Novel Approach to detect Anomalies using Defensive Algorithm in SDN flows

Sugandhi Midha[a], Dr. Khushbu Tripathi[b] and Dr. M.K.Sharma[c]

A
Chandigarh University, Mohali
B AMITY University, Gurgaon
CAmrapali Group of Institutes, Uttrakhand

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: The exponential growth of network users has to lead to poor management of networks that use the traditional networking approach. Traditional networking approaches have become an overhead in terms of flexibility, innovations, complexity, and programmability among the network. SDN guarantees a holistic approach to network flexibility and programmability. Network visibility in SDN gives scope for rapid innovation. SDN being a new paradigm, less work has been done towards security. Security is one of the biggest concerns in SDN. Separation of control and data plane in SDN has to lead to the emergence of Denial of Service (DoS) attack. The centralized controller in SDN makes it the best target for attackers and acts as a single point of failure. Attacks on the SDN controller can bring the entire network down. This paper presents an approach to monitor traffic and we propose a novel method to mitigate these anomalies and attacks in the network. We believe that the DoS attack can be toned down using this new technique.

Keywords: Software Defined Network (SDN), Security, Denial of Service (DoS), UDP flood, SYN flood, HTTP flood, ICMP flood

1. Introduction

Software Defined Networks is one of the prominent topics in the market these days. SDN is not only an academic dream and but also has tapped a strong foundation in the industry. SDN offers numerous opportunities like flexibility, scalability, monitoring, and fine-grained control for network management. Even though SDN can have physically distributed architecture but it still represents a network which is logically centralized controlled. SDN has a centralized console that manages the network traffic [23] dynamically as per the demands and requirements of the user. Traditional networking is built on a weak foundation and control plane 3 layers of abstraction. Networks need to be simple and SDN has made this possible. SDN architecture offers abstraction by decomposing network functioning into 3 different layers: Application, Control, and Data plane. The Data plane acts merely as a forwarding plane. The forwarding interface shields higher layers from forwarding hardware. The Control plane is the console controller that sets the rules in the flow table stored at a forwarding device. The application plane is responsible for load management, security, etc. problems.

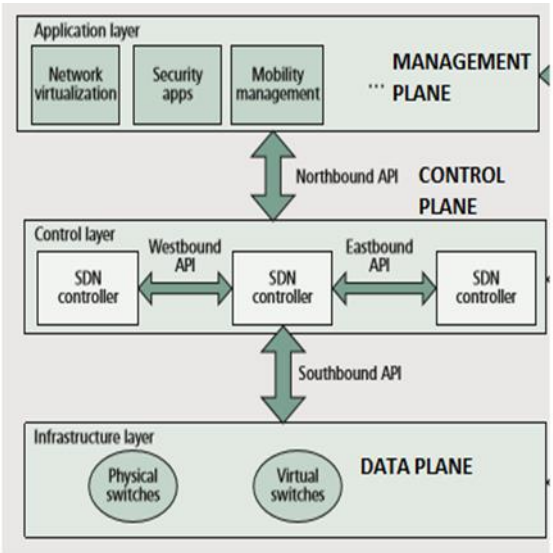

Fig 1. SDN Architecture (Sources: IEEE Communications Magazine, April 2005)

SDN focuses on the logical placement of centralized control in the form of an SDN controller. SDN is flexible and quickly adapts it to change in the environment. SDN supports innovations as programming with the SDN controller serves the same purpose.SDN architecture is more prone to DoS attacks due to SDN centralized controller.

DoS disrupt the availability of services even to legitimate users. DoS is an attempt to overload the server by consuming its entire resources CPU, memory, or available bandwidth of the network. We have tried and believe that the DoS attack on the SDN controller can be detected and mitigated using an effective algorithm.

