# SDWN for End-to-End QoS Path Selection in a Wireless Network Ecosystem

Eugeniu Semenciuc, Andra Pastrav, Tudor Palade, and Emanuel Puschita(✉)

Communication Department, Technical University of Cluj-Napoca, G. Baritiu 26-28, 400027 Cluj-Napoca, Romania
{eugeniu.semenciuc, andra.pastrav, tudor.palade, emanuel.puschita}@com.utcluj.ro

**Abstract.** In recent years, a significant increase of mobile data traffic has been observed, resulting in more complex and diverse network services. Thus, it becomes more difficult to fulfill the requirements of QoS-sensitive applications. The emergence of new technologies such as Cloud Computing, Software Defined Networking and Network Function Virtualization provides an excellent ecosystem for QoS routing solutions to support real-time applications requirements in wireless networks. The scope of this paper is to propose a Software Defined Wireless Network controller implementation to determine the appropriate end-to-end QoS path in a wireless network ecosystem. The proposed solution is based on Bayesian reasoning and Fuzzy logic algorithms.

**Keywords:** QoS routing · End-to-end QoS path · Wireless network ecosystem Bayesian reasoning · Fuzzy logic

## 1 Introduction

The development of Internet and telecommunication technologies in the last couple of years established a well-connected and communicative information society. New applications and services like social networking, multimedia, Voice over IP (VoIP), virtual reality, mobile applications and Internet of Things (IoT), generate a big amount of data and require a scalable, high performance, energy efficient and reliable networking infrastructure to provide strict Quality of Service (QoS). Meeting these requirements is getting more challenging each day with existing network architecture and devices due to their limited capabilities. Thereby the Software Defined Network (SDN) paradigm has emerged [1].

SDN is a network architecture that consists of three layers: applications, control plane and data plane plus a set of interfaces [2]. SDN architecture separates the control plane (that makes decisions about where traffic is sent), and the data plane (which contains the devices responsible for traffic forwarding) [2]. The applications use specific Northbound interfaces (NBIs) to enforce their policies in the data plane, while the SDN controller uses Southbound interfaces (SBIs) to communicate with the network equipment [2]. Due to programmability of SDN controllers and network devices, the introduction of new protocols and improvements requires only modifications of


O. Fratu et al. (Eds.): FABULOUS 2017, LNICST 241, pp. 134–140, 2018.
https://doi.org/10.1007/978-3-319-92213-3_20

software programs, creating a more flexible, extensible and open framework for network design and upgrade. Such attractive features of SDN provoke the implementation of the SDN architecture into wireless networks, resulting in the emergence of Software Defined Wireless Networks (SDWN) [3].

Wireless network ecosystems consist of different technologies, services and network nodes that create a complex hybrid access network with specific requirements like mobility management, dynamic channel configuration and rapid client re-association. With the increase of the diversity of network services, fulfilling the QoS requirements becomes more and more challenging. Since the SDN approach was proved to be successful on legacy wired networks, it becomes intensely adopted by other networks such as wireless, mobile and sensor networks. Therefore, the scope of this paper is to propose a solution that determines a feasible end-to-end QoS path in a wireless network ecosystem based on SDWN concept. The originality of the proposed solution consists in (1) the use of selective probing to estimate the network state, (2) the estimation of new application performances using Bayesian reasoning, and (3) the classification of available routes based on multiple metrics (delay, jitter, packet loss) integrated in a Fuzzy logic algorithm.

The remainder of the paper is organized as follows. In Sect. 2 the most notable open-source SDN controller are compared. Section 3 introduces some attempts to enhance QoS support using SDWN. Section 4 describes the proposed Fuzzy logic QoS solution in a SDWN architecture. Finally, Sect. 5 concludes the paper.

