SDN/legacy Hybrid Network Control System

Feliksas Kuliesius and Mindaugas Giedraitis
Faculty of Physics
Vilnius University
Vilnius, Lithuania
feliksas.kuliesius@ff.vu.lt

***Abstract*—**The widespread replacement of traditional network to SDN is still restricted in the enterprise environment, and hybrid network where SDN and legacy network appliances coexist by necessity complementing each other is deployed usually. To answer the network parts integration issues, the hybrid network control system is developed and discussed in the present work.

Thorough network topology assessment (BDDP complemented with LLDP/SNMP based legacy network discovery) and the possibility of converting SDN flow rules into the configurations of legacy appliances enables to orchestrate hybrid network as well as to enforce the network security.

Keywords—Software defined networking, Hybrid SDN, Computer network management, Topology discovery

I. INTRODUCTION

The paradigm of Software Defined Networking (SDN) is introduced into the networking world for years [1-4], but the implementation is still limited to large Data Centers with their particular architecture [5,6], traffic shaping when inter-connecting WAN [7], Internet Exchange Points [8-10], and other special cases.

One of the key concepts of SDN which result in additional functionality is the separation of the data plane from the control plane. Opposed to traditional (legacy) IP networks, where appliances perform both packet forwarding (data plane) and run control protocols which discover network paths, make routing and other L2/L3 filtering decisions (control plane), in SDN, control functions are separated from the forwarding elements (routers and switches), and concentrated in a logically centralized unit called the SDN controller [11,12]. Centralized network view allows packets forwarding and other traffic engineering tasks solve not solely on the ground of destination addresses and shortest path principles, as forwarding can be defined in terms of layer-2, layer-3 and even layer-4 header fields, including source, destination, priority and other. Solving security, scalability and performance tasks the controller can be implemented as a set of physically distributed elements [13]. Since the most common SDN protocol (other protocols suggested as the southbound interface are BGP, PCEP, SNMP, etc. [14]) connecting control and data planes is OpenFlow (OF), the terms SDN and OF usually are used interchangeably.

Despite the evident advantages and new feature set that is provided by SDN due to centralized, sophisticated control and technology that allows much more packet's attributes (fields) and actions to be introduced for flow identification and processing than IP protocols can, the use of SDN is very limited in the enterprise network environment [15-17]. Larger deployment of SDN technology in the networks is suspended due to several reasons. Some of the networking services well developed and robust in legacy structures are lack of support in SDN. So, additionally to difficulties and impossibility of simultaneous replacement of legacy switches with OpenFlow switches, related to the technical, timing and financial issues, legacy switches frequently can't be substituted since there is no equivalent OF enabled equipment supporting required services or legacy devices are very critical for the production network.

The enterprises are reserved with the introduction the young technology as well as have a fear of stoppage during the transition to SDN. Possibly, one of the most important reasons is the limited budget for new network infrastructure. The promising solution to moderate these limitations is to gradually install a small number of SDN-enabled components [18.19] combined with the traditional (legacy) network devices; thus, incrementally replacing traditional network boxes by SDN devices. A network containing a combination of SDN and legacy network devices is commonly denoted as a hybrid SDN network [16,17].

The hybrid network can be organized by deploying dual-stack (i.e. the same device supports both legacies and OF protocols) switches or separate Legacy and SDN areas. The former one requires a contiguous installation of hybrid programmable switches capable of handling packets according to both legacy and SDN mechanisms rather than an integration of legacy hardware and approaching the resulting transitional network to SDN gradually which is the paradigm of the second one, allowing incremental partial insertion of SDN switches interoperating with existing legacy devices. The strategy of inserting SDN devices can be different, subject on the objectives to be achieved. SDN switches can be partially implemented in access [20,21], core [22,23] layers, in centrality points [19] or other ways. The control system of such hybrid networks depends on the network goals and the deployment strategy preferred, also.

The hybrid network control system introduced in the present work is based on the complete network topology discovery and the possibility of converting OF rules into the configurations of legacy devices. Trading on complete network visibility the controller can support every known high-performance legacy procedures, e.g. based on shortest path network technics [24], and additionally complemented with direct influence to legacy devices.