Some well known DoS attacks are:

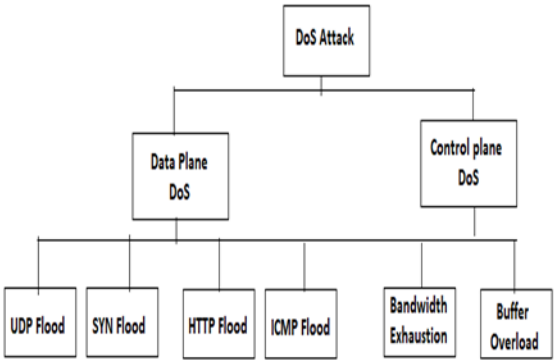

Fig 2. Types of DoS

Data Plane DoS: This category of attack is done on the forwarding machine. It overwhelms the victim machine with a large number of messages leading to machine failure by exhausting its resources.

UDP Flood: A large number of UDP framework packets are transferred to a random port on the victim machine to bring down the server. The attacker sends a stream of UDP packets to the target machine. Due to the low queue capacity of the victim machine; it overflows. The target machine will go down and won't be able to connect to the legitimate user. Mostly in these attacks, the IP address of UDP packets is spoofed to hide its identity.

SYN Flood: In contrast to connectionless UDP, this attack uses TCP connection-oriented message to the target machine. A stream of SYN packets is sent to the victim. In this case, no ACK is returned to the target machine which causes the breakdown of the machine.

HTTP Flood: A web server is flooded with a huge number of requests to a level where it is not able to respond to legitimate users.

ICMP Flood: This attack exhausts the resources of the target machine by flooding it with a huge number of ICMP request packets, it keeps the server occupied in replying to fake echo requests.

Control Plane DoS: In this attack, the attacker exhausts the bandwidth of the control plane. The Control plane is compromised by flooding of fake messages of an attacker. This flooding of fake messages paralyzes the controller and in turn, brings down the entire network.

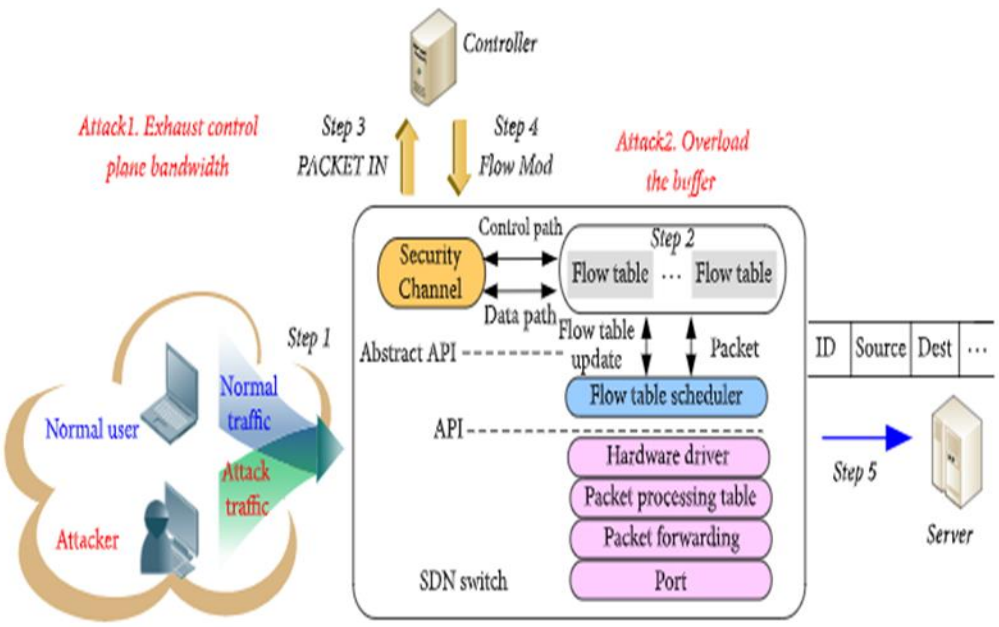

Fig 3. SDN architecture and Control Plane DoS

(Sources: Security and Communication Networks Volume 2018)

The paper is organized in the various sections as follows: Section II discusses an overview of related work done in SDN for anomaly detection and DoS attack. Section III of this paper describes the setup used and simulation DoS. Next Section IV illustrates the proposed model and open flow table management. Section V depicts a comparison

between the existing and proposed model in terms of performance metrics. Conclusion and future aspects are drawn in section VI.

