Architecture for Core Networks Utilizing Software Defined Networking

Pavol Helebrandt*
Institute of Computer Engineering and Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
pavol.helebrandt@stuba.sk

Abstract

New and popular approach to computer network architecture - Software Defined Networking aims to programmatically and centrally control the whole network providing many advantages. However, deployment of SDN in large scale networks of telco operators and service providers is limited due to lack of standardized communication between SDN controllers and use of routing algorithms of traditional networks.

In this dissertation we provide analysis of SDN principles, existing solutions and methods to scale their performance for large scale networks. Based on the analysis we formulate problem of SDN domain interconnection for east-west communication. To solve this problem, we propose a new architecture for interconnection of controllers in various SDN domains called INT Architecture. INT Architecture is formally verified by modelling in Petri Nets and practical tests of INT Architecture prototype using virtual machines. INT Architecture is beneficial enhancement of SDN enabling greater cooperation of SDN controllers and applications in large scale multi-domain networks.

Categories and Subject Descriptors

C.2.1 [**Computer-communication networks**]: Network Architecture and Design; C.2.2 [**Computer-communication networks**]: Network Protocols; C.2.3 [**Computer-communication networks**]: Network Operations

*Recommended by thesis supervisor: Assoc. Prof. Ivan Kotuliak
Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on August 25, 2016.

Helebrandt, P. Architecture for Core Networks Utilizing Software Defined Networking. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 8, No. 2 (2016) 56-61

Keywords

Software defined networking, Wide area networks, Internet, Scalability, Network management, SDN multi-domain, Interconnect, SDN peering

1. Introduction

Internet traffic keeps growing because of better access and rising popularity of multimedia and P2P, while new and innovative applications require even more resources and better network parameters. Telco carrier networks nowadays are mostly based on Ethernet and Multiprotocol Label Switching (MPLS) solutions, some being upgraded with Shortest Path Bridging (SPB) and Generalized MPLS (GMPLS). Routing between these carriers Autonomous Systems (AS) is performed by Internet global routing table and Border Gateway Protocol (BGP).

Software Defined Networking [14] is a new approach to networking that aims to programmatically control the whole network from a logically centralized node. While research in the area of Software Defined Networking is in the centre of attention and improving the controller performance is just getting into the spotlight, there is very little research into the advantages of interconnection of various controllers in heterogeneous SDN domains.

Motivation for this project is to facilitate deployment of SDN and new innovative applications in large scale networks and enable increase of SDN deployment among ISPs. In this paper we introduce new architecture for interconnection of control planes and SDN applications. The interconnection system can be deployed in various SDN domains and enable control plane communication and coordination for better provisioning of services across multiple domains. To accomplish this, we propose a new interface for SDN controllers for interconnection of domains using a new vendor neutral communication protocol.

The rest of this paper is organized as follows: state of the art of SDN and scaling of SDN control plane is described in Section 2. Section 3 identifies problem statement with premises, thesis and goals of this paper. Section 4 introduces INT Architecture proposal for interconnection and cooperation between SDN controllers in multi-domain environment to enhance SDN. Section 5 presents methods used for formal verification of proposed INT Architecture. Section 6 concludes this paper by summarizing results and benefits of INT Architecture proposal.

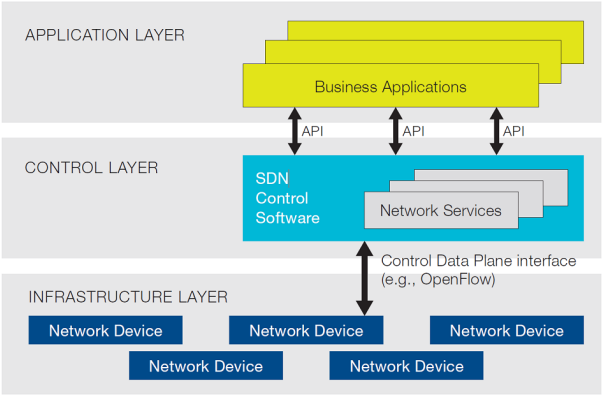

Figure 1: SDN Architecture [8].

