertically-coordinated secret elements fuels the test arrange administrators and chairmen confront.

Another practically unsurmountable test arrange professionals and specialists confront has been alluded to as "Web solidification". In view of its colossal organization base and the reality it is considered piece of our general public's basic foundation (quite recently like transportation and power matrices), the Internet has turned out to be to a great degree hard to develop both as far as its physical framework and additionally its conventions and execution. In any case, as present and developing Internet applications and administrations turn out to be progressively more unpredictable and requesting, it is basic that the Internet have the capacity to advance to address these new difficulties.

The possibility of "programmable systems" has been proposed as an approach to encourage arrange advancement. Specifically, Software Defined Networking (SDN) is another systems administration worldview in which the sending equipment is decoupled from control choices. It guarantees to drastically disentangle arrange administration and empower advancement and development. The primary thought is to enable programming designers to depend on arrange assets in an indistinguishable simple way from they do on capacity and processing assets. In SDN, the system knowledge is coherently brought together in programming based controllers (the control plane), and system gadgets end up plainly basic bundle sending gadgets (the information plane) that can be modified by means of an open interface (e.g., ForCES [1], OpenFlow [2], and so forth.).

SDN is presently drawing in huge consideration from both scholarly world and industry. A gathering of system administrators, specialist co-ops, and merchants have as of late made the Open Network Foundation [3], a modern driven association, to advance SDN and institutionalize the OpenFlow convention [2]. On the scholarly side, the OpenFlow Network Research Center [4] has been made with an emphasis on SDN inquire about. There have additionally been institutionalization endeavours on SDN at the IETF and IRTF and different models delivering associations.

The field of programming characterized organizing is very later, yet developing at a quick pace. In any case, there are critical research difficulties to be tended to. In this paper, we overview the cutting edge in programmable systems by giving a notable point of view of the field and furthermore depicting in detail the SDN worldview and engineering. The paper is sorted out as takes after: in Section II, it starts by depicting early endeavours concentrating on programmable systems. Area III gives a review of SDN and its engineering. It likewise portrays the OpenFlow convention. Segment IV portrays existing stages for creating and testing SDN arrangements including copying and recreation devices, SDN controller usage, and in addition check and troubleshooting apparatuses. In Section V, we talk about a few SDN applications in zones, for example, server farms and remote systems administration. At long last, Section VI talks about research difficulties and future bearings.

## II. EARLY PROGRAMMABLE NETWORKS

SDN can possibly change the way organizes work, and OpenFlow specifically has been touted as a "radical new thought in systems administration" [5]. The proposed benefits extend from concentrated control, disentangled calculations, commoditizing system equipment, taking out middleboxes, to empowering the outline and sending of outsider 'applications'.

While OpenFlow has gotten impressive consideration from industry, it is important that the possibility of programmable systems and decoupled control rationale has been around for a long time. In this segment, we give an outline of early programmable systems administration endeavours, antecedents to the current SDN worldview that established the framework for a considerable lot of the thoughts we are seeing today.

**a) Open Signalling:**The Open Signalling (OPENSIG) working gathering started in 1995 with a progression of workshops devoted to "making ATM, Internet and portable systems more open, extensible, and programmable" [6]. They trusted that a division between the correspondence equipment and control programming was important however difficult to understand; this is essentially due to vertically coordinated switches and switches, whose shut nature made the quick arrangement of new system administrations and conditions unthinkable. The centre of their proposition was to give access to the system equipment by means of open, programmable system interfaces; this would permit the organization of new administrations through an appropriated programming condition.

Roused by these thoughts, an IETF working gathering was made, which prompted the particular of the General Switch Management Protocol (GSMP) [7], a universally useful convention to control a name switch. GSMP enables a controller to build up and discharge associations over the switch, include and erase leaves a multicast association, oversee switch ports, ask for setup data, ask for and erase reservation of switch assets, and demand measurements. The working gathering is authoritatively finished up and the most recent benchmarks proposition, GSMPv3, was distributed in June 2002.

**b) Active Networking:**Additionally in the mid-1990s, the Active Networking [8], [9] activity proposed the possibility of a system framework that would be programmable for tweaked administrations. There were two principle approaches being considered, in particular: (1) client programmable switches, with in-band information exchange and out-of-band administration channels; and (2) containers, which were program sections that could be conveyed in client messages; program pieces would then be deciphered and executed by switches. Notwithstanding significant action it spurred, Active Networking never assembled minimum amount and exchanged to broad utilize and industry sending, primarily because of common sense security and execution concerns [10].

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 11
November 2017

**c) DCAN:**Another activity that occurred in the mid-1990s is the Devolved Control of ATM Networks (DCAN) [11]. The point of this venture was to outline and build up the important foundation for adaptable control and administration of ATM systems. The start is that control and administration elements of the numerous gadgets (ATM switches on account of DCAN) ought to be decoupled from the gadgets themselves and assigned to outside elements committed to that reason, which is essentially the idea driving SDNs. DCAN accept a moderate convention between the supervisor and the system, in the lines of what happens today in recommendations, for example, OpenFlow. More on the DCAN task can be found at [12]. Still in the lines of SDNs and the proposed decoupling of control and information plane over ATM systems, among others, in the work proposed in [13] various heterogeneous control models are permitted to run at the same time finished single physical ATM arrange by dividing the assets of that switch between those controllers.

**d) 4D Project:**Beginning in 2004, the 4D venture [14],[15], [16] pushed a fresh start outline that accentuated partition between the directing choice rationale and the conventions administering the connection between arrange components. It proposed giving the "choice" plane a worldwide perspective of the system, adjusted by a "spread" and "revelation" plane, for control of an "information" plane for sending activity. These thoughts gave guide motivation to later works, for example, NOX [17], which proposed a "working framework for systems" with regards to an OpenFlow-empowered system.

**e) NETCONF:**In 2006, the IETF Network Configuration Working Group proposed NETCONF [18] as an administration convention for altering the design of system gadgets. The convention permitted arrange gadgets to uncover an API through which extensible design information could be sent and recovered.

Another administration convention, generally conveyed in the past and utilized until today, is the SNMP [19]. SNMP was proposed in the late 80's and turned out to be an exceptionally prevalent system administration convention, which utilizes the Structured Management Interface (SMI) to get information contained in the Management Information Base (MIB). It could be utilized also to change factors in the MIB keeping in mind the end goal to alter design settings. It later ended up plainly clear that regardless of what it was initially expected for, SNMP was not being utilized to design arrange gear, but instead as an execution and blame checking instrument. Additionally, different weaknesses were distinguished in the origination of SNMP, the most outstanding of which was its absence of solid security. This was tended to in a later form of the convention.