2. Related Work

DDoS attacks and its detection methods have huge implications in terms of security of any network. Still, this area requires more exploration in context to SDN environments. Several recent papers [3][4][5] also listed out that the SDN controller is a central target of DDoS attacks and various measures are required to validate and verify the packets transmitted to SDN controller.

Also, very algorithms and techniques have been proposed for intrusion detection and DoS attack. For example, [1] focused on DoS attacks on switches, central console controller, northbound API, and southbound API, and [7] discussed decomposition of control and data plane & how SDN opens security challenges such as MITM, DoS, and saturation attacks. An evaluation of assertions on the flow table and its impact in terms of inconsistencies concerning a network security policy was done in [8]. How to develop a detailed security architecture that can offer security services in terms of enforcement of correct mandatory network policy for SDN [9] has been answered.

The art of SDN security and analysis of security issues concerning 3 layers was done by Zhaonang et al. [10]. They discussed several preventive and mitigation techniques. Many techniques [11] [12] provide flow detection & validation schemes but do not provide any confidence in them. An application that can block and detect DDoS to SDN [13] requires a two-way communication channel between DDoS blocking application and SDN controller server seeking protection. Various flow detection defects have been solved recently and its security architecture which can detect anomalies in the flow and an SDN environment was proposed namely [14][15].

In summary, Use rises, and capacity is denser in SDNs so basically SDN, in its full fulfillment, expels the go-between layers, making the product what is organized. SDN has affected the OpenStack open-source cloud programming just as virtualization stages like VMware's, and now SDN is coming snappy and quick to holders. Solving the issue of DDoS and taking full use of SDN has become a challenging task. The existing attack detection algorithms for SDN are characterized by low accurate results and are poor in timeliness.

Despite various work in SDN anomalies has been done still this area needs exploration as there is no strict solution to defend against DDoS attack on SDN. Our proposed algorithm is based on statistical analysis that has tried to select an apt feature for the attack.

3. Experimental Setup and Dos Simulation

For simulation purposes, mininet hypervisor installed on Ubuntu 14.04.4 OS, Secure Shell (SSH) is tunneled using putty for communication via windows platform to support secure network Graphical User Interface. Figure 4 illustrates the simple tree topology used for simulation purposes and Table 1 defines the used experimental setting for evaluating the performance of our proposed customized POX controller [25] [26].

Table 1: Experimental Setting

OS Type	Virtualization Software	X Server	Terminal	Emulator
Windows 7	Virtual Box	Xming	Putty	Mininet on Ubuntu 14.04.4

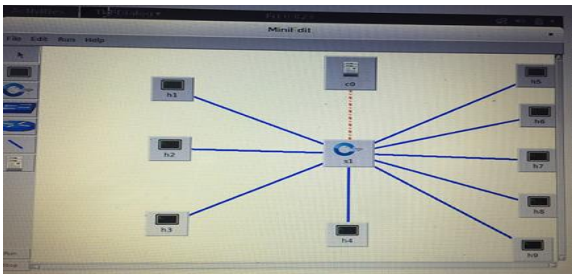

Fig 4. Simple Tree Network Topology

Figure 5 shows where an ip address is spoofed and it generates traffic on port 80 for ip “10.10.1.3” with a view to overload the target with multiple messages which will exhaust the network bandwidth and will also lead to buffer overflow of a target in turn causing DoS.

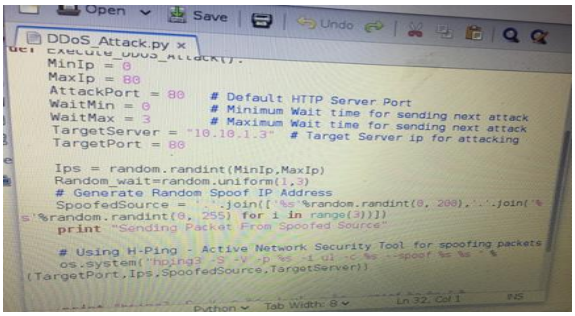

Fig 5. Target node "10.10.1.3" with traffic overflow from IP spoofed node

Figure 6 shows the impact of the DoS attack which leads to a 97% packet drop rate and low or no connectivity.

