Performance evaluation of SDN WLAN architecture

Kristián Košťál, Rastislav Bencel, Michal Ries, Ivan Kotuliak
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Bratislava, Slovakia
{kristian.kostal, rastislav.bencel, ivan.kotuliak, michal.ries}@stuba.sk

***Abstract*—This article focuses on mobility of stations under 802.11 standards, with use of Software-Defined Networking (SDN). Lately, there are several SDN-based solutions for seamless handover. These solutions typically use separate control channels to utilize the wireless part of network. In our work, we demonstrate high performance of SDN WLAN architecture under real word test scenarios. Our common control channels solution extends the OpenFlow protocol. Our proposed method is notably improving performance of network architecture in next topics: network scalability, security and parallelly it allows delay critical scenarios to be delivered smoothly including seamless handover.**

Keywords—SDN; WLAN handover; IEEE 802.11

I. Introduction

Fast and seamless handover is crucial for maintaining sufficient service quality especially in case of delay critical scenarios, e.g. VoIP, video streaming and online gaming and case of network congestion. The major drawback of data delivery under 802.11 standards [1] is poor handover performance [2], [3]. The handover performance is mainly limited due to the fact that a client can be associated to only one access point (AP) at any given time and the client performs a new association to the AP each time the signal strength falls below a given threshold. In case of provisioning delay critical services or network congestion of AP in mobile environment (user moves in a larger area in between multiple 802.11's APs) such handover prerequisites negatively affect quality of user experience (QoE) [2], [4] or maximum throughput decrease. Moreover, 802.11 standards assign handover decision to the client. Only standards 802.11r and 802.11k mitigate handover issue but decision is still assigned to the client [1], [3].

Software-Defined Networking (SDN) introduces mechanisms to solve client's mobility, load balancing and 802.11 network management [5]. In order to improve network performance in mobile environment it is necessary to integrate handover management over the wireless network to SDN.

The SDN network architecture allows to decouple control and data planes. Routers and switches become simple forwarding devices because control logic is moved to central logic device called SDN Controller. The SDN Controller has two interfaces; Southbound and Northbound. The Southbound interface is standardized [6] and it controls data plane in the network. The Northbound interface is determined for services in the form of applications on the top of the SDN Controller. These applications are independent from infrastructure and network devices. In general, the SDN architecture simplifies network design and allows vendor-independent network devices. Therefore the SDNs provide better management for the wireless networks as well [5], [6].

There are several solutions how to solve issues of user mobility. Authors in [7] proposed Odin framework for introducing programmability in WLAN. This framework allows implementation of own handover and load balancing logic. The basic principle is bridging of the virtual and physical entity of APs. The virtual entity of AP is represented as the light virtual access point (LVAP), which contains BSSID, SSID, client's MAC address and client's IP address. The migration of the LVAP is managed by SDN Controller and the process is undetectable by a client. Physical APs are extended by Odin agent for WiFi support and OpenFlow protocol for a wired network part. The Odin Agent communicates with SDN Controller using custom Odin protocol which is transferred via TCP or UDP protocols depending on the type of a message. Besides that, the physical AP uses OpenFlow protocol for a control data plane in wired part of the network. The architecture with Odin Framework has three separated control channels to ensure network functionality. The SDN architecture provides standardized Southbound interface and this interface is bypassed by the Odin custom protocol. Authors of the SWAN solution [8] provide similar solution as Odin framework. This solution primarily focuses on migrating VAP context called SAP (Software Access Point). A communication protocol between SAP agent and SWAN Controller is realized by custom SWAN protocol. Both solutions [7], [8] are suffering similar drawbacks; usage of separated control channels for management of 802.11 part of the network.

Authors of "One Big AP" solution [9] chose different approach how to solve slow handover in WLAN. Every physical AP has assigned the same BSSID and the same channel. When the user station is associated to the network all user traffic is managed by SDN Controller. The handover is done by changing a data flow from one AP to another AP. The data flow is controlled by extended OpenFlow protocol. Using the same channel in network is a disadvantage because of overlapping signals from physical APs due to network interferences. These interferences have negative impact on the network performance.

In this article, our solution bridges virtual and physical APs in order to achieve efficient seamless handover. The proposed architecture is derived from the Odin framework and extends solution in [7] and our previous article [10]. Our extension improves original Odin framework in following points: more efficient use of 802.11 network resources and improved security

ISBN 978-3-903176-03-4© 2018 IFIP

of network management. Furthermore, our architecture is more transparent because we use only standard SDN architecture interfaces.

This paper is organized as follows. The design of our solution is presented in Section II. Implementation of the solution is described in Section III. The testbed description is situated in Section IV. and performance measurements are described in Section V. The conclusions and future work are discussed in Section VI.

II. Design of our solution

Architecture and design of our solution is described in this section. The key novelty is merging control channels under SDN management. The original Odin solution or its derivations [7], [11] use separated control channels for part of network running under 802.11 standards and the rest. These separated control channels are depicted in Fig. 1a. and proposed architecture in Fig. 1b. Common control channels have following advantages:

- Clear design and transparency – using standardized SDN interfaces with purpose they were defined. These interfaces are Southbound and Northbound. We choose OpenFlow protocol extended by new control messages for the Southbound interface [12]. The northbound interface does not have general standard and it is implementation-based. There are no additional interfaces or new protocols implemented into our architecture.
- Higher security due to encryption between the SDN Controller and the SDN forwarding device using OpenFlow protocol. The OpenFlow protocol assumes usage of underlying security protocols for increasing security of SDN network management.