## 2 SDN Controllers

The SDN architecture is managed by a central controller that enables on-demand resource allocation, self-service provisioning and virtual networks. The controller is a fundamental part of the SDN architecture. A set of programming languages, architectures, APIs, and protocols have been adopted by controllers [4, 5]. Various SDN controller features are compared in Table 1.

**Table 1.** Feature-based comparison of the most popular open source SDN controllers.

| | Language | Architecture | SBI APIs | NBI APIs | Storage |
|---|---|---|---|---|---|
| ONOS | Java | Flat distributed | OpenFlow1.5 OVSDB, BGP, SNMP, PCEP NETCONF, P4, IS-IS, OSPF | REST API | RAFT LOG |
| Open Daylight | Java | Flat distributed | OpenFlow1.4, NETCONF, OVSDB, CEP, BGP, LTSP, SNMP | REST API RESTCONF | In memory datastore |
| NOX | C++ | Centralized | OpenFlow 1.0 | REST | WheelFS |
| IRIS | Java | Centralized | OpenFlow 1.0, 1.3, OVSDB | REST API | Mongo |

OpenFlow [6] is the first communication interface defined between control and data layers of an SDN architecture, and permits switches to update their flow table entries to take corresponding actions per the incoming flows and defined rules.

Considering specific application constraints, we have selected OpenDaylight and ONOS as the best choices, thanks to their distributed architecture, wider applications, topology discovery and stability. In [7] ONOS and OpenDaylight performances are compared using Cbench and Mininet. The results show that ONOS has good performance on clusters, link-up, switch-up and throughput.

## 3 SDWN and QoS Support

Open Roads [8] was one of the first efforts in the development of a wireless SDN architecture, based on OpenFlow, NOX controller and FlowVisor. This solution was deployed on Cambridge University Campus Network. In this framework, OpenFlow separates control of the forwarding plane, SNMP protocol allows for device configuration and FlowVisor creates the network slices and ensure isolation between them. Therefore, multiple scenarios can be run on the same physical infrastructure and researchers can easily adopt SDN paradigms in cellular networks.

OpenRadio [9] proposes a novel architecture that provides modular and declarative programming capability for the entire wireless stack. Because multiple blocks at the PHY and MAC layers are reused across multiple wireless protocols (3G, 4G, WLAN) OpenRadio decouples wireless protocol definition from the hardware, offering the possibility of defining and customizing protocols programmatically. The processing plane includes actions, signal processing operations. The decision plane, includes rules, responsible for choosing which actions must be performed.

CloudMAC [10] presents a new management distributed architecture for WLAN systems by moving the MAC layer processing to virtual machines connected by an OpenFlow network. CloudMAC architecture consists of Virtual APs (VAPs), Wireless Termination Points (WTPs), an OpenFlow switch and an OpenFlow controller. This architecture enables several new applications, such as dynamic spectrum use, on-demand AP and downlink scheduling. The performance evaluation has shown that CloudMAC achieves good performance and seamlessly interworks with standard IEEE 802.11 stations.

In [11] Xu et al. proposes a SDWN multipath-transmission (MP-SDWN) solution. Using an extension of Floodlight OpenFlow, the MP-SDWN provides network slicing to support multiple virtual WLANs running on the same physical infrastructure. Also, it introduces the multi-connection virtual access point (MVAPs) concept, which forms the same virtual access point over different physical APs. Based on MVAP, all the traffic from a user can be transmitted over multiple APs simultaneously, thus increasing the throughput considerably. The experiment results show that compared to WiFi, MP-SDWN achieves notable throughput improvement.

In [12] the load imbalance problem is examined by proposing a centralized Software Defined LTE RAN (SD-LTE-RAN) framework and a new QoS Aware Balance (QALB) algorithm. The solution provides QoS satisfaction using Priority Set Scheduler (PSS). PSS tries to allocate more resources for higher priority UEs according to their configured GBRs (Guaranteed Bit Rate). The study results are based on simulations realized using NS-3 simulator and they show that QALB obtains better QoS data rates and decreases the total network overload by 15% compared to existing algorithms. As shown in [11, 12] SDN permits QoS decisions based on network state, but to the best of our knowledge, none of the existing mechanisms estimates the future performances using Bayesian Networks and Fuzzy Logic.