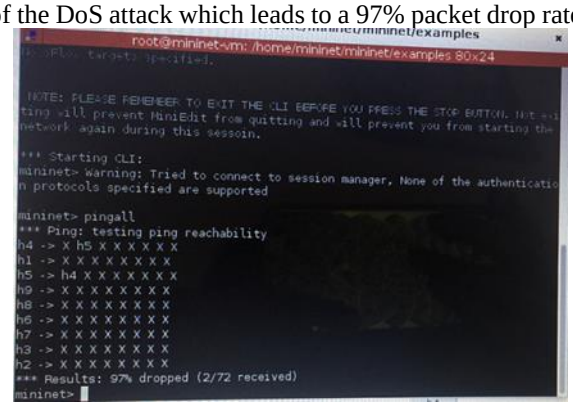

Fig 6. DoS Affect – Higher Packet Drop Rate

4. Defensive Algorithm and Ddos Switch

It's an automatic approach that ensures accuracy. Traffic statistics are collected, analyzing those statistics help in anomaly detection and to generate the plan of action to mitigate that attack. A controller handles the blocking and nonblocking flow table entries by the result analysis. Our algorithm works in two phases:

Statistics Collection Module: Traffic statistics on the network are gathered. Traffic behavior is observed and compared with what is expected traffic by an IP i.e. its normal profile. In case of any deviation, a mitigation step is launched.

Mitigation and Report Management Module: Next step is launched with 2 things in mind:

i) DoS on Controller

ii) DoS on Data Plane

i.e. a measure is found to identify either the attacker or the target who is the actual victim. Entries are made into blocking and non-blocking tables accordingly and appropriate action is taken to discard the incoming packets and shifting of the inappropriate load to the black hole.

Firstly, the controller initializes the Hash Flow Tables for blocking 'M' and nonblocking 'H' with entries. Custom DDoS_Switch [21] has been designed which is based on a customized POX controller [22] which upgrades the entries and smooth the functioning of the network.

Figure 7 shows the blocking and nonblocking flow table entries set up by the controller on DDoS_Switch [24] after carefully analyzing the traffic and observing the normal flow from an IP.

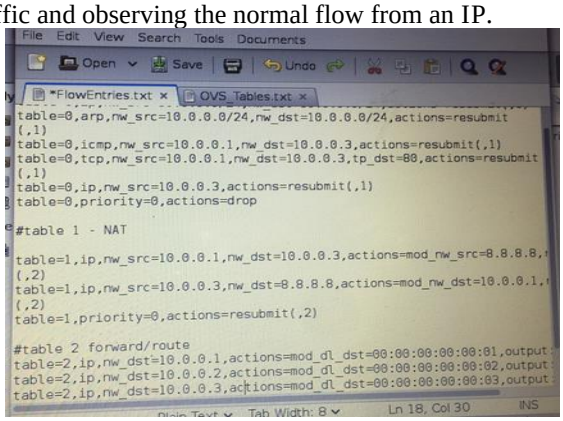

Fig. 7 Blocking 'M' and Nonblocking 'H' flow table entries

Figure 8 shows the aftereffect of our algorithm on traffic flow after the DDoS attack (refer to figure 6).