NETCONF, at the time it was proposed by IETF, was seen by numerous as another approach for arrange administration that would settle the previously mentioned inadequacies in SNMP. In spite of the fact that the NETCONF convention fulfils the objective of improving gadget (re)configuration and goes about as a building obstruct for administration, there is no detachment amongst information and control planes. The same can be expressed about SNMP. A system with NETCONF ought not be viewed as completely programmable as any new usefulness would need to be executed at both the system gadget and the supervisor so any new usefulness can be given; besides, it is outlined basically to help mechanized design and not for empowering direct control of state nor empowering brisk arrangement of inventive administrations and applications. All things considered, both NETCONF and SNMP are valuable administration devices that might be utilized as a part of parallel on half and half switches supporting different arrangements that empower programmable systems administration.

The NETCONF working gathering is at present dynamic and the most recent proposed standard was distributed in June 2011.

**f) Ethane:**The prompt forerunner to OpenFlow was the SANE/Ethane venture [20], which, in 2006, characterized another design for big business systems. Ethane's attention was on utilizing a brought together controller to oversee approach and security in a system. A prominent illustration is giving personality based access control. Like SDN, Ethane utilized two parts: a controller to choose if a bundle ought to be sent, and an Ethane switch comprising of a stream table and a protected channel to the controller.

Ethane established the framework for what might move toward becoming Software-Defined Networking. To place Ethane with regards to the present SDN worldview, Ethane's personality based access control would

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 11
November 2017

likely be actualized as an application over a SDN controller, for example, NOX [17], Maestro [21], Beacon [22], SNAC [23], Helios [24], and so on.

## III. SOFTWARE-DEFINED NETWORKINGARCHITECTURE

Information correspondence arranges normally comprise of end-client gadgets, or hosts interconnected by the system foundation. This foundation is shared by hats and utilizes exchanging components, for example, switches and switches and in addition correspondence connects to convey information between has. Switches and switches are normally "shut" frameworks, frequently with restricted and seller particular control interfaces. Thusly, once conveyed and underway, it is very troublesome for current system framework to develop; as it were, sending new forms of existing conventions (e.g., IPv6), also sending totally new conventions and administrations is a practically unconquerable obstruction in current systems. The Internet, being a system of systems, is no special case.

Programming Defined Networking was produced to encourage development and empower basic automatic control of the system information way. As envisioned in Figure 1, the partition of the sending equipment from the control rationale permits simpler arrangement of new conventions and applications, clear system perception and administration, and solidification of different middleboxes into programming control. Rather than upholding approaches and running conventions on a convolution of scattered gadgets, the system is decreased to "basic" sending equipment and the basic leadership organize controller(s).

### A. Current SDN Architectures

In this segment, we audit two surely understood SDN designs, to be specific ForCES [1] and OpenFlow [2]. Both OpenFlow and ForCES take after the fundamental SDN guideline of division between the control and information planes; and both institutionalize data trade between planes. Be that as it may, they are in fact altogether different as far as plan, engineering, sending model, and convention interface.

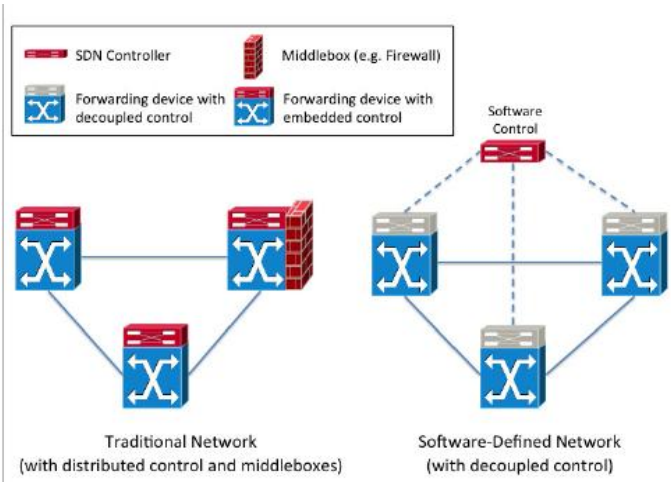


*Fig. 1. The SDN architecture decouples control logic from the forwarding hardware, and enables the consolidation of middleboxes, simpler policy management, and new functionalities. The solid lines define the data-plane links and the dashed lines the control-plane links.*

**1) ForCES:**The approach proposed by the IETF ForCES (Forwarding and Control Element Separation) Working Group, rethinks the system gadget's inward design having the control component isolated from the sending component. Be that as it may, the system gadget is as yet spoken to as a solitary element. The driving use case gave by the working gathering considers the want to join new sending equipment with outsider control inside a solitary system gadget. Hence, the control and information planes are kept inside nearness (e.g., same box or room). Conversely, the control plane is tore totally from the system gadget in "OpenFlow-like" SDN frameworks.

Powers characterizes two rationale elements called the Forwarding Element (FE) and the Control Element (CE), both of which execute the ForCES convention to impart. The FE is in charge of utilizing the hidden equipment to give per-parcel taking care of. The CE executes control and flagging capacities and utilizes the ForCES convention to train FEs on the best way to deal with bundles. The convention works in view of an ace slave demonstrate, where FEs are slaves and CEs are experts.

A critical building piece of the ForCES design is the LFB (Logical Function Block). The LFB is a very much characterized useful square dwelling on the FEs that is controlled by CEs by means of the ForCES convention. The LFB empowers the CEs to control the FEs' design and how FEs process parcels.

Powers has been experiencing institutionalization since 2003, and the working gathering has distributed an assortment of records including: a relevance proclamation, a design outline work characterizing the elements and their connections, a displaying dialect characterizing the legitimate capacities inside a sending component, and the convention for correspondence between the control and sending components inside a system component. The working gathering is at present dynamic.

**2) Discussion**: In [26], the similitudes and contrasts amongst ForCES and OpenFlow are examined. Among the distinctions, they feature the way that the sending model utilized by ForCES depends on the Logical Function Blocks (LFBs), while OpenFlow utilizes stream tables. They call attention to that in OpenFlow activities related with a stream can be joined to give more noteworthy control and adaptability to the reasons for organize administration, organization, and improvement. In ForCES the blend of various LFBs can likewise be utilized to accomplish a similar objective.

We ought to likewise re-repeat that ForCES does not take after the same SDN demonstrate supporting OpenFlow, but rather can be utilized to accomplish similar objectives and execute comparative usefulness [26].