II. MANAGEMENT OF HYBRID NETWORKS: STATE OF THE ART

Since it is practically impossible to replace an entire enterprise legacy network to SDN, partially solutions i.e. incremental updates usually applied. As a result, network operators can increase automation and network control, allowing them to build flexible and programmable networks implementing partial equipment replacement only. Unsurprisingly, control plane often is loaded heavier in this case since it must additionally process traffic via the legacy network. Different applications and intermediate devices usually are used to translate SDN forwarding rules (flow tables) entries into legacy switch configurations, such as HybNet [25], LegacyFlow [26,27], Closedflow [28]. Additional hardware devices can be used to expand legacy switch capabilities [29].

Another approach is based on the injection of fake but harmless information into the network which indirectly affects L2 and/or L3 filtering/routing. So, the controller manages traditional switches indirectly by sending seed packets with the intention to change MAC or route tables pursuing to change flow filtering rules. Telekinesis [30] and Magneto [31] (both of them generate fake MAC addresses) operate at layer 2 in networks consisting of legacy and OpenFlow devices and divert flows through SDN switches which provide more fine-grained legacy path control. Fibbing [32,33] works in layer 3 legacy networks consisting of only legacy routers. To improve routing flexibility while ensuring loop-free paths it injects fake network topologies into link-state routing.

The very effective solution both in the sense of performance and deployment simplicity is proposed as Shear project [19]. Leveraging spanning tree protocol (common for legacy switches) as shortest path tool, SHEAR applies the OpenFlow switches as supervisory points and breakers for the decomposition of the physical network into loop-free components, enabling a fast response to topology changes and failures, and increasing path diversity. In the legacy network part, the slicing is organized using VLANs which are mapped one VLAN to one particular end node, forming broadcast segments (pathless) ending at OF switches and defined by one STP domain. Basically, SHEAR does not constrain routing through middle-boxes or access control units however, its deployment strategy can also be applied to enforce traffic to pass through SDN switches which can be used as centrally controlled nodes.

In the present work, moreover to technics discussed above the additional functionality to the control system is introduced and discussed. The presented system consists of two main modules: HybN-Topo responsible for entire hybrid network topology discovery and HybN-C, responsible for commanding legacy appliances from the controller. It enables comprehensive network topology and device status gathering and adjusting legacy network configurations to centralized SDN logic.

III. DEPLOYMENT OF THE HYBRID NETWORK CONTROLLER SYSTEM

The model proposed must answer the two main concerns discussed earlier. The first one is how to make the legacy and SDN networks integrated into the view of the controller, and how to get the correct status of the network devices so that the controller can make the correct forwarding decision. The second one is how to include legacy appliances into a common operating system under SDN control. The deployed hybrid network control system consists of two modules: HybN-Topo responsible for topology discovery and HybN-C responsible for legacy network control.

A. Topology discovery

Topology discovery is one of the most important services provided by the controller and it is the basis for the normal operation of the network [34]. Since a centralized view of the network topology holds the key point for SDN operation, it has made considerable attempts in the area by the research community in the past few years [35-38]. As for layer 2 discovery protocol, OpenFlow Discovery Protocol (OFDP) is used in a pure SDN network. OFDP uses LLDP [39] as a base: the controller periodically commands SDN switches to send LLDP messages, and the neighboring SDN switches will append metadata and forward them back to the controller. In the OFDP v2 [36], the controller commands per switch interface were replaced by one command per switch, initiating to flood LLDP overall switch interfaces. In sOFDP, an OFDP version with included security functions was introduced [38].

The main limitation of both OpenFlow Discovery Protocol (OFDP) and OFDP v2 [34] is that they operate and discover topology links between adjacent OF switches and operate in entirely SDN environment only. OFDP frames as usual LLDP ones (both they are of link-local significance) are dropped by legacy switches inserted in SDN structure and does not reach neighbor OF switch which could pass the message to the controller. It is a serious issue in hybrid legacy/SDN networks. Modified discovery approach based on broadcast messages [37] overcome the issue and controller can collect information about the links between two ports of OF switches in the same broadcast domain even they are connected indirectly with legacy switch includes. Broadcast Domain Discovery Protocol (BDDP) has the same structure as the LLDP/OFDP except that destination multicast address is changed to broadcast one and protocol field is changed from 0x88CC to 0x8999. Never-the-less, even in this case the controller also gathers only SDN devices information, since legacy switches are seen as fully transparent by BDDP. Additionally, legacy devices do not support BDDP protocol and neither can be activated by nor report to the controller. Legacy switches do not register to SDN controller as OF devices do, so the controller does know nothing (ID, capabilities, location…) about them even if they exist.