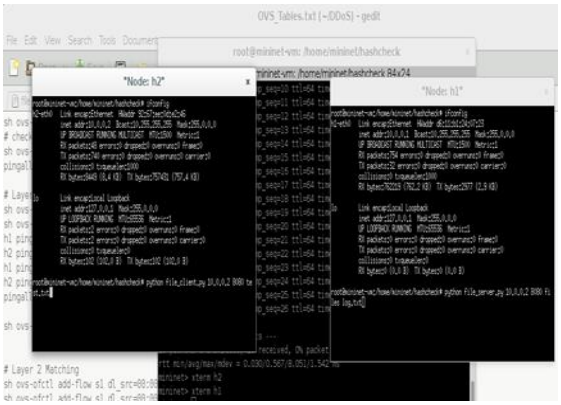

Fig. 8. Normal Traffic Flow after DDoS

5. Results and Simulation of Our Customised Pox Controller

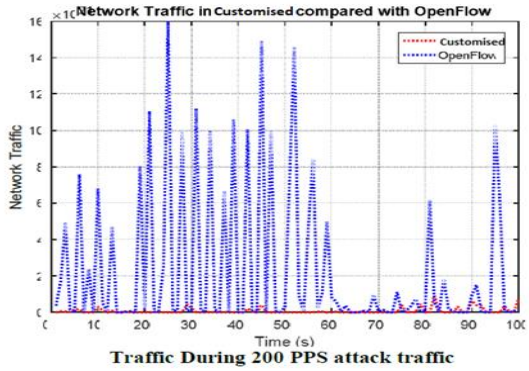

Fig. 9. Network Traffic in Customised controller and Open Flow Controller

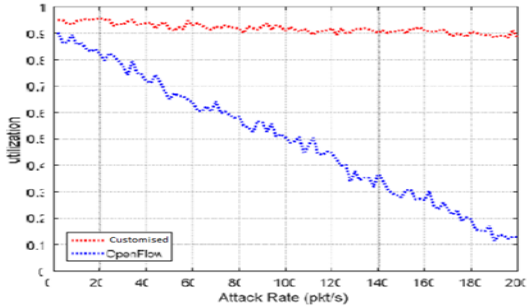

Fig. 10. Bandwidth utilization in Customised controller and Open Flow Controller

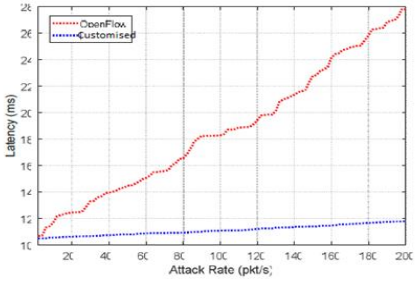

Fig. 11. Expected Latency in-network with the customized controller and Open Flow Controller

Noted Facts:

DOS: Packet Loss in the customized controller is low. We calculated the traffic at different amounts of time once there was a 16% drop rate. And later hardly 1% drop rate found.

Bandwidth Utilisation: In the case of the customized controller is high as the drop rate is less.

Latency Rate: low in the customized controller as found hardly 12 ms.

Network Utilisation is high in the customized controller.

6. Conclusion and Future Scope

In this paper, we have proposed a novel approach for detecting DDoS attacks in SDN network. This paper also presents a new and effective technique for detecting DDoS attack in SDN by making use of the combined traffic

monitoring and analysing methods by proposing a new customised PoX controller. The experiments are carried out. Through simulations we have proved the superiority of the proposed technique. Our proposed method compared with the open flow scheme, significantly reduces the error by reducing impact of attack. For our future works, we want to focus on how the proposed solution can be directly incorporated via POX Controller in MININET.