The solid help from industry, research, and the scholarly community that the Open Networking Foundation (ONF) and its SDN proposition, OpenFlow, has possessed the capacity to accumulate is very noteworthy. The subsequent minimum amount from these diverse segments has created countless as research papers, reference programming usage, and even equipment. To such an extent that some contend that OpenFlow's SDN engineering is the current SDN true standard. In accordance with this pattern, the rest of this area concentrates on OpenFlow's SDN show. All the more particularly, we will depict the diverse segments of the SDN design, in particular: the switch, the controller, and the interfaces exhibit on the controller for correspondence with sending gadgets (south-bound correspondence) and system applications (northbound correspondence). Area IV additionally has an OpenFlow centre as it depicts existing stages for SDN advancement and testing, including imitating and recreation devices, SDN controller executions, and check and investigating devices. Our dialog of future SDN applications and research bearings is broader and is SDN engineering freethinker.

### B. Southbound Communication: Controller-Switch

An imperative part of SDNs is the connection between the information plane and the control-plane. As sending components are controlled by an open interface, it is imperative that this connection stays accessible and secure.

The OpenFlow convention can be seen as one conceivable usage of controller-switch cooperation, as it characterizes the correspondence between the exchanging equipment and a system controller. For security, OpenFlow 1.3.0 gives discretionary help to scrambled Transport Layer Security (TLS) correspondence and a declaration trade between the switches and the controller(s); in any case, the correct execution and testament organize isn't at present determined. Additionally outside the extent of the present determination are fine-grained security alternatives in regards to situations with different controllers, as there is no technique indicated to just give halfway access consents to an approved controller.

## IV. SDN APPLICATIONS

Programming characterized organizing has applications in a wide assortment of arranged conditions. By decoupling the control– and information planes, programmable systems empower modified control, a chance

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 11
November 2017

to take out middleboxes, and also streamlined advancement and organization of new system administrations and conventions. Underneath, we look at changed conditions for which SDN arrangements have been proposed or executed.

**A. Enterprise Networks**

Undertakings frequently run vast systems, while likewise having strict security and execution necessities. Moreover, unique endeavour conditions can have altogether different necessities, qualities, and client populace, For instance, University systems can be viewed as an extraordinary instance of big business systems: in such a domain, a large number of the associating gadgets are impermanent and not controlled by the University, additionally difficult security and asset allotment. Furthermore, Universities should frequently offer help for inquire about testbeds and trial conventions.

Sufficient administration is basically essential in Enterprise situations, and SDN can be utilized to automatically uphold and change arrange strategies and in addition enable screen to organize movement and tune arrange execution.

Furthermore, SDN can be utilized to streamline the system by freeing it from middleboxes and coordinating their usefulness inside the system controller. Some remarkable cases of middlebox usefulness that has been executed utilizing SDN incorporate NAT, firewalls, stack balancers [74] [75], and organize get to control [76]. On account of more mind boggling middleboxes with functionalities that can't be straightforwardly executed without execution corruption (e.g., profound bundle investigation), SDN can be utilized to give brought together control and administration [77].

The work displayed in [78] addresses the issues identified with predictable system refreshes. Arrangement changes are a typical wellspring of unsteadiness in systems and can prompt blackouts, security blemishes, and execution interruptions. In [78], an arrangement of abnormal state deliberations are recommended that permit organize managers to refresh the whole system, ensuring that each parcel crossing the system is handled by precisely one reliable worldwide system design. To help these deliberations, a few OpenFlow-based refresh components were produced.

As talked about in before areas, OpenFlow advanced from Ethane [20], a system engineering composed particularly to address the issues looked by big business systems.

**B. Data Centers**

Server farms have advanced at a stunning pace as of late, always endeavouring to meet progressively higher and quickly evolving request. Cautious movement administration and strategy requirement is basic while working at such substantial scales, particularly when any administration interruption or extra postponement may prompt gigantic profitability or potentially benefit misfortune. Because of the difficulties of designing systems of this scale and intricacy to powerfully adjust to application necessities, it is regularly the case that server farms are provisioned for top request; therefore, they run well underneath limit more often than not yet are prepared to quickly benefit higher workloads.

An undeniably vital thought is vitality utilization, which has a non-minor cost in substantial scale server farms. Heller et al. [79] demonstrates that much research has been centredon enhanced servers and cooling (70% of aggregate vitality) through better equipment or programming administration, however the server farm's system framework (which represents 10-20% of the aggregate vitality cost) still devoured 3 billion kWh in 2006. They proposed ElasticTree, a system wide power director that uses SDN to locate the base power arrange subset which fulfils current activity conditions and turns off switches that are not required. Subsequently, they indicate vitality reserve funds between 25-62% under shifting movement conditions. One can envision that these funds can be additionally expanded if utilized as a part of parallel with server administration and virtualization; one probability is the Honeyguide [80] way to deal with vitality advancement which utilizes virtual machine movement to build the quantity of machines and switches that can be shutdown.

Be that as it may, not all SDN arrangements might be proper in superior systems. While rearranged activity administration and deceivability are valuable, it must be sensibly adjusted with versatility and execution

overhead. Curtis et al. [34] trust that OpenFlow too much couples focal control and finish deceivability, when as a general rule just "noteworthy" streams should be dealt with; this may prompt bottlenecks as the control-information correspondence adds postponement to stream setup while switches are over-burden with a large number of stream table sections. In spite of the fact that forceful utilization of proactive strategies and special case guidelines may resolve that issue, it might undermine the capacity of the controller to have the correct granularity to adequately oversee movement and assemble insights. Their structure, DevoFlow, proposes some humble outline changes to keep streams in the information plane however much as could be expected while keeping up enough deceivability for viable stream administration. This is refined by pushing duty over most streams back to the switches and including more effective insights gathering systems, through which "critical" streams (e.g. enduring, high-throughput) are distinguished and overseen by the controller. In a heap adjusting recreation, their answer had 10-53 times less stream table sections and 10-42 times less control messages by and large finished OpenFlow.

A viable case of a genuine use of the SDN idea and design with regards to server farms was exhibited by Google in mid-2012. The organization displayed at the Open Network Summit [81] a huge scale usage of a SDN-based system interfacing its server farms. The work in [82] exhibits in more detail the plan, usage, and assessment of B4, a WAN interfacing Google's server farms around the world. This work depicts one of the first and biggest SDN arrangements. The inspiration was the requirement for redid directing and movement designing and the way that the level of versatility, adaptation to non-critical failure, cost proficiency and control required, couldn't be accomplished by methods for a conventional WAN engineering. An altered arrangement was proposed and an OpenFlow-based SDN engineering was worked to control individual switches. Following three years underway, B4 is appeared to be productive as in it drives many connections at close to 100% usage while part streams among numerous ways. Besides, the experience detailed in the work demonstrates that the bottleneck coming about because of control-plane to information plane correspondence and overhead in equipment writing computer programs are essential issues to be considered in future work.