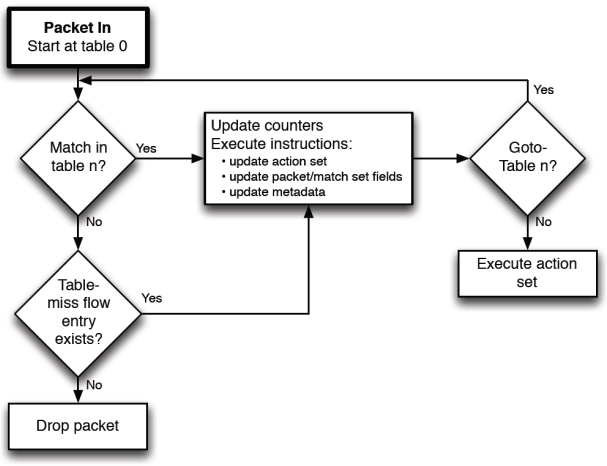

Figure 2: OpenFlow switch packet handling [1].

2. The State of the Art

In this section we analyse SDN and its potential benefits and limitations. Furthermore, we investigate problem of scaling the SDN control plane for deployment in large networks and their interconnection.

2.1 Software Defined Networking

Software Defined Networking is a novel paradigm in computer network architecture with aim to control all network nodes with programme. This enables solving of many problems in traditional approaches to networking while also enabling new features as well. The general drive behind SDN is to increase flexibility, manageability and extensibility of computer networks, with secondary goal of decreasing equipment costs. This is achieved by taking advantage of fast development and deployment cycle of relatively cheap software applications in contrast to expensive specialized networking hardware.

Functionality of networking equipment can be conceptually divided into switching of traffic data between interfaces - Data Forwarding plane; and Control plane - rules created by processor running operating system, routing algorithms, address translation, and other higher functions. In traditional networking, both control plane and data forwarding plane are implemented in every network node, often using specialized hardware. This enables every device to be totally autonomous and make all high level decisions, such as packet routing independently.

The fundamental principle of SDN architecture depicted in Figure 1, is separation of control and data forwarding planes that communicate over standard interface. By implementing of separated control plane by software for general purpose computer from forwarding plane on network equipment, it is possible to centralize decisions as well as configuration of all network devices. Using centralized view of the whole network enables high level decisions about traffic management and computations to be made only once, results then propagated to be used by all nodes in data forwarding plane. Furthermore, the centralized control plane implemented in software executed on general purpose processors can bring many advantages to networking - especially speeding up innovation, new network features development and deployment. While there are many approaches to SDN, the main enabler of SDN and de facto standard communication protocol is OpenFlow.

2.1.1 OpenFlow

OpenFlow [13] is an open standard originally developed at universities and currently maintained by Open Network Foundation (ONF) - non-profit consortium with mission to commercialize and promote OpenFlow based SDN. The OpenFlow switch standard [1] defines communication interface between control plane and forwarding plane devices and so must be implemented by both sides.

OpenFlow Controller makes high level switching decisions in control plane formulating them into forwarding rules, composed of matches and actions. These rules are entries for Flow Tables used by OpenFlow switch in forwarding plane to handle incoming packets.

When a packet is received by OpenFlow enabled switch, it is handled in OpenFlow pipeline composed of one or more Flow Tables, each containing entries with rules and actions to be performed on the packet belonging to flow. If match for the packet is not found in any Flow Table and rule to send unknown packets to Controller is set-up, it is sent to the controller. Controller processes the packet and either drops the packet or establishes a new flow, by creating a new entry in Flow Tables. The handling mechanism of a received packet inside the OpenFlow switch is charted in Figure 2.