## 4 Fuzzy QoS Support in SDWN

Although the SDWN concept is quite new and not standardized, it offers support for a stable and mature development ecosystem. Nearly all proposed QoS solutions use SDWN and NFV (network function virtualization) to manage in a better way available resources, offer load balancing and provide different queuing techniques. However, there are a few solutions that approach QoS routing in wireless access networks, and even fewer that consider composite metrics.

This section proposes a solution to determine the appropriate end-to-end QoS path in a wireless network ecosystem using the SDWN paradigm. The proposed implementation represents a transition of the QoS support mechanism presented in [13] to SDWN architecture to obtain a more flexible, robust and scalable solution.

In [13] the proposed algorithm was integrated with QualNet simulator and the simulations results showed estimation accuracy and good learning capabilities.

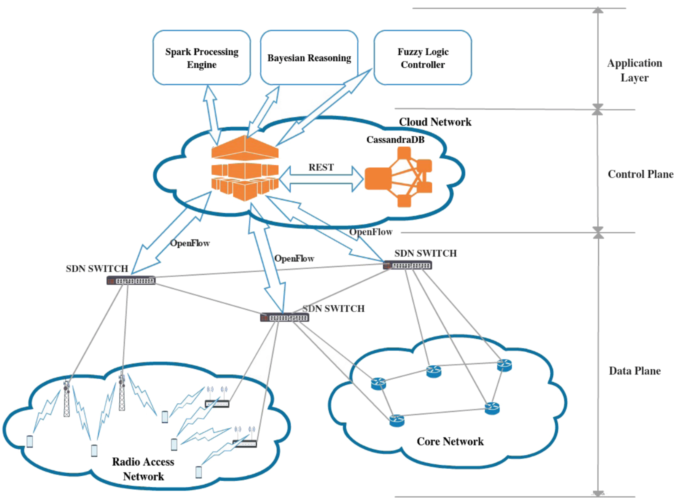


**Fig. 1.** SDWN based QoS support model.

As shown in Fig. 1, the selected architecture is based on ONOS controller that manages the core wired network and the radio access network and uses OpenFlow to communicate with data plane. The pseudocode of the algorithm is presented above:

```
Step1. Gathering Intelligence
foreach (node){
 if (packet) {
     computeDelay();computeJitter();getBandwidth();
     sendPacketToONOS();savePacketInDB();}}
Step2. Estimate QoS performances of a new Application:
sendsAppRequirementsToONOS();estimateQoS();
makesFuzzyDecisions();sendsPathToSource();
estimateQoS () {
 getHistoryDBData(source,destination,appData);
 List<QoSData> learningData;groupBy(AppID);
 foreach (route){
  assign(delay, jitter, bandwidth)
          to QoS{POOR,MODERATE,GOOD,EXCELLENT}}
 P(metric= QoSValue|DBData) = (1 + #(QoSValue))/(3+N);}
```

A NoSQL database cluster is considered for network state parameters storage. Thereby ONOS controller will have the global view of the network based on applications performance logging. Additionally, to maintain a more accurate and current network state, a probing mechanism is proposed considering applications characteristics [13]. The probes are sent in a hop-by-hop manner, from the source to the destination node. Each intermediate node analyzes the performance in terms of the QoS parameters and updates the database with new results. Central controller will use this information to perform Bayesian reasoning to estimate the performance of each path for applications. To classify these routes, a Fuzzy algorithm, named Fuzzy Logic Controller (FLC) is being used. FLC is described by the following steps: 1 - using defined membership functions, applied for metrics, such as delay, jitter, bandwidth, the affiliation degree to each fuzzy set is determined; 2 - fuzzy linguistic rules for inputs obtained in the previous step are evaluated and 3 - a percentage, characterizing the performance of the route, is issued by applying defuzzification [14].