References

1. AdmelaJukan, Marcel Caria, Siquan Zhao, "Security in SDN", in proceedings of IEEE conference in IEEE, 2014.
2. Al-Shaer E, Al-Haj S (2010) FlowChecker: configuration analysis and verification of federated OpenFlow infrastructures. In: Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration, pp 37–44.
3. Diego Kerutz, Paulo Essteves, Siamak Azodolmolky, "Software Defined Networking: A c omprehensive Survey", Proceedings of IEEE, Vol. 103, No. 1, January 2015.
4. Fazal Hadia , Muhammad Imrana , Muhammad Hanif Durada , Muhammad Warisb, "A Simple Security Policy Enforcement System for an Institution Using SDN Controller", in proceedings of IEEE 15th international conference at IBCAST, January 2018.
    I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Analysis of Deployment Challenges of Host Identity Potocol", IEEE 2017.
5. Kaur, S., Singh, J., & Ghumman, N. S. (2014, August). Network programmability using POX controller. In ICCCS International Conference on Communication, Computing & Systems, IEEE (Vol. 138).
6. Lim S, Ha J I, Kim H, Kim Y, Yang S (2014) A SDN-oriented DDoS blocking scheme for botnet-based attacks. In: IEEE Sixth International Conference on Ubiquitous and Future Networks (ICUFN), pp 63–68
7. Midha S., Kaur G. (2020), " SVM Implementation for DDoS Attacks in Software Defined Networks", 'International Journal of Innovative Technology and Exploring Engineering (IJITEE)', ISSN: 2278–3075 (Online), Volume-10 Issue-1, November 2020, Page No. 205-212.
8. Midha S., Tripathi K. (2020) Data hiding based PKI Authentication Protocol in SDN. Testmagazine journal; Elsevier;Vol. 82: Jan/Feb 2020
9. Midha S., Tripathi K. (2020) Remotely Triggered Blackhole Routing in SDN for Handling DoS. In: Dutta M., Krishna C., Kumar R., Kalra M. (eds) Proceedings of International Conference on IoT Inclusive Life (ICIIL 2019), NITTTR Chandigarh, India. Lecture Notes in Networks and Systems, vol 116. Springer, Singapore. https://doi.org/10.1007/978-981-15-3020-3_1
10. Midha S., Tripathi K. (2021) Extended Security in Heterogeneous Distributed SDN Architecture. In: Hura G., Singh A., Siong Hoe L. (eds) Advances in Communication and Computational Technology. Lecture Notes in Electrical Engineering, vol 668. Springer, Singapore. https://doi.org/10.1007/978-981-15-5341-7_75
11. Noxrep. (2015, Jan.) POX: OpenFlow Controller. [Online]https://www.opennetworking.org/wp-content/uploads/2014/10/Principles_and_Practices_for_Securing_Software-defined_Networks_applied_to_OFv1.3.4_V1.0.pdf
12. Qamar Ilyas, Rahamatullah Khondoker, "Security Analysis of FloodLight, ZeroSDN, Beacon and POX SDN Controllers", in SDN and NFV security, (pp. 85-98) springer, cham.
13. Qioa Yan and F. Richard Yu, "Distributed Denial of Services Attacks in Software Defined Networking with Cloud Computing", in IEEE Communications Magazine, April 2005.
14. S. Midha and K. Triptahi, "Extended TLS security and Defensive Algorithm in OpenFlow SDN," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 141-146, doi: 10.1109/CONFLUENCE.2019.8776607
15. S. Midha, G. Kaur and K. Tripathi, "Cloud deep down — SWOT analysis," 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), Noida, 2017, pp. 1-5, doi: 10.1109/TEL-NET.2017.8343560.
16. S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model Checking Invariant Security Properties in OpenFlow." Available: http://faculty.cse.tamu.edu/guofei/paper/Flover-ICC13.pdf
17. Sandeep Pisharody, Janakarajan Natarajan, Ankur Chowdhary, Abdullah Alshalan , "A Security Policy Analysis Framework for Distributed SDN-Based Cloud Environments", in IEEE Transactions on Dependable and Secure Computing (Volume: PP, Issue: 99), July 2017.
18. Sherwood R, Gibb G, Yap K K, Appenzeller G, Casado M, McKeown N, Parulkar G (2009) Flowvisor: a network virtualization layer. OpenFlow Switch Consortium, Tech. Rep.
19. Tao Wang , Hongchang Chen, Guozhen Cheng, and Yulin Lu, "SDNManager: A Safeguard Architecture for SDN DoS Attacks Based on Bandwidth Prediction", in Security and Communication Networks Volume 2018, Article ID 7545079, 16 pages.
20. Wang, B., Zheng, Y., Lou, W., & Hou, Y. T. (2015). DDoS attack protection in the era of cloud computing and software-defined networking. Computer Networks, 81, 308-319.

21. Yan, Q., & Yu, F. R. (2015). Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Communications Magazine, 53(4), 52-59.
22. Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Communications Surveys & Tutorials, 18(1), 602-622.
23. Zhaogang Shu, Jiafu Wan, Di Li, Jiaxiang Lin , Athanasios V. Vasilakos, Muhammad Imran, "Security in Software-Defined Networking: Threats and Countermeasures", in Springer Science, Business Media New York 2016.
24. Zhiyuan Hu, Mingwen Wang, Xueqiang YAN, Yueming YIN, "A comprehensive security architecture for SDN", in the IEEE proceedings of 18th international conference on Intelligence in Next Generation Networks, IEEE, 2015.
25. Zhiyuan Hu, Mingwen Wang, Xueqiang YAN, Yueming YIN, "A comprehensive security architecture for SDN", in the IEEE proceedings of 18th international conference on Intelligence in Next Generation Networks, IEEE, 2015.