**C. Infrastructure-based Wireless Access Networks**

A few endeavours have concentrated on omnipresent availability with regards to framework based remote access systems, for example, cell and Wi-Fi.

For instance, the OpenRoads venture [83], [84] imagines a world in which clients could uninhibitedly and flawlessly move crosswise over various remote foundations which might be overseen by different suppliers. They proposed the organization of a SDN-based remote design that is in reverse good, yet open and sharable between various specialist organizations. They utilize a testbed utilizing OpenFlow-empowered remote gadgets, for example, Wi-Fi APs and WiMAX base stations controlled by NOX– and FlowVisor controllers and show enhanced execution on handover occasions. Their vision gave motivation to resulting work [85] that endeavours to address particular prerequisites and difficulties in conveying a product characterized cell arrange.

Odin [86] presents programmability in big business remote LAN situations. Specifically, it manufactures an entrance point deliberation on the controller that isolates the affiliation state from the physical access point, empowering proactive versatility administration and load adjusting without changes to the customer. At the flip side of the range, OpenRadio [87] concentrates on sending a programmable remote information plane that gives adaptability at the PHY and MAC layers (rather than layer-3 SDN) while meeting strict execution and time due dates. The framework is intended to give a measured interface that can procedure activity subsets utilizing distinctive conventions, for example, Wi-Fi, WiMAX, 3GPP LTE-Advanced, and so forth. In view of the possibility of detachment of the choice and sending planes, an administrator may express choice plane principles and comparing activities, which are gathered from preparing plane modules (e.g., FFT, Viterbi disentangling, and so on.); the final product is a state machine that communicates a completely utilitarian convention.

**D. Optical Networks**

Taking care of information movement as streams, permits programming characterized net-works, and OpenFlow organizes specifically, to help and coordinate numerous system advancements. Thus, it is

conceivable to give likewise innovation freethinker brought together control for optical transport organizes and encouraging communication between both bundle and circuit-exchanged systems. As indicated by the Optical Transport Working Group (OTWG) made in 2013 by the Open Network Foundation (ONF), the advantages from applying SDN and the OpenFlow standard specifically to optical transport systems include: enhancing optical trans-port system control and administration adaptability, empowering arrangement of outsider administration and control frameworks, and conveying new administrations by utilizing virtualization and SDN [88].

There has been a few endeavours and proposition to control both circuit exchanged and bundle exchanged systems utilizing the OpenFlow convention. In [89] a NetFPGA [90] stage is utilized as a part of the proposition of a bundle exchanging and circuit exchanged systems structures in light of Wavelength Selective Switching (WSS), utilizing the OpenFlow convention. Another control plane engineering in view of OpenFlow for empowering SDN operations in optical systems was proposed in [91], which examines particular necessities and portrays execution of OpenFlow convention augmentations to help optical transport systems.

A proof-of-idea show of an OpenFlow-based wavelength way control in straightforward optical systems is displayed in [92]. In this work, virtual Ethernet interfaces (veths) are presented. These veths, are mapped to physical interfaces of an optical hub (e.g. photonic cross-associate - PXC), and empower a SDN controller (e.g. the NOX controller for this situation) to work the optical lightpaths (e.g., by means of the OpenFlow convention). In their test setup, they quantitatively assess organize execution measurements, for example, the dormancy of lightpath setup and discharge, and check the attainability of directing and wavelength task, and the dynamic control of optical hubs in an OpenFlow-based system made by four PXCs hubs in a work topology.

A Software Defined Optical Network (SDON) engineering is presented in [93] and a QoS-mindful bound together control convention for optical burst exchanging in OpenFlow-based SDON is created. The execution of the proposed convention was assessed with the customary GMPLS-based disseminated convention and the outcomes demonstrate that SDON offers a framework to help bound together control conventions to better upgrade organize execution and enhance limit.

### E. Home and Small Business

A few tasks have inspected how SDN could be utilized as a part of littler systems, for example, those found in the home or independent companies. As these conditions have turned out to be progressively intricate and predominant with the boundless accessibility of minimal effort organize gadgets, the requirement for more cautious system administration and more tightly security has correspondingly expanded. Inadequately secured systems may end up plainly unwitting targets or has for malware, while blackouts because of system setup issues may cause disappointment or lost business. Lamentably, it isn't viable to have a devoted system executive in each home and office.

Calvert et al. [94] state that the initial phase in overseeing home systems is to realize what is really happening; all things considered, they proposed instrumenting the system door/controller to go about as a "Home Network Data Recorder" to make logs that might be used for investigating or different purposes.

Conversely, Mortier et al. [96] trust that clients want more noteworthy comprehension and control over their systems' conduct; instead of following conventional arrangements, a home system might be better overseen by their clients who better comprehend the elements and requirements of their condition. Towards this objective, they made a model system in which SDN is utilized to give clients a view into how their system is being used while offering a solitary purpose of control.

## V. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

As SDN turns out to be all the more broadly received and conventions, for example, OpenFlow are additionally characterized, new arrangements are proposed and new difficulties emerge. In this section we discuss various challenges posed by SDN as well as future research directions, namely:

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 11
November 2017

- Controller and switch design.
- Scalability and performance in SDNs.
- Controller-service interfacing.
- Virtualization and cloud service applications.
- Information centric networking.
- Enabling heterogeneous networking with SDN.

**A. Controller and Switch Design**

SDN raises noteworthy versatility, execution, heartiness, and security challenges. Beneath we audit various re-seek endeavours concentrating on tending to these issues at the switch– and controller configuration level.

In DIFANE [35], stream passages are proactively pushed to switches trying to diminish the quantity of solicitations to the controller. Devoflow [34] proposes to deal with "brief" streams in switches and "seemingly perpetual" streams in the controller to relieve stream setup postponement and controller overhead. The work proposed in [28] advocates supplanting counters on ASIC by a flood of run coordinating records and handling them in the CPU to enable effective access to counters. FLARE [98] is another system hub demonstrate concentrating on "profoundly programmable systems" that gives programmability to the information plane, the control plane, and also the interface between them. The work introduced in [99] talks about essential viewpoints in controller configuration including progressive control, information model, versatility, and extensibility.

