A Quick Survey on Selected Approaches for Preparing Programmable Networks

Pingping Lin, Jun Bi, Hongyu Hu, Tao Feng, Xiaoke Jiang
Network Research Center, Department of Computer Science, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology
4-204, FIT Building, Beijing, 100084, China
linpp@netarchlab.tsinghua.edu.cn, junbi@tsinghua.edu.cn, huhongyu@cernet.edu.cn
fengt09@mails.tsinghua.edu.cn, justok@netarchlab.tsinghua.edu.cn

ABSTRACT
Currently, new protocols or architectures related to core network layer or network forwarding equipment are hard to deploy. And the core network evolves slowly. To solve this problem, various programmable Internet architectures and approaches are proposed. Programmable networks allow network researchers (not only the equipment vendors like Cisco and Juniper) to program and manage their customized architectures or protocols. This paper firstly describes the ossification with the current Internet architecture. And then it analyzes several most representative approaches for programmable networks with their mechanisms, advantages, and shortcomings. By the analysis above, this paper at last discusses the future research trends and gives a detailed description of the key issues in the future research of the programmable networks.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Network communication

General Terms
Documentation

Keywords
Approach, Programmability, Resource Abstraction.

1. INTRODUCTION
After more than 30 years development, the Internet has achieved unprecedented prosperity and has become an essential infrastructure in our daily work and lives.

However, the Transfer Control Protocol (TCP)/Internet Protocol (IP)-based architecture of the Internet was born with defects. The best-effort forwarding strategy, for example, cannot provide users with required Quality of Service (QoS); network management is hard to implement; network security vulnerabilities are increasingly exposed; IP address resources are almost exhausted; and routing in a large-scale network suffers poor scalability.

Efforts to reform TCP/IP-based architecture have been unceasing, with new protocols and algorithms being introduced one after another. Unfortunately, new changes always need years to achieve the deployment like TRILL [1] use six years from design to deployment, and many changes have not been incorporated into the Internet architecture due to the following barriers: 1) New protocol needs a long standardization process. 2) The software and the hardware in network equipment are vertically integrated and closed. Researchers have to rely on vendors to implement the new protocols. 3) In reality, new protocols related to the Internet core network layer or network devices always need global deployment or modifications on all network devices. However, network operators and device manufacturers are often disinclined to make such large-scale deployments or modifications due to lack of incentives and lack of trust to the maturation of protocols. The evolution of Internet core technology has come to a standstill.

To solve the problem above, programmable networks were proposed as a solution for the fast and flexible deployment of new network protocols. The programmable networks in this paper mainly refer to networks allowing network researchers (not only the equipment vendors) to program and manage their customized network architectures or protocols on their own.

The remainder of this paper is organized as follows: in Section 2, we survey a number of the currently most representative approaches for preparing programmable networks. Section 3 gives the future research trends and Section 4 discusses key research issues for further exploration. And then we conclude in Section 5.

2. REPRESENTATIVE APPROACHES FOR PROGRAMMABLE NETWORKS

2.1 Improvements based on Hardware Router

2.1.1 Active Network
Active Network [2] allows packet to carry programs. And the network equipment like routers or switches can execute programs in packet. In such way, new services can be deployed at runtime within the confines of existing IP

AINTEC'11, November 9–11, 2011, Bangkok, Thailand.

networks and the network environment can be dynamically changed. To achieve the function above, active network requires adding many new functionalities to the current network equipment.

To understand the program in active packet, active node like switch or router needs CPU to process it; in this aspect it is not efficient. Besides, it depends on vendors to change their hardware and closed network equipment system significantly.

2.1.2 OpenSig

OpenSig [2] uses a set of open programmable network interfaces to open the access to switches and routers. In this way, OpenSig enables third party software providers to enter the market. It also suffers from the problem of exposing the closed network equipment system which vendors are not willing to.

2.1.3 The Juniper Operating System Software Development Kit

Juniper opened up a Software Development Kit (SDK) [4] on the Juniper Operating System (JUNOS) to enable developers to innovate on top of JUNOS and Juniper Networks platforms. So Juniper partners who want to develop applications can create, deploy, and validate innovative applications tailored to their needs for their routers on their own.

The SDK (formerly the Partner Solution Development Platform) provides developers with interfaces to routing and services functions of the JUNOS. Developers can use the SDK in Routing Engine to write a daemon on the Routing Engine panel, or use the services SDK to write an application on the MultiServices PIC [4] or MS DPC [4] panel, and the applications between the two panels can communicate with each other.

It's a pity that only in the Juniper platform developers can create, deploy, and validate innovations. So in network which not only uses the Juniper equipment but also uses equipment from others like Cisco, a new proposed protocol like improved BGP which needs global deployment still unable to run successfully in such circumstance, because other equipment might not support the new protocols. Besides, the opened functions are limited and functions to be opened are totally determined by the Juniper.

2.2 Software Router

Software routers are software and PC based solutions like Click Router [13], XORP [10] and so on. They are installed on normal host or servers. And then researchers can program their new architectures or protocols freely.