2.1.2 Alternative and Related Technologies

Although OpenFlow is currently the most popular approach, SDN is an extensive discipline in constant flux and many novel viewpoints to SDN implementation exist and continue to be developed. Furthermore, there are technologies that can be considered partial SDN enablers and in addition to other projects that can greatly benefit from deployment together with SDN.

From commercial solutions, such as Cisco ONE and Nuage Virtualized Services Platform aimed at providing some programmability for devices from these vendors, to various open solutions. Among these are Interface to the Routing System (I2RS) [2], ForCES [5], NETCONF [6] and PCEP [18]. While I2RS is a new and ambitious approach with goal to provide standard unified interface to routing system for control and management; ForCES, NETCONF and PCEP are older technologies repurposed

for use in SDN. Especially PCEP in combination with MPLS and its multiple extensions are seen as a gradual migration path towards SDN without network disruption and maintains existing interoperability, which is very important factor to telco operators. As such PCE can be considered as an evolutionary path towards SDN using already deployed traditional network equipment, while OpenFlow is revolutionary.

Network Function Virtualization (NFV) [7] is a carrier-driven initiative to virtualize network functions and migrate them from purpose-built devices to generic servers. While SDN and NFV both cover similar themes and can benefit from one another, they are independent and do not require deployment of the other in network.

2.2 Scaling SDN Control Plane

This section analyses the problem of scaling the SDN control plane in large scale networks, i.e. the Internet, and challenges in SDN controller interconnection.

Scale has been an active and often contentious topic in the discourse around SDN for a long time. Criticism of the SDN paradigm argues that changing the control plane implementation model from anything but full distribution will lead to scalability challenges. Furthermore, there is the common concern that questions the scalability of using traditional SDN, i.e. OpenFlow, to control physical switches due to forwarding table limits.

In theory, any SDN approach can have the same scaling properties as traditional networking. For example, there is no reason that controllers cannot run traditional routing protocols between them. However, scaling properties of a system built using an SDN approach that actually benefits from the architecture, and scaling properties of an SDN system different from the traditional networks is much more interesting endeavour.

There are various approaches to scaling SDNs that are currently in different stages of development and/or deployment. These can be classified into two categories. The first method scales up the performance of single SDN controller with increased optimization and parallelization of execution, such as NOX_MT [17] and Maestro [3]. There are also hybrid solutions using physically distributed controllers in clusters with multiple instances of single logical controller ElastiCon [4], xBar [12], HyperFlow [16], Onix [10]. However, these are designed for single domain use only.

Second approach scales out by deployment of multiple interconnected controllers that communicate for cooperation. Additionally, there is also the aspect of whether the interconnected SDN controllers are all in the same domain or some of them are in different domains. To clarify meaning of interface orientation and terms used in this paper, we provide illustration of their position in Figure 3.

Performance of controllers is becoming more important to developers as increasing number of OpenFlow controllers are being both developed and deployed. However, a single physical controller, albeit a high performance one, is not enough to manage a sizeable network. High availability and maintaining low response times are among the critical reasons why a network needs multiple controllers.

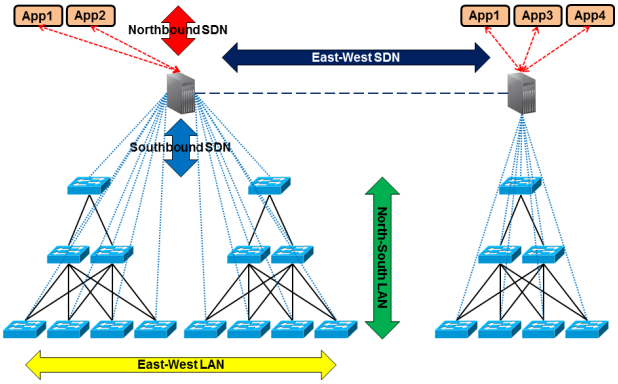

Figure 3: SDN interface orientation compass.