**B. Software-Defined Internetworking**

The Internet has changed the path we, as people and as a general public, live, work, lead business, mingle, get diversion, and so forth. Accordingly, the Internet is currently considered piece of our general public's basic framework much like the power, water, and transportation lattices.

Adaptability and execution necessities from progressively complex applications have represented an assortment of difficulties hard to address with the present Internet engineering. This has driven the examination group to inspect "fresh start" arrangements [105]. As the Internet has developed past the time when a "banner day, for example, the one used to "redesign" the ARPANET with the TCP/IP convention suite, would be practical, another significant test is advancing its physical framework and conventions. An outstanding case is the sending of IPv6: regardless of over 10 years in the models track and two overall arrangement occasions, IPv4 still makes up the lion's share of Internet activity.

**C. Controller-Service Interaction**

While controller-switch ("southbound") communication is genuinely very much characterized in conventions, for example, OpenFlow and ForCES, there is no standard for collaborations amongst controllers and system administrations or applications ("northbound"). One conceivable clarification is that the northbound interface is characterized altogether in programming, while controller-switch cooperation must empower equipment usage.

Moreover, the northbound API ought to enable applications to apply distinctive arrangements to a similar stream (e.g. sending by goal and checking by source IP). The work in [114] proposed modularization to guarantee that principles introduced to perform one undertaking don't supersede different standards. This was refined by methods for a deliberation layer actualized with a dialect in light of Frenetic.

Until the point that a reasonable northbound interface standard rises, SDN applications will keep on being produced in a "specially appointed" form and the idea of adaptable and versatile "system applications" may need to hold up.

**D. Virtualization and Cloud Services**

The interest for virtualization and cloud administrations has been developing quickly and drawing in impressive enthusiasm from industry and the scholarly world. The difficulties it presents incorporate quick provisioning, effective asset administration, and adaptability which can be tended to utilizing SDN's control show.

For instance, FlowVisor [48] and AutoSlice [115] make distinctive cuts of system assets (e.g., data transfer capacity, topology, CPU, sending table), appoint them to various controllers, and uphold separation between cuts. Other SDN controllers can be utilized as a system backend to help virtualization in cloud working frameworks, for example, Floodlight for OpenStack [38] and NOX for Mirage [116]. FlowN [117] intends to offer an adaptable answer for organize virtualization by giving an effective mapping amongst virtual and physical systems and by utilizing versatile database frameworks.

### E. Information-Centric Networking

Data Centric Networking (ICN) is another worldview proposed for the future design of the Internet, which means to expand the productivity of substance conveyance and substance accessibility. This new idea has been promoted as of late by various engineering proposition, for example, Content-Centric Networking (CCN), otherwise called the Named Data Networking (NDN) venture [122]. Their driving inspiration is that the present Internet is data driven, yet organizing innovation is as yet centred around the possibility of area based tending to and have to-have correspondence. By proposing a design that locations named information as opposed to named has, content conveyance is actualized straightforwardly into the system texture instead of depending on the entangled mapping, accessibility, and security instruments as of now used to outline to a solitary area.

### F. Heterogeneous Network Support

Future systems will turn out to be progressively more heterogeneous, interconnecting clients and applications over net-works extending from wired, framework based remote (e.g., cellular– based systems, remote work systems), to foundation less remote systems (e.g. versatile specially appointed net-works, vehicular systems). Meanwhile, portable activity has been expanding exponentially finished the previous quite a long while, and is required to increment 18– crease by 2016, with more versatile associated gadgets than the total populace, which is now a reality [129]. As cell phones with different system interfaces wind up noticeably ordinary, clients will request excellent correspondence benefit paying little heed to area or kind of system get to. Self-sorting out systems (e.g., remote multi-jump impromptu systems) may frame to expand the scope of framework based systems or handle roundabout availability interruptions. Self-sorting out systems may consequently empower an assortment of new applications, for example, cloud-based administrations, vehicular correspondence, group administrations, social insurance de-uniform, crisis reaction, and natural checking, to give some examples. Effective substance conveyance over remote access systems will end up noticeably basic, and self– sorting out systems may turn into a common piece without bounds half and half Internet.

## VI. CONCLUDING REMARKS

In this paper, we provided an overview of programmable networks and, in this context, examined the emerging field of Software-Defined Networking (SDN). We look at the history of programmable networks, from early ideas until recent developments. In particular we described the SDN architecture in detail as well as the OpenFlow [2] standard. We presented current SDN implementations and testing platforms and examined network services and applications that have been developed based on the SDN paradigm. We concluded with a discussion of future directions enabled by SDN ranging from support for heterogeneous networks to Information Centric Networking (ICN).

## VII. REFERENCES

[1] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong,
R. Gopal, and J. Halpern. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810 (Proposed Standard), March 2010.
[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson,
J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2):69–74, 2008.
[3] Open networking foundation. https://www.opennetworking.org/about.