2.3 Testbed

Testbed like PlanetLab [6] and GENI [11] also gives architecture, protocols, and algorithms of the future Internet a programmable environment for creation, evaluation and verification. But there is no large real user traffic, economic factors, routing strategy or deployment incentives in testbed. Experiments running successfully in testbed cannot prove that they will run successfully in real network.

2.4 Virtualization

2.4.1 Network Virtualization

Network Virtualization [7] can logically segment a single physical network into multiple logical networks and provides a powerful way to run multiple Internet architectures each customized to a specific purpose, at the same time over a shared substrate. The common technologies to realize network virtualization are: VLAN, VPN.

Further, network virtualization suggests decoupling the role of the traditional Internet Service Providers (ISPs) into two: infrastructure providers (InPs), who manage the physical infrastructure, and service providers (SPs), who create virtual networks by aggregating resources from multiple infrastructure providers and offer end-to-end network services [7]. And the service providers could dynamically compose multiple heterogeneous virtual networks that coexist together and are isolated from each other. Then, researchers can program freely on each network.

It still needs lots of work to do to realize the coexistence of multiple networking architectures, such as system stability, security, resource management like resource calculation and resource isolation, and so on.

2.4.2 Overlay Network

An overlay network [5] is a logical network which is built on the top of another network. In this way, the overlay solutions enable innovation and programmability on overlay networks. Nodes in the overlay network can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links in the underlying network. However, overlay networks do not have the ability to control the routing between two overlay network nodes.

2.4.3 Virtual Routers

Juniper and Cisco both support virtual routers [15] [16] which partition a single router into multiple logical devices that perform independent routing tasks. The functionalities of virtualized router are a sub set of the master router. But they do not support innovation and programmability of new network architectures or protocols by researchers.

2.5 Software Defined Network

Recently, Software Defined Network (SDN) [14] is emerging as a new way to architect networks by providing network programmability and by exposing network APIs. It is a new network architecture that enables innovations by researchers, operators, application/service providers, and third parties as well as by network equipment vendors.

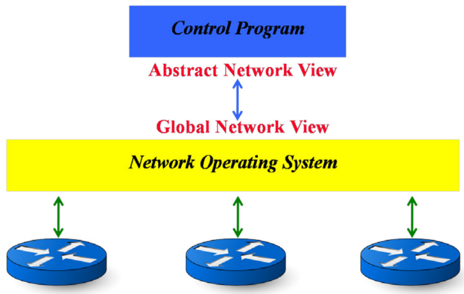

Figure 1. Three control plane abstractions create SDN

SDN focuses on the role of abstractions in networking and believes the network control plane needs three abstractions: 1) Forwarding Abstractions: a flexible forwarding model which should abstracts away forwarding hardware and be crucial for evolving beyond vendor-specific solutions. 2) State Distribution Abstractions: Control program should not have to deal with varies of distributed state and should operate on network view. And the network operating system as in the Figure above can create a network view by communicating with forwarding elements. 3) Specification Abstraction: mapping the controls expressed on the abstract view into configuration of the global view.

The bottom forwarding abstractions separate the data and control planes with a well defined vendor agnostic API/protocol between the data and control planes. This is a breakthrough in the principles of the Internet. So the standardization of the forwarding abstractions will make the control plane compatible to different data plane hardware from different vendors.

2.5.1 OpenFlow

OpenFlow [3] is treated as one of the implementations of SDN and is a solution providing real data flows for Internet researchers to carry out their Internet innovations by adopting a structure of OpenFlow controller and OpenFlow switch as in the Figure below:

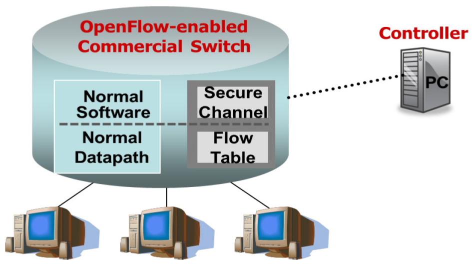

Figure 2. The OpenFlow Architecture

OpenFlow separates the control plane from the data plane. And the control plane can be moved to the external controllers which can be located in personal computers. So the main contribution of OpenFlow is giving Internet the ability to get rid of the dependence that vendors implement all network protocols.

OpenFlow enables networks to evolve by giving the remote controller the power to modify and control the behavior of OpenFlow network devices, through a well-defined forwarding instruction set to perform related operations. In this way, its technology can be used for software defined network.

The limitations and future works of OpenFlow: OpenFlow adopts the centralized management solution. Currently, it works in the local area networks, and mainly deals with IPv4 protocol and the statistics function of the flowtable is rough. Besides, it faces lots of challenges to achieve the worldwide deployment, such as the scalability problem, the action definition problem in order to support arbitrary protocols, management problem and so on. As for packet processing, OpenFlow forwards the packets to specific external hardware for processing. This method is possible for trial networks, but for real networks, it suffers forwarding bottleneck and has poor scalability.

Before it can be used in real networks, problems such as the scalability and redundancy of the controller, and the secure channel bottleneck (similar to the internal channel bottleneck of a router), must be solved. Currently, there are several research in extensions and improvements of openflow such as OpenRouter[8], devoflow[12] and so on.