SDN interconnect (SDNi) was among the first to deal with connecting SDN domains using an automated system. SDNi draft [19] proposes an open protocol SDNi for the interface between Software Defined Networking domains to exchange information between the domain SDN Controllers. However, this draft expired in 2012 and was abandoned with no further work.

Another approach to interconnection of SDN controllers is East-West Bridge (EWBridge) [11], which is still in development. EWBridge proposes a design for high performance communication system between heterogeneous Network Operating Systems and partition large telco operator network domain into subnetworks.

3. Problem Statement

Although there are projects to scale or distribute the SDN controller functions to better accommodate a large network with several thousands of active nodes, very little work was done on interconnecting controllers of such large networks and leveraging advantages of SDN.

At the moment it is very difficult to deploy SDN architecture in very large scale networks - i.e. the Internet - and utilize its benefits, because of lacking effective method for large scale controller distribution. While SDN paradigm is getting traction in data centres and campuses, that can be large networks with several thousands of nodes, these are typically managed by a single controller. However, controllers in these SDN islands are using traditional network protocols like BGP to exchange only routing information between domains. This is a limiting factor for usefulness and flexibility that could be provided by completely SDN based networks. Even though remote access and manual alteration of controllers in partners' network is possible to some extent, this method is not feasible for the Internet and is antithesis to SDN principle of network management automation.

Thesis of this dissertation is proposing architecture for interconnection of controllers in various SDN domains to communicate and exchange information for better provisioning of services with greater flexibility across different network domains. From the thesis arise these partial objectives:

- Improve SDN architecture to benefit from east-west interface between controllers

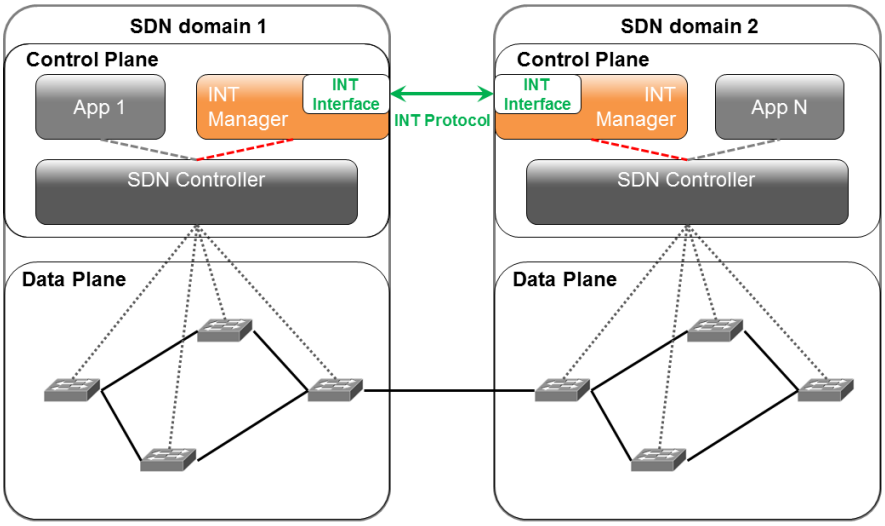

Figure 4: INT Architecture.

- Design new universal east-west communication protocol for interconnection of heterogeneous SDN networks
- Define a communication interface in SDN controller for the interconnection protocol
- Verify the functionality of the designed protocol and methods by comparing it with unmodified network and alternative existing methods

4. INT Architecture

To achieve goals defined in problem statement, a new interconnection architecture is necessary. In this section we describe design of INT Architecture for interconnection of SDN controllers. We define communication protocol for SDN controllers, functions and interfaces used, as well as formal model of communication protocol. The high level architecture of the proposed system interconnecting two SDN domains is depicted in Figure 4.

Let us have n SDN domains $[SDN_1, SDN_2, \ldots, SDN_n]$ composed each of exactly one Controller $[CNT_i$ belonging to domain $SDN_i]$ and set of forwarders $[FWD_{i1}, FWD_{i2}, \ldots, FWD_{ij}]$ belonging to domain SDN_i. SDN domain is part of SDN network that is managed by single logical SDN controller CNT_i, although it can be implemented as collection of multiple physical controllers. One SDN domain is administered by a single organization and can be thought of as similar to Autonomous System in BGP.