[4] Open Networking Research Center (ONRC). http://onrc.net.
[5] Thomas A. Limoncelli. Openflow: a radical new idea in networking. Commun. ACM, 55(8):42–47, August 2012.
[6] A.T. Campbell, I. Katzela, K. Miki, and J. Vicente. Open signaling for atm, internet and mobile networks (opensig'98). ACM SIGCOMM Computer Communication Review, 29(1):97–108, 1999.
[7] A. Doria, F. Hellstrand, K. Sundell, and T. Worster. General Switch Management Protocol (GSMP) V3. RFC 3292 (Proposed Standard), June 2002.
[8] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Minden. A survey of active network research. Communications Magazine, IEEE, 35(1):80–86, 1997.
[9] D.L. Tennenhouse and D.J. Wetherall. Towards an active network architecture. In DARPA Active NEtworks Conference and Exposition, 2002. Proceedings, pages 2–15. IEEE, 2002.
[10] J.T. Moore and S.M. Nettles. Towards practical programmable packets. In Proceedings of the 20th Conference on Computer Communications (INFOCOM). Citeseer, 2001.
[11] Devolved Control of ATM Networks. http://www.cl.cam.ac.uk/research/srg/netos/old-projects/dcan/#pub.
[12] J. E. Van Der Merwe and I. M. Leslie. Switchlets and dynamic virtual atm networks. In Proc Integrated Network Management V, pages 355–
368. Chapman and Hall, 1997.
[13] J.E. Van der Merwe, S. Rooney, I. Leslie, and S. Crosby. The tempest-a practical framework for network programmability. Network, IEEE, 12(3):20–28, 1998.
[14] J. Rexford, A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers,
G. Xie, J. Zhan, and H. Zhang. Network-wide decision making: Toward a wafer-thin control plane. In Proc. HotNets, pages 59–64. Citeseer, 2004.
[15] A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford,
G. Xie, H. Yan, J. Zhan, and H. Zhang. A clean slate 4d approach to network control and management. ACM SIGCOMM Computer Communication Review, 35(5):41–54, 2005.
[16] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and
J. van der Merwe. Design and implementation of a routing control platform. In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2, pages 15–28. USENIX Association, 2005.
[17] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards an operating system for networks. ACM SIGCOMM Computer Communication Review, 38(3):105–110, 2008.
[18] R. Enns. NETCONF Configuration Protocol. RFC 4741 (Proposed Standard), December 2006. Obsoleted by RFC 6241.
[19] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. Simple network management protocol (snmp), rfc1157, 1990.
[20] M. Casado, M.J. Freedman, J. Pettit, J. Luo, N. McKeown, and
S. Shenker. Ethane: Taking control of the enterprise. ACM SIGCOMM Computer Communication Review, 37(4):1–12, 2007.
[21] Z. Cai, AL Cox, and TSE Ng. Maestro: A system for scalable openflow control. Technical Report TR10-08, Rice University, December 2010.
[22] Beacon. https://openflow.stanford.edu/display/Beacon/Home. Simple Network [23] Access Control http://www.openflow.org/wp/snac/.
[24] Helios by nec. http://www.nec.com/.
[25] Andrea Passarella. Review: A survey on content-centric technologies for the current internet: Cdn and p2p solutions. Comput. Commun., 35(1):1–32, January 2012.
[26] Zhiliang Wang, Tina Tsou, Jing Huang, Xingang Shi, and Xia Yin. Analysis of Comparisons between OpenFlow and ForCES, March 2012.
[27] Guohan Lu, Rui Miao, YongqiangXiong, and Chuanxiong Guo. Using cpu as a traffic co-processing unit in commodity switches. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 31–36, New York, NY, USA, 2012. ACM.
[28] Jeffrey C. Mogul and Paul Congdon. Hey, you darned counters!: get off my asic! In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 25–30, New York, NY, USA, 2012. ACM.
[29] Yan Luo, Pablo Cascon, Eric Murray, and Julio Ortega. Accelerating openflow switching with network processors. In Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS '09, pages 70–71, New York, NY, USA, 2009. ACM.
[30] VoravitTanyingyong, Markus Hidell, and Peter Sj¨odin. Improving pc-based openflow switching performance. In Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Commu-nications Systems, ANCS '10, pages 13:1–13:2, New York, NY, USA, 2010. ACM.

[31] A. Bianco, R. Birke, L. Giraudo, and M. Palacin. Openflow switching: Data plane performance. In Communications (ICC), 2010 IEEE International Conference on, pages 1–5, May.
[32] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia. Modeling and performance evaluation of an openflow architecture. In Teletraffic Congress (ITC), 2011 23rd International, pages 1–7, Sept.
[33] Brent Stephens, Alan Cox, Wes Felter, Colin Dixon, and John Carter. Past: scalable ethernet for data centers. In Proceedings of the 8th international conference on Emerging networking experiments and technologies, CoNEXT '12, pages 49–60, New York, NY, USA, 2012. ACM.
[34] Andrew R. Curtis, Jeffrey C. Mogul, Jean Tourrilhes, Praveen Yala-gandula, Puneet Sharma, and Sujata Banerjee. Devoflow: scaling flow management for high-performance networks. SIGCOMM Comput. Commun. Rev., 41(4):254–265, August 2011.
[35] M. Yu, J. Rexford, M.J. Freedman, and J. Wang. Scalable flow-based networking with difane. In Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM, pages 351–362. ACM, 2010.
[36] Yossi Kanizo, David Hay, and Isaac Keslassy. Palette: Distributing tables in software-defined networks. In INFOCOM, pages 545–549, 2013.
[37] Nanxi Kang, Zhenming Liu, Jennifer Rexford, and David Walker. Op-timizing the one big switch abstraction in software-defined networks.
[38] Floodlight, an open sdn controller. http://floodlight.openflowhub.org/.
[39] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sher-wood. On controller performance in software-defined networks. In USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2012.
[40] Andreas Voellmy and Junchang Wang. Scalable software defined network controllers. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, SIGCOMM '12, pages 289–290, New York, NY, USA, 2012. ACM.
[41] Brandon Heller, Rob Sherwood, and Nick McKeown. The controller placement problem. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 7–12, New York, NY, USA, 2012. ACM.
[42] K. Phemius and M. Bouet. Openflow: Why latency does matter. In Integrated Network Management (IM 2013), 2013 IFIP/IEEE Interna-tional Symposium on, pages 680–683, 2013.
[43] S.H. Yeganeh, A. Tootoonchian, and Y. Ganjali. On scalability of software-defined networking. Communications Magazine, IEEE, 51(2):136–141, February.
[44] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu,
R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al. Onix: A distributed control platform for large-scale production networks. OSDI, Oct, 2010.
[45] A. Tootoonchian and Y. Ganjali. Hyperflow: A distributed control plane for openflow. In Proceedings of the 2010 internet network management conference on Research on enterprise networking, pages 3–3. USENIX Association, 2010.
[46] Dan Levin, Andreas Wundsam, Brandon Heller, Nikhil Handigol, and Anja Feldmann. Logically centralized?: state distribution trade-offs in software defined networks. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 1–6, New York, NY, USA, 2012. ACM.
[47] SoheilHassasYeganeh and YasharGanjali. Kandoo: a framework for efficient and scalable offloading of control applications. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 19–24, New York, NY, USA, 2012. ACM.
[48] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Hand-igol, T.Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, et al. Carving research slices out of your production networks with openflow. ACM SIGCOMM Computer Communication Review, 40(1):129–130, 2010.
[49] http://www.itu.int/en/ITU T/sdn/Pages/default.aspx. Itutelecommuni-cation standardization sector's sdn portal, 2013.
[50] http://irtf.org/sdnrg. Software-defined networking research group -sdnrg at irtf, 2013.
[51] http://www.openstack.org/. Openstack, 2013.
[52] http://www.opendaylight.org/. Opendaylight, 2013.
[53] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a lap-top: rapid prototyping for software-defined networks. In Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, 2010.
[54] T.R. Henderson, M. Lacage, G.F. Riley, C. Dowell, and JB Kopena. Network simulations with the ns-3 simulator. SIGCOMM demonstra-tion, 2008.
[55] Open vswitch and ovs-controller. http://openvswitch.org/. Pantou: Openflow 1.0 for [56] http://www.openflow.org/wk/index.php/
Flow 1.0 for OpenWRT.
[57] ofsoftswitch13 - cpqd. https://github.com/CPqD/ofsoftswitch13.