2.5.2 NetOpen

NetOpen [9] uses primitives as linkages between NetOpen networking services and programmable underlying network substrates. NetOpen also identifies the key features required in realizing required networking services and links the key features to the resources of programmable network substrates. Also, the NetOpen can use the programming substrates such as OpenFlow to build the SDN. Currently, the NetOpen networking is in its incipient stage. The next step is to elaborate the design and implementation of NetOpen networking services.

3. FUTURE RESEARCH TRENDS

Active networks have been proposed more than 10 years, but ultimately it has not been put into use due to significant change to the network equipment. Recent years, researchers changed their concept and avoid big change to the productive network equipment.

The consensuses from the current approaches for preparing programmable networks are mainly as follows: 1) The number of customized requirements and the network innovations are increasing amazingly. Predicting the future is a daunting challenge; our community has not fared well in the past. It's a good way to openup network interface to form a network ecosystem for researchers to develop and maintain protocols on their own without the long process of protocol standardization. 2) The future Internet needs diversity architectures to meet the various and unpredictable demands of future and to coexist in a complementary way with each other. The researchers here are not only includes the equipment vendors like Cisco and Juniper, but also includes other network researchers from academic, industry, government and so on.

4. FUTRUE KEY REASERCH ISSUES

No matter what kind of the approach it is, the basic design principle is to achieve high performance and high forwarding rate. Besides, the future research trends discussed in Section 3 need new features as follows.

4.1 Resource Management

The resource mainly refers to memory, storage, bandwidth, CPU, port, topology. The resource management includes resource abstractions, resource calculation, resource discovery, resource isolation, resource allocation, resource scheduling and so on.

In programmable networks, resource abstraction is an essential concept. For example, in SDN a well defined vendor agnostic interface set can make the control plane compatible to different data plane hardware from different equipment vendors.

4.2 Programmability Interface

Programming interface to the network infrastructure is another important issue of programmability. It ranges from node resource to complete programmable network resource. And the granularity or level of programmability is from the basic primitives, to service-level, and then to application-level, at last to the network level.

Another issue is related to deployment. Programming interfaces must enable the easy deployment and management for users to operate their private networks on the Internet.

4.3 Security

The future Internet needs a security and authentication mechanism for both the admission control and the isolation between customized networks. Besides, the programmable network gives rise to a new array of security vulnerabilities. This is a new challenge in programmable networks.

5. CONCLUSION

This paper described the ossification of the current Internet architecture, and gave a detailed analysis why many existing solutions to such ossification have not been incorporated into the Internet. Then, this paper gave an analysis and evaluation to several representative approaches for preparing programmable networks with structural changes to the current Internet architecture. At last, some possible key issues that the programmable networks would face were discussed. The detailed review of those programmable networks gave us a bright future that the ossification could be fended off and the evolution of the Internet could be accelerated. This paper is mainly based on a selection of representative approaches; the future work will give a more comprehensive and detail review.

6. ACKNOWLEDGMENTS

This research is supported by National Science Foundation of China under Grant 61073172.

7. REFERENCES

[1] Touch J, perlman R. "Transparent Interconnection of Lots of Links (TRILL): Problem and applicability statement[R]". IETF RFC 5556.2009.

[2] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research ", IEEE Commun. Mag., vol. 35, pp.80 - 86, 1997.

[3] 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), 2008.

[4] Junos SDK: http://www.juniper.net/as/en/products-services/junos-developer/junos-sdk/

[5] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris. "Resilient overlay networks", SIGOPS Operating Systems Review 35 (5) (2001) 131–145.

[6] PlanetLab: http://www.planet-lab.org/

[7] N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba. "A survey of network virtualization". Technical Report CS-2008-25, University of Waterloo, 2008.

[8] Tao Feng, Jun Bi, Hongyu Hu, "OpenRouter: OpenFlow Extension and Implementation Based on a Commercial Router", in proceedings of the 19th IEEE International Conference on Network Protocols (ICNP11), Vancouver, Canada, 2011.

[9] N. Kim and J. Kim. "Building netopen networking services over open flow-based programmable networks". In Proc. of ICOIN'11, Jan. 2011.

[10] XORP : http://www.xorp.org/

[11] GENI: http://www.geni.net/

[12] Andrew R. Curtis, Jeffrey C. Mogul, Jean Tourrilhes, Praveen Yalagandula, Puneet Sharma, Sujata Banerjee. "DevoFlow: scaling flow management for high-performance networks". SIGCOMM 2011:254-265.

[13] The Click Modular Router Project: http://www.read.cs.ucla.edu/click/

[14] N. McKeown. Keynote talk: Software-defined networking. In Proc. of IEEE INFOCOM'09, Apr.2009.

[15] Cisco: http://www.cisco.com/web/CN/products/products_netsol/switches/products/ca7000/nexus_7000_virtualization.html?pnBAC=OTC-CC5OB001RF01

[16] Juniper: http://www.juniper.net/techpubs/software/junos/junos85/feature-guide-85/id-11139212.html