Currently any domains SDN_a and SDN_b are interconnected only with traditional routing protocols, e.g. BGP to provide IP connectivity in data forwarding plane. We propose to replace traditional routing protocols with INT Architecture for interconnection of domains SDN_a and SDN_b in control plane to leverage advantages of SDN applications across these domains. Interconnection of domains SDN_a and SDN_b is managed by INT Manager application MNG_i of controller CNT_i for each domain SDN_i. INT Manager MNG_i controls all interconnections and configuration of INT Interface IF_i for handling the message exchange between controllers. Connection of control planes itself is created between INT Interfaces IF_a and IF_b that are components of MNG_a and MNG_b in domains SDN_a and SDN_b respectively. INT Manager applications MNG_a and MNG_b control operation of data forwarding plane connection between edge forwarders FWD_{ai} and FWD_{bj} linking domains SDN_a and SDN_b.

4.1 INT Architecture Functions

Main functions or components of the INT Architecture are INT Manager and INT Interface.

The **INT Manager** function is responsible for interpreting the network topology - data plane routers and their links managed by a SDN controller, into a virtual router. This virtual router presents all networks in the controller domain and networks reachable by it, with various path metrics for all of them. It is further responsible for setting up connections to other controllers and management of existing sessions. It also provides interface for domain administrator to manage peer connection setup, session management and advertised domain SDN and/or NFV capabilities.

For the INT Architecture to be useable in heterogeneous environment with various SDN controllers, INT Manager needs to use appropriate API for Northbound communication with a given controller. Every controller uses slightly different data structures to store the information about its network topology and traffic data, but most provide common Northbound interface to access this data.

While the INT Manager compiles network topology data and presents administration point for interconnection session management, **INT Interface** is responsible for handling the communication between INT Managers of different SDN controllers. This communication can be classified into two categories:

- **Intra-domain** - between controllers in the same administrative domain, often managing the same network in cooperation to provide higher processing power and/or controller redundancy. OpenFlow switch specification has already defined mechanisms to support connection to multiple controllers, but controller cooperation remains unstandardized.

- **Inter-domain** - between controllers in different administrative domains is more substantial to adoption of SDN in large scale networks. While routing between different networks is possible using existing routing protocols - BGP for linking AS and various Interior Gateway Protocols for networks inside an AS - it negates the advantages that can be leveraged by using SDN inside the networks being linked.

For inter-domain communication one of the controllers in the domain is selected to be the "master controller" to communicate with peer domains and represent the focal point of the SDN domain control plane. It provides logical single point for peer controllers outside the domain to concentrate and disseminate information to controllers inside the domain and vice-versa. This minimizes the need for connections between all controllers and as such functions similarly to route reflector in BGP or designated router in OSPF. Both these types of communication between SDN controllers is achieved over INT Interface using the INT Protocol explained in the following section.

4.2 INT Protocol

The INT Protocol enables both intra-domain and inter-domain path setup and exchange of information between controllers about their capabilities. This protocol further provides not only scalability features for controllers in single administrative domain, but also various levels of network topology abstraction and control for peer controllers in separate SDN domains. INT Manager functions of different SDN controllers are connected and communicate using INT protocol over TLS or plain TCP session. The protocol itself is partly inspired by existing protocols such as CDNi and BGP and is composed of three sub-layers:

- **INT Session Management** - used for peer connection setup and session management. Connection establishment inside the administrative domain should be automated to minimize administration overhead, but require manual setup for inter-domain for security and purposes,
- **Capabilities Information Exchange** - responsible for exchange of information about domain capabilities and networks available inside and through the domain together with path metrics,
- **Path Setup** - used for end-to-end flow path setup between client nodes in all the domains along the traffic route according to path metrics requirements of the original domain controller.