[58] Indigo: Open source openflowhttp://www.openflowhub.org/display/Indigo/.
[59] Pox. http://www.noxrepo.org/pox/about-pox/.
[60] Mul. http://sourceforge.net/p/mul/wiki/Home/.
[61] Tremaopenflow controller framework. https://github.com/trema/trema.
[62] Jaxon:java-basedopenflow controller. http://jaxon.onuos.org/.
[63] Ryu. http://osrg.github.com/ryu/.
[64] The nodeflowopenflow controller. http://garyberger.net/?p=537.
[65] Node.js. http://nodejs.org/.
[66] Marcelo R. Nascimento, Christian E. Rothenberg, Marcos R. Salvador, Carlos N. A. Corrˆea, Sidney C. de Lucena, and Maur´ıcio F. Magalh˜aes. Virtual routers as a service: the routeflow approach leveraging software-defined networks. In Proceedings of the 6th International Conference on Future Internet Technologies, CFI '11, pages 34–37, New York, NY, USA, 2011. ACM.
[67] Christian Esteve Rothenberg, Marcelo Ribeiro Nascimento, Mar-cos Rogerio Salvador, Carlos Nilton Araujo Corrˆea, Sidney Cunha de Lucena, and Robert Raszuk. Revisiting routing control platforms with the eyes and muscles of software-defined networking. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 13–18, New York, NY, USA, 2012. ACM.
[68] M. Canini, D. Venzano, P. Peresini, D. Kostic, and J. Rexford. A nice way to test openflow applications. NSDI, Apr, 2012.
[69] Haohui Mai, Ahmed Khurshid, Rachit Agarwal, Matthew Caesar, P.Brighten Godfrey, and Samuel Talmadge King. Debugging the data plane with anteater. In Proceedings of the ACM SIGCOMM 2011 conference, SIGCOMM '11, pages 290–301, New York, NY, USA, 2011. ACM.
[70] Ahmed Khurshid, Wenxuan Zhou, Matthew Caesar, and P. Brighten Godfrey. Veriflow: verifying network-wide invariants in real time. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 49–54, New York, NY, USA, 2012. ACM.
[71] Andreas Wundsam, Dan Levin, SriniSeetharaman, and Anja Feldmann. Ofrewind: enabling record and replay troubleshooting for networks. In Proceedings of the 2011 USENIX conference on USENIX annual technical conference, USENIXATC'11, pages 29–29, Berkeley, CA, USA, 2011. USENIX Association.
[72] Nikhil Handigol, Brandon Heller, VimalkumarJeyakumar, David Mazi´eres, and Nick McKeown. Where is the debugger for my software-defined network? In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 55–60, New York, NY, USA, 2012. ACM.
[73] Sdn troubleshooting simulator. http://ucb-sts.github.com/sts/.
[74] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Jo-hari. Plug-n-serve: Load-balancing web traffic using openflow. ACM SIGCOMM Demo, 2009.
[75] R. Wang, D. Butnariu, and J. Rexford. Openflow-based server load balancing gone wild. In Workshop of HotICE, volume 11, 2011.
[76] A.K. Nayak, A. Reimers, N. Feamster, and R. Clark. Resonance: Dynamic access control for enterprise networks. In Proceedings of the 1st ACM workshop on Research on enterprise networking, pages 11–18. ACM, 2009.
[77] A. Gember, P. Prabhu, Z. Ghadiyali, and A. Akella. Toward software-defined middlebox networking. 2012.
[78] Mark Reitblatt, Nate Foster, Jennifer Rexford, Cole Schlesinger, and David Walker. Abstractions for network update. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, SIGCOMM '12, pages 323–334, New York, NY, USA, 2012. ACM.
[79] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma,
S. Banerjee, and N. McKeown. Elastictree: Saving energy in data center networks. In Proceedings of the 7th USENIX conference on Networked systems design and implementation, pages 17–17. USENIX Association, 2010.
[80] H. Shirayanagi, H. Yamada, and K. Kono. Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks. In Computers and Communications (ISCC), 2012 IEEE Symposium on, pages 000460–000467. IEEE, 2012.
[81] Inter-datacenter wan with centralized te using sdn and openflow. In Open Networking Summit, April 2012.
[82] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, SubbaiahVenkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. B4: Experience with a globally-deployed software defined wan. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pages 3–14. ACM, 2013.
[83] K.K. Yap, R. Sherwood, M. Kobayashi, T.Y. Huang, M. Chan,
N. Handigol, N. McKeown, and G. Parulkar. Blueprint for introducing innovation into wireless mobile networks. In Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures, pages 25–32. ACM, 2010.
[84] K.K. Yap, M. Kobayashi, R. Sherwood, T.Y. Huang, M. Chan,