B. Our contribution

To overcome the before mentioned issues, HybN-Topo application was developed in the present work. In our test-bed, this application reveals legacy network topology on the SMNP/LLDP basis and incorporate it into OF device (which is possible to call native OF) topology, validated by BDDP and stored in controller storage module, since it is very important to achieve the unified view of SDN and legacy network in the controller and to offer the controller and network administrator

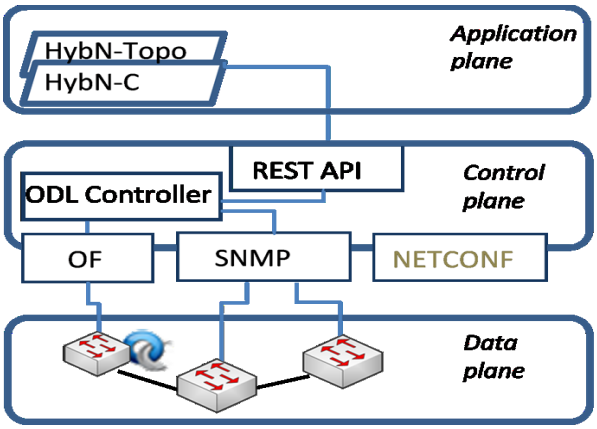

Fig. 1. The architecture of hybrid network control

the centralized control over both legacy and SDN network segments.

The logical design of the control system is shown in Fig.1. The topology discovery module was designed as an external application communicating to the controller via open REST API (representational state transfer application program interface) services. Such approach enforces to fit any possible controller which support REST interface. The hybrid network control testbed was developed on the basis of the OpenDaylight (ODL) controller, though external architecture does not bind the application to a particular controller and is absolutely portable. The only existing restriction is the possibility to collect SNMP information – OpenDaylight has SNMP southbound plugin SNMP4SDN [40] and ODL model-driven service abstraction layer was exploited for information exchange between program modules in our deployment. Installing the application with the controller without SNMP module, a separate SNMP server should be used and inter-device communication should be tuned.

The topology discovery is presented in Algorithm 1. The topology related to SDN devices is retrieved from ODL controller internal database topology storage, accessible by

ALGORITHM 1. PSEUDOCODE OF THE TOPOLOGY DISCOVERY ALGORITHM IN A HYBRID NETWORK ENVIRONMENT.

```
topology = new Topology();

# Calculate OpenFlow topology
openFlowInformation[] = topologyRequestREST();
FOREACH openFlowInformation[„node information“] as nodeInformation
    topology->addNode(getOpenFlowNodeInformation(nodeInformation));
END

FOREACH openFlowInformation[„link information“] as linkInformation
    # Add interface and link to another interface
    IF linkNotExist(linkInformation)
        topology->addInterface(getOpenFlowLinkInformation(linkInformation));
        topology->addLink(getOpenFlowLinkInformation(linkInformation));
    END
END

# Calculate legacy device topology
legacyDeviceInformation[] = getLegacyDeviceList();
FOREACH legacyDeviceInformation as deviceInformation
    nodeInformation = sendSnmpRequest(deviceInformation[„IP address“]);
    topology->addNode(nodeoInformation[„node ID“]);
    nodeLinkInformation = nodeInformation[„link information“];

    FOREACH nodeLinkInformation as linkInformation
        topology->addInterface(linkInformation);
        IF remoteInterfaceExist(linkInformation[„remote interface“])
            # Check if there is OpenFlow links with legacy device inserts
            IF remoteInterfaceHasLink(linkInformation[„remote interface“])
                topology-> addLinkRewritingExisting(linkInformation);
            ELSE
                topology-> addLink(linkInformation);
            END
        END
    END
END

RETURN topology;
```

HybN-Topo application. The ODL DB is filled by traditional SDN discovery technique: the registered OF switches are polled by a controller which sends BDDP orders to switches and gets BDDP with associated metadata from their neighbors.