Interconnection of SDN controllers in single administrative domain is relatively straightforward process. Since controllers manage parts of the same network and there are no restrictions on shared data, little to no manual administration is needed for interconnect session to be initiated and network information shared between controllers. Creation of interconnection session and path setup between two peering controllers using the INT Protocol is illustrated by message flow in Figure 5.

Interconnecting controllers in different domains is more difficult because of added technical intricacies stemming from typically unsafe connection, distinct management

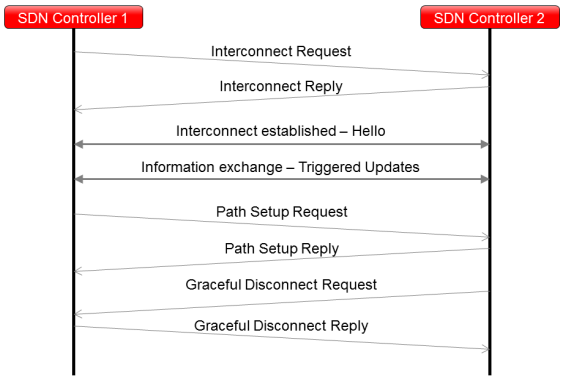

Figure 5: INT Protocol.

and security policies. Furthermore, SDN domains understood in the form of traditional AS are operated by independent organizations with different levels of trust and involvement of agreements and contracts between separate legal entities.

5. Verification

Verification of proposed INT Architecture that can be split into two parts. Firstly, formal verification of INT Architecture design correctness with use of mathematical modelling. Secondly, practical tests using prototype implementation of INT Architecture in virtual environment used for testing.

5.1 Methodology for Formal Verification

Petri Nets (PN) are a mathematical instrument well suited for modelling of discrete event systems. Graphical representation of Petri Net is a bipartite directed multigraph, as defined by Petersen [15]. Bipartite because vertices can be divided into two disjoint groups - conditions and tasks that are connected by arcs. Every edge connects a place to a transition or vice versa, as can be seen in Figure 6. No edge can be between two places or two transitions. Conditions (or places, states) are graphically represented by circles, tasks (or transitions) by bars, and arcs by directed edges. Tokens placed in places define state of the Petri Net, also called marking.

Using formal definition, Petri net is a five-tuple $PN = (P, T, FW, M0)$, where:

- $P = \{p_1, \ldots, p_n\}$ is finite nonempty set of places,
- $T = \{t_1, \ldots, t_m\}$ is finite nonempty set of transitions,
- $P \cap T = \varnothing, P \cup T = \varnothing$,

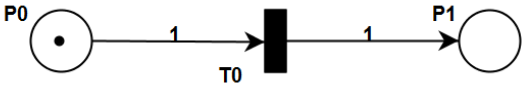

Figure 6: Petri Net example.

- $F \subseteq (P \times T) \cup (T \times P)$ is set of arcs,
- $W : F \to (Z^+)$ is weight function,
- $M_0 : P \to (Z^+ \cup \varnothing)$ is the initial marking.

Petri Nets are distinguished by the fact that thanks to their mathematical model, we can investigate their various properties and by extension protocol properties. According to [9], properties of Petri nets important for communication protocol model validation are:

- **Reachability** - any marking M_n is reachable from initial marking M_0 if there is firing sequence of transitions $t_1, \ldots, t_k$, where $M_0, t_1, M_1, t_2, M_2, \ldots, M_{n-1}, t_k, M_n$,
- **Reversibility** - property of PN, when initial marking of PN is reachable M_0 after firing finite number of transitions $t_1, \ldots, t_k$,
- **Boundedness** - PN is k-bounded, or simply bounded, if the number of tokens in each place does not exceed a finite number k for any marking reachable from the initial marking M_0. PN is called safe if it is 1-bounded,
- **Liveness** - if for every PN marking M reachable from the initial marking M_0, there exists fireable transition t that leads to different marking M', meaning there are no deadlocks.