N. Handigol, and N. McKeown. Openroads: Empowering research in mobile networks. ACM SIGCOMM Computer Communication Review, 40(1):125–126, 2010.
[85] L.E. Li, Z.M. Mao, and J. Rexford. Toward software-defined cellular networks. In Software Defined Networking (EWSDN), 2012 European Workshop on, pages 7–12, 2012.
[86] Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao. Towards programmable enterprise wlans with odin. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 115–120, New York, NY, USA, 2012. ACM.
[87] M. Bansal, J. Mehlman, S. Katti, and P. Levis. Openradio: a pro-grammable wireless dataplane. In Proceedings of the first workshop on Hot topics in software defined networks, pages 109–114. ACM, 2012.
[88] Optical transport working group otwg. In Open Networking Foundation ONF, 2013.
[89] V. Gudla, S. Das, A. Shastri, G. Parulkar, N. McKeown, L. Kazovsky, and S. Yamashita. Experimental demonstration of openflow control of packet and circuit switches. In Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC), pages 1–3, 2010.
[90] Netfpga platform. http://netfpga.org.
[91] Dimitra E. Simeonidou, Reza Nejabati, and MayurChannegowda. Soft-ware defined optical networks technology and infrastructure: Enabling software-defined optical network operations. In Optical FiberCommu-nication Conference/National Fiber Optic Engineers Conference 2013, page OTh1H.3. Optical Society of America, 2013.
[92] Lei Liu, T. Tsuritani, I. Morita, HongxiangGuo, and Jian Wu. Openflow-based wavelength path control in transparent optical net-works: A proof-of-concept demonstration. In Optical Communication (ECOC), 2011 37th European Conference and Exhibition on, pages 1–3, 2011.
[93] A.N. Patel, P.N. Ji, and Ting Wang. Qos-aware optical burst switching in openflow based software-defined optical networks. In Optical Network Design and Modeling (ONDM), 2013 17th International Conference on, pages 275–280, 2013.
[94] K.L. Calvert, W.K. Edwards, N. Feamster, R.E. Grinter, Y. Deng, and
X. Zhou. Instrumenting home networks. ACM SIGCOMM Computer Communication Review, 41(1):84–89, 2011.
[95] N. Feamster. Outsourcing home network security. In Proceedings of the 2010 ACM SIGCOMM workshop on Home networks, pages 37–42. ACM, 2010.
[96] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, AW Moore,
A. Koliousis, and J. Sventek. Control and understanding: Owning your home network. In Communication Systems and Networks (COM-SNETS), 2012 Fourth International Conference on, pages 1–10. IEEE, 2012.
[97] S. Mehdi, J. Khalid, and S. Khayam. Revisiting traffic anomaly detection using software defined networking. In Recent Advances in Intrusion Detection, pages 161–180. Springer, 2011.
[98] Akihiro Nakao. Flare : Open deeply programmable network node architecture. http://netseminar.stanford.edu/10 18 12.html.
[99] R. Bifulco, R. Canonico, M. Brunner, P. Hasselmeyer, and F. Mir. A practical experience in designing an openflow controller. In Software Defined Networking (EWSDN), 2012 European Workshop on, pages 61–66, Oct.
[100] Controller performance comparisons. http://www.openflow.org/wk/index.php/Controller Performance Comparisons.
[101] David Erickson. The beacon openflow controller, 2012.
[102] Advait Dixit, Fang Hao, Sarit Mukherjee, T.V. Lakshman, and RamanaKompella. Towards an elastic distributed sdn controller. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, HotSDN '13, pages 7–12, New York, NY, USA, 2013. ACM.
[103] Martin Casado, TeemuKoponen, Scott Shenker, and Amin Tootoonchian. Fabric: a retrospective on evolving sdn. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 85–90, New York, NY, USA, 2012. ACM.
[104] Minlan Yu, Lavanya Jose, and Rui Miao. Software defined traffic mea-surement with opensketch. In Proceedings 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI'13, 2013.
[105] Anja Feldmann. Internet clean-slate design: what and why? SIGCOMM Comput. Commun. Rev., 37(3):59–64, July 2007.
[106] BarathRaghavan, Mart´ınCasado, TeemuKoponen, Sylvia Ratnasamy, Ali Ghodsi, and Scott Shenker. Software-defined internet architecture: decoupling architecture from infrastructure. In Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, pages 43–48, New York, NY, USA, 2012. ACM.
[107] R. Bennesby, P. Fonseca, E. Mota, and A. Passito. An inter-as routing component for software-defined networks. In Network Operations and Management Symposium (NOMS), 2012 IEEE, pages 138–145, 2012.
[108] E. Kissel, G. Fernandes, M. Jaffee, M. Swany, and M. Zhang. Driving software defined networks with xsp. In SDN12: Workshop on Software Defined Networks, 2012.

[109] Andreas Voellmy, Hyojoon Kim, and Nick Feamster. Procera: a language for high-level reactive network control. In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 43–48, New York, NY, USA, 2012. ACM.

[110] Nate Foster, Rob Harrison, Michael J. Freedman, Christopher Mon-santo, Jennifer Rexford, Alec Story, and David Walker. Frenetic: a network programming language. In Proceedings of the 16th ACM SIGPLAN international conference on Functional programming, ICFP '11, pages 279–291, New York, NY, USA, 2011. ACM.

[111] Timothy L. Hinrichs, Natasha S. Gude, Martin Casado, John C. Mitchell, and Scott Shenker. Practical declarative network manage-ment. In Proceedings of the 1st ACM workshop on Research on enterprise networking, WREN '09, pages 1–10, New York, NY, USA, 2009. ACM.

[112] Andreas Voellmy and Paul Hudak. Nettle: taking the sting out of programming network routers. In Proceedings of the 13th international conference on Practical aspects of declarative languages, PADL'11, pages 235–249, Berlin, Heidelberg, 2011. Springer-Verlag.

[113] Hyojoon Kim and N. Feamster. Improving network management with software defined networking. Communications Magazine, IEEE, 51(2):114–119, February.

[114] Christopher Monsanto, Joshua Reich, Nate Foster, Jennifer Rexford, and David Walker. Composing software-defined networks. In Pro-ceedings 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI'13, 2013.

[123] Xuan-Nam Nguyen, Damien Saucez, and Thierry Turletti. Efficient caching in Content-Centric Networks using OpenFlow. In 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 67–68. IEEE, April 2013.

[124] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi, and A. Detti. Supporting information-centric functionality in software defined net-works. IEEE ICC Workshop on Software Defined Networks, June 2012.

[125] N. Blefari-Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, and L. Veltri. An openflow-based testbed for information centric networking. Future Network & Mobile Summit, pages 4–6, 2012.

[126] Junho Suh, Hyogi Jung, Taekyoung Kwon, and Yanghee Choi. C-flow: Content-oriented networking over openflow. In Open Networking Summit, April 2012.

[127] D. Syrivelis, G. Parisis, D. Trossen, P. Flegkas, V. Sourlas, T. Korakis, and L. Tassiulas. Pursuing a software defined information-centric network. In Software Defined Networking (EWSDN), 2012 European Workshop on, pages 103–108, Oct.

[128] X.N. Nguyen. Software defined networking in wireless mesh network. Msc. thesis, INRIA, UNSA, August 2012.

[129] Cisco visual networking index: Global mobile data traffic forecast update, 2011–2016. Technical report, Cisco, February 2012.

[130] B. Rais, M. Mendonca, T. Turletti, and K. Obraczka. Towards truly het-erogeneous internets: Bridging infrastructure-based and infrastructure-less networks. In Communication Systems and Networks (COMSNETS), 2011 Third International Conference on, pages 1–10. IEEE, 2011.

[131] A. Coyle and H. Nguyen. A frequency control algorithm for a mobile adhoc network. In Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, November 2010.

[132] P. Dely, A. Kassler, and N. Bayer. Openflow for wireless mesh net-works. In Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pages 1–6. IEEE, 2011.