Additionally, to reveal information about legacy devices, the topology discovery application (communicating via REST to the controller and using controller SNMP SAL capabilities) generates SNMP requests to registered devices and/or gets SNMP trap messages. To register switches to the controller they must be preconfigured with SNMP IP address and authentication data. The key SNMP MIB used by the discussed topology discovery system is related to the LLDP information, device ID, capabilities and interface information such as interface ID, state and load. Additionally, port load values sent to the server from the SNMP MIBs are leveraged as additional information which allows revealing the rogue infrastructure nodes. If the port statistics show activity of the port but the LLDP discovery did not find the neighbor on this active port (due to misconfiguration or intentional blockage of LLDP) the application should activate the port shutdown and log entry This means that if flow is detected on the neighboring device interface, the rogue switch should be discovered even it should behave passively in the listening mode. So, no one untrusted device can be hidden in the network set-up.

In short, the HybN-Topo utility is capable to obtain topology and device state information from legacy network part, include and adjust it into the common SDN view, reflect the various network events and legacy device status as well as notify the administrator about the suspicious behavior of the network or block some malicious flows.

The topology discovery illustrations are presented in Fig. 2. The examples of larger topologies are not included in the picture since the readability of notations deteriorated. The test-bed evaluation set-up was deployed using HP-2920 switches (blue ellipses in visualization plot) as SDN devices and Cisco Catalyst 3750 (red ellipses) as legacy appliances. The physical layout was extended by virtual network generated by Mininet [41]. LLDP chassisId was shown as a device identifier. The active interfaces are represented as a link from the device with an interface number. The link can connect the neighbor device interface or not and be represented as a broken one.

The time of topology discovery measured for different topologies varies from one-tenth to some seconds depending on topology size and legacy/SDN switch layout. The times are fully acceptable for network control operation. It is worth to note that measured times are the times of initial topology discovery and sporadic changes which happen afterward are captured and reported to the application by SNMP traps and processed shortly. As compared topology discovery times in our test-bed (<10 s) and in totally legacy network (~103-104 s) [42] (reported as the satisfactory result for network management), the lower time we got due to both the lesser network extent and more optimized SNMP MIB.

C. *Hybrid network control*

The second application in our framework is HybN-C (see Fig.1) which is devoted to integrated network control. It is

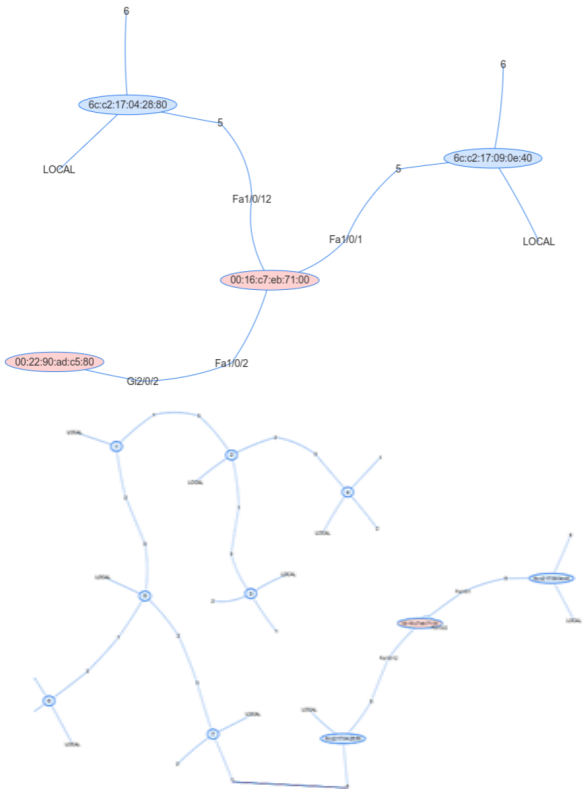

Fig. 2. Examples of hybrid network visualization. SDN HP2920 (blue ellipses), legacy Cisco 3750 (red ellipses) and virtual (circles) switches.

minimized as compared to [43] with an eye to ensure network performance and simplicity. As for the first application (topology discovery) the ODL model-driven service abstraction layer was exploited for information exchange between program modules. The built-in SNMP plugin was used to communicate and configure the legacy device. NETCONF southbound interface can be used to address legacy devices, though SDN and many of legacy device NETCONF standards differ from each other and therefore NETCONF wasn't used in our work.

Depending on the goals, entirely the simple SHEAR [19] like path control and STP based legacy set-up redundancy management can be used with additional thorough assurance over network topology and status. Moreover, complete network topology assessment and the possibility of converting SDN flow rules into the configurations of legacy appliances enables to adjust network slicing on the fly to orchestrate hybrid network as well as to enforce the security devices. In conjunction with high-performance procedures, based on STP, path formation via SDN switches where filtering is provided, our approach effectively enhances the enterprise network availability and security.