The data processing and estimation tasks will be implemented as an ONOS application using Apache Spark, cluster-computing framework. Apache Spark contains the MLlib (Machine Learning Library), which due to its distributed memory-based architecture performs nine times faster than the disk-based implementations.

With increasingly sophisticated ways of probing and recording network activities, the network data analysis becomes a Big Data problem. Different algorithms can be applied on these data, to predict and estimate future network behavior, and to create an central controller which learns to manage the network on its own. To experimentally evaluate the proposed SWDN concept, we will use Mininet-WiFi [15].

## 5 Conclusions

This paper proposes a QoS support based on SDWN concept to estimates the future network performance based on previous network states and selects an end-to-end path that fulfills the application requirements. The particularities of this solution are: probing to save network states in a distributed NoSQL database, centralized data processing and decision taken by ONOS controller using Bayesian reasoning and Fuzzy logic algorithms. Conducted studies confirm that SDWN and Machine Learning represent a promising QoS solutions for wireless network ecosystems.

**Acknowledgement.** This paper was partially supported by a grant of the Ministry of Innovation and Research, UEFISCDI, project number 6 Sol/2017 within PNCDI III and by a grant of the Ministry of National Education and Scientific Research, RDI Programme for Space Technology and Advanced Research - STAR, project number 116/2016.

## References

1. Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**, 14–76 (2015)
2. Masoudi, R., Ghaffari, A.: Software defined networks: a survey. J. Netw. Comput. Appl. **67**, 1–25 (2016)
3. Akyildiz, I., Lin, S., Wang, P.: Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: an overview and qualitative evaluation. Comput. Netw. **93**, 66–79 (2015)
4. The OpenDaylight Platform—OpenDaylight. https://www.opendaylight.org/
5. Open Network Operating System. http://onosproject.org/
6. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow. ACM SIGCOMM Comput. Commun. Rev. **38**, 69 (2008)
7. Yamei, F., Qing, L., Qi, H.: Research and comparative analysis of performance test on SDN controller. In: 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI) (2016)
8. Yap, K., Kobayashi, M., Sherwood, R., Huang, T., Chan, M., Handigol, N., McKeown, N.: OpenRoads. ACM SIGCOMM Comput. Commun. Rev. **40**, 125 (2010)
9. Bansal, M., Mehlman, J., Katti, S., Levis, P.: OpenRadio. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks - HotSDN 2012 (2012)
10. Dely, P., Vestin, J., Kassler, A., Bayer, N., Einsiedler, H., Peylo, C.: CloudMAC - an OpenFlow based architecture for 802.11 MAC layer processing in the cloud. In: 2012 IEEE Globecom Workshops (2012)
11. Xu, C., Jin, W., Zhao, G., Tianfield, H., Yu, S., Qu, Y.: A novel multipath-transmission supported software defined wireless network architecture. IEEE Access. **5**, 2111–2125 (2017)
12. Farshin, A., Sharifian, S.: A chaotic grey wolf controller allocator for Software Defined Mobile Network (SDMN) for 5th generation of cloud-based cellular systems (5G). Comput. Commun. **108**, 94–109 (2017)

13. Semenciuc, E., Pastrav, A., Palade, T., Puschita, E.: Performance evaluation of a Cloud-based QoS support mechanism. In: 2016 International Conference on Communications (COMM) (2016)
14. Zadeh, L.: Theory of fuzzy sets. Electronics Research Laboratory, College of Engineering, University of California, Berkeley, Berkeley, CA (1977)
15. Fontes, R., Afzal, S., Brito, S., Santos, M., Rothenberg, C.: Mininet-WiFi: emulating software-defined wireless networks. In: 2015 11th International Conference on Network and Service Management (CNSM) (2015)