By using Petri Nets for modelling of communication protocol and investigating these selected properties of the model, we can determine behaviour of the modelled protocol and correctness of its design.

6. Conclusions

Increasing and network management complexity makes concept of centrally controlled and programmable SDN very appealing. On the other hand, scaling SDN control plane for large networks has been an active and often contentious topic. Criticism of the SDN paradigm argues that changing the control plane implementation model from anything but full distribution of traditional networks will lead to scalability challenges.

There are projects to scale or distribute the SDN controller functions to better accommodate a large network with several thousands of active nodes. However, current SDN architecture is limited in leveraging most of benefits it offers in large scale interconnected networks by lack of standardized communication between controllers.

To solve this problem, we proposed INT Architecture to improve SDN for interconnection and cooperation between SDN controllers in multi-domain environment. INT Architecture includes INT Manager and INT Interface functions, together with extensible INT Protocol for standardizes communication between various heterogeneous SDN controllers.

Acknowledgements. This work was partially supported by the Slovak National Grant agency VEGA 1/0676/12, NTB 2012et011 and STU Grant for Young Researchers 2016.

References

[1] OpenFlow switch specification version 1.3.1. 2012.

[2] A. Atlas, T. Nadeau, and D. Ward. Interface to the Routing System Problem Statement. *IETF, draft-atlas-i2rs-problem-statement-01, work in progress*, 2013.

[3] Z. Cai. *Maestro: Achieving scalability and coordination in centralizaed network control plane*. PhD thesis, Rice University, 2012.

[4] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella. Towards an elastic distributed SDN controller. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 7–12. ACM, 2013.

[5] A. Doria, J. H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. Forwarding and control element separation (ForCES) protocol specification. Technical report, 2010.

[6] R. Enns, M. Bjorklund, and J. Schoenwaelder. Network configuration protocol (NETCONF). *Network*, 2011.

[7] ETSI. Network Functions Virtualisation: Architectural Framework. Technical report, Technical Report ETSI GS NFV 002 v1. 1.1, 2013.

[8] O. N. Fundation. Software-Defined Networking: The New Norm for Networks. *ONF White Paper*, 2012.

[9] G. Holzmann. *Design and validation of computer protocols*. Prentice Hall, 1990.

[10] 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. In *OSDI*, volume 10, pages 1–6, 2010.

[11] P. Lin, J. Bi, and Y. Wang. East-West Bridge for SDN Network Peering. In *Frontiers in Internet Technologies*, pages 170–181. Springer, 2013.

[12] J. McCauley, A. Panda, M. Casado, T. Koponen, and S. Shenker. Extending SDN to large-scale networks. *Open Networking Summit*, pages 1–2, 2013.

[13] 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.

[14] T. D. Nadeau and K. Gray. *SDN: Software Defined Networks*. O'Reilly Media, Inc., 2013.

[15] J. L. Peterson. *Petri net theory and the modeling of systems*. Prentice Hall PTR, 1981.

[16] 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, 2010.

[17] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood. On Controller Performance in Software-Defined Networks. In *Presented as part of the 2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, 2012.

[18] J. Vasseur and J. Le Roux. Path computation element (PCE) communication protocol (PCEP). 2009.

[19] H. Yin, H. Xie, T. Tsou, D. Lopez, P. Aranda, and R. Sidi. Sdni: A message exchange protocol for software defined networks (sdns) across multiple domains. *IETF draft, work in progress*, 2012.

Selected Papers by the Author

P. Helebrandt, I, Kotuliak. Novel SDN multi-domain architecture. In *Proceedings of IEEE 12th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2014, pages 139-143.

P. Truchly, P. Helebrandt, L. Danielovic. Implementation and Evaluation of IPv6 to IPv4 Transition Mechanisms in Network Simulator 3. In *Proceedings of IEEE 23rd International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2016, In print.