Changing legacy device configurations the end device flow filtering can be moved from OF switches located in the centrality points (near distribution layer) down to the access if the channel actuation needed for trustworthy flows only, and

initial flow should be directed to authentication server or blocked from accessing the segment which does not fall under SDN control prior to authorization. Additionally, the network administrator can be informed about suspicious devices and flows.

IV. Conclusions

In the present work the topology discovery and management issues in SDN and hybrid networks are thoroughly discussed as well as the hybrid network control system based on thorough topology and state discovery and possibility to modify legacy configurations by converting OF flow rules into traditional device commands and instructions, if needed, incorporating these devices into SDN control area, is developed. The performance evaluation confirms the proof-of-concept applicability.

References

[1] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: an intellectual history of programmable networks", ACM SIGCOMM Comp. Commun. Rev., vol. 44, pp. 87–98, Apr. 2014.

[2] K. Benzekki, A. El Fergougui, and A. E. Elalaoui, "Software-defined networking (SDN): a survey", Security Commun. Networks, vol. 9, pp. 5803–5833, 2016.

[3] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," IEEE Commun. Surveys & Tut., vol. 17, pp. 27-51, 2015.

[4] J. H. Cox, et al. "Advancing software-defined networks: a survey", IEEE Access, vol. 5, pp. 25487-25526, 2017.

[5] R. Cziva, S. Jouët, D. Stapleton, F. P. Tso, and D. P. Pezaros, "SDN-based virtual machine management for cloud data centers", IEEE Trans. Netw. and Service mngmt., vol. 13, pp. 212-225, June 2016

[6] R. Ji, J. Li, X. Tuo, W. Wang, and X. Li, "A congestion control method of SDN data center based on reinforcement learning", Int J Commun Syst., vol. 31, pp. 1-11, Sept. 2018

[7] S. Jain, et al. "Experience with a globally deployed software defined WAN", in Proc. ACM SIGCOMM Conf., B.4, 2013.

[8] J. Griffioen, T. Wolf, and K. L. Calvert, "A Coin-Operated Software-Defined Exchange". 25th ICCCN. 2016.

[9] J. Mambretti, et al. "Designing and deploying a bioinformatics software-defined network exchange (SDX): architecture, services, capabilities, and foundation technologies", in Conf. On Innovations in Clouds, Internet Netw., 2017.

[10] J. Chung et al., "AtlanticWave-SDX: An international SDX to support science data applications", in Proc. Int. Conf. High Perform. Comput., Netw., Storage, Anal., pp. 1-7, 2015.

[11] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proc. IEEE, vol. 103, no. 1, pp. 14–76, 2015.

[12] E.Haleplidis, et al. "SDN layers and architectures terminology", RFC 7426, Jan 2015.

[13] T. Koponen, et al. "Onix: A distributed control platform for large-scale production networks." in Proc. 9th USENIX Conf. OSDI, pp. 351-364, 2010.

[14] OpenDaylight http://www.opendaylight.org/

[15] S. Y. Sinha, and K. Haribabu, "A survey: hybrid SDN", J. of Netw. and Comp. Appl., vol. 100, pp. 35–55, 2017.

[16] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: a survey of existing approaches", IEEE Commun. Surveys & Tuts, vol. 20 pp. 3259- 3306, 2018.

[17] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, S. Hu, "A survey of deployment solutions and optimization strategies for hybrid SDN networks, Commun. Surveys & Tuts, 2018 DOI 10.1109/COMST.2018.287106

[18] M. Huang and W. Liang, "Incremental SDN-enabled switch deployment for hybrid software-defined networks, in: 26th Int. Conf. ICCCN, 2017.

[19] M. Markovitch and S. Schmid, "SHEAR: A highly available and flexible network architecture marrying distributed and logically centralized control planes," in Proc. IEEE Int. Conf. ICNP, pp. 78–89, Nov. 2015.

[20] M. Casado, et al. "Rethinking enterprise network control," IEEE/ACM Trans. on Networking, vol. 17, pp. 1270–1283, Aug. 2009.

[21] F. Kuliesius, V.Dangovas, "SDN enhanced campus network authentication and access control system", in: 8 Int. Conf. ICUFN, pp. 894–899, 2016.

[22] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian, "Fabric: a retrospective on evolving SDN," in Proc. ACM SIGCOMM Workshop HTSDN, pp. 85–90, 2012.

[23] D. K. Hong, Y. Ma, S. Banerjee, and Z. M. Mao, "Incremental deployment of SDN in hybrid enterprise and ISP networks," in Proc. ACM SOSR, pp. 1–7, Mar. 2016.

[24] D. Levin, M. Canini, S. Schmid, F. Schaffert, and A. Feldmann, "Panopticon: Reaping the benefits of incremental SDN deployment in enterprise networks," in Proc. USENIX Annual Techn. Conf., pp. 333–345, Jun. 2014.

[25] H. Lu, N. Arora, H. Zhang, C. Lumezanu, J. Rhee, and G. Jiang, "Hybnet: Network manager for a hybrid network infrastructure," in Proc. Industr. Track of ACM/IFIP/USENIX Int. Middleware Conf., pp. 1–6, 2013.

[26] F. Farias, J. Salvatti, P. Victor, and A. Abelem, "Integrating legacy forwarding environment to OpenFlow/SDN control plane," in 15th APNOMS, pp. 1–3, Sept. 2013.

[27] C. E. Rothenberg et al., ''Hybrid networking towards a software defined era,'' in Network Innovation Through OpenFlow and SDN: Principles and Design, London, U.K.: Taylor & Francis/CRC Press, 2014.

[28] R. Hand and E. Keller, "Closedflow: Openflow-like control over proprietary devices," in Proc. Third Workshop HotSDN'14, pp. 7–12, August 2014.

[29] D. J. Casey and B. E. Mullins, "SDN shim: Controlling legacy devices," in Proc. IEEE LCN, pp. 169–172, 2015.

[30] C. Jin, C. Lumezanu, Q. Xu, Z.L. Zhang, and G. Jiang, "Telekinesis: Controlling legacy switch routing with OpenFlow in hybrid networks," in Proc. ACM SIGCOMM Symp. on SDN Research, pp. 1–7, 2015.

[31] C. Jin, C. Lumezanu, Q. Xu, H. Mekky, Z.-L. Zhang, and G. Jiang, "Magneto: Unified fine-grained path control in legacy and OpenFlow hybrid networks," in Proc. ACM Symp. on SDN Research, pp. 75–87, 2017.

[32] O. Tilmans, S. Vissicchio, L. Vanbever, and J. Rexford, "Fibbing in action: On-demand load-balancing for better video delivery," in Proc. ACM SIGCOMM Conf., pp. 619–620, 2016.

[33] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford, "Central control over distributed routing," ACM SIGCOMM Comp. Commun. Review, vol. 45, pp. 43–56, 2015.

[34] S. Khan, A. Gani, A. Wahid, A. Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software defined networks: threats, taxonomy, and state-of-the-art", IEEE Commun. Surveys & Tuts, vol. 19, pp. 303-324, 2017.

[35] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in software defined networks," in Proc. 8th ICSPCS, pp. 1–8, 2014.

[36] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in openflow-based software defined networks", Computer Commun vol. 77, pp. 52-61, 2016.

[37] L. Ochoa-Aday, C. Cervelló-Pastor, and A. Fernández-Fernández, "Current trends of topology discovery in OpenFlow-based software defined networks", External research report, 2015. http://hdl.handle.net/2117/77672

[38] A. Azzouni, N. Thi M.Trang, R. Boutaba, and G. Pujolle, "Limitations of OpenFlow Topology Discovery Protocol", 16th Ann. Mediterranean Ad Hoc Networking Workshop, 2017.

[39] IEEE Standard for Local and Metropolitan Area Networks - Station and Media Access Control Connectivity Discovery, IEEE Std 802.1AB, 2009.

[40] SNMP4SDN: Architecture and Design, https://wiki.opendaylight.org/view/SNMP4SDN:Architecture_and_Design

[41] S. Pandey, M.-J. Choi, Y. J. Won, J. W.-K. Hong, "SNMP-based enterprise IP network topology discovery", Int. J. of Netw. Mngmt IJNM, vol. 21, pp. 169-184, May 2011.

[42] Mininet: An Instant Virtual Network, http://mininet.org/

[43] C. Sieber, A. Blenk, A. Basta, D. Hock, and W. Kellerer, "Towards a programmable management plane for SDN and legacy networks," in Proc. IEEE NetSoft, pp. 319–327, 2016.