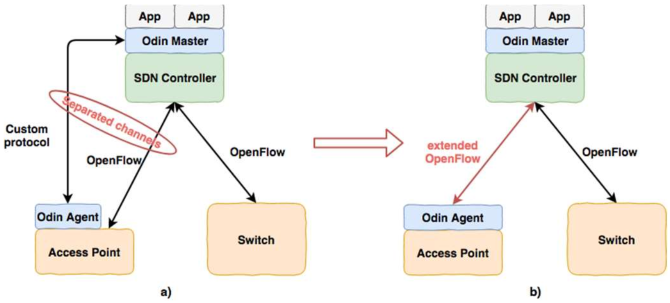
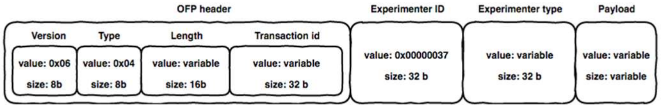


Fig. 1 a) Original Odin architecture b) Odin improved architecture

The UDP and TCP control channels are integrated to one OpenFlow control channel. The OpenFlow [12] is communication protocol between SDN Controller and network devices. This protocol is used in Odin solution for wired part of the network. Moreover, it is necessary to use OpenFlow protocol extensions to define new features for transporting control messages between APs and Controller. In parallel it is necessary to keep full compatibility with OpenFlow protocol standard. Our implementation is based on extension of OFP (OpenFlow protocol) Experimenter message features. The original structure of the OFP Experimenter message as designed for our architecture is depicted in the Fig. 2. This message consists of two parts. The first part is the OFP header and the second is the experimenter part. The OFP header determines the version and the type of OFP messages. Experimenter part contains Experimenter ID for determination of the message and Experimenter Type which determines a message subtype of the Experimenter ID. This is followed by payload. The structure is compatible with OpenFlow protocol version 1.5.x. The proposed subtypes of experimenter messages are listed in TABLE I and the payload structure is explained in [13].



OFP header				Experimenter ID	Experimenter type	Payload
Version	Type	Length	Transaction id			
value: 0x06 size: 8b	value: 0x04 size: 8b	value: variable size: 16b	value: variable size: 32 b	value: 0x00000037 size: 32 b	value: variable size: 32 b	value: variable size: variable

Fig. 2 Structure of the OpenFlow Experimenter message for proposed solution

TABLE I. Messages of the control protocol between APs and Controller

Message name	*Experimenter Type*
RECVD_PROBE	0x00000001
ADD_LVAP	0x00000002
REMOVE_LVAP	0x00000003
ADD_SUB	0x00000004
PUBLISH	0x00000005
QUERY_STATS	0x00000006
PING	0x00000007

A. Access Point

The access point in the improved architecture was changed to support extended OpenFlow protocol by new messages. The new messages are contained in a part called Wireless enhancer. The Wireless enhancer functions are used to join OpenFlow switch part with Odin agent and to match custom Odin protocol messages with the proposed OpenFlow experimenter messages. The experimenter messages are transferred to SDN Controller via Southbound interface by OpenFlow protocol. Apart from adding the Wireless enhancer to the AP, we improved the Odin Agent as well. It is modified by removing direct communication with the Odin Master (protocol UDP and TCP) and adding internal communication with the Wireless enhancer. The structure of the access point is depicted in Fig. 3.

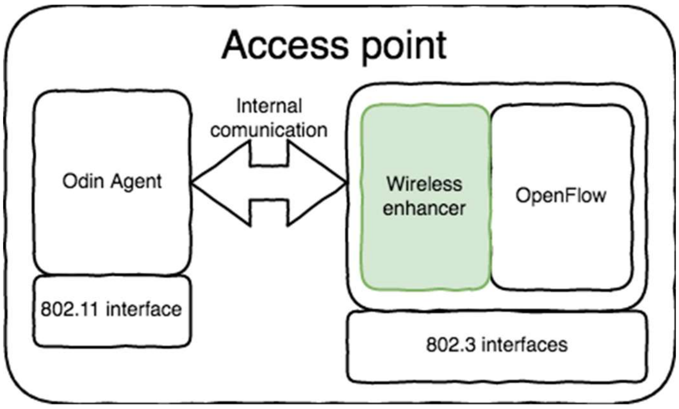


Fig. 3 Structure of the access point

B. *SDN Controller*

The SDN Controller manages the infrastructure and wireless parts of the network. The wireless part is managed by the Odin Master and in the proposed architecture it is changed by removing the direct control channel between Odin Master and the AP. The Odin Master gets all the information from the SDN Controller about the wireless part of the network. The SDN Controller receives this information via Southbound interface and passes it to the Odin Master. This information is extracted from the new proposed OpenFlow messages. The infrastructure part of the network is managed by OpenFlow protocol too. This part doesn't have to support new extended OpenFlow messages.

C. *Communication protocol between AP and Controller*

The communication protocol between AP and SDN Controller is designed as extension of OpenFlow protocol following specification for extension design [11]. We divide functionality of the communication protocol to two main parts based on 802.11. These two parts are:

- Connection to the network – involves reaching 802.11 associated state and assigned IP address to the user station. The station is provided with access to network resources. The connection to the network is depicted in Fig. 4 points 1 and 2.
- Handover process – is migration of LVAP from one physical AP to another physical AP. To this part, we include also statistic reports. The handover process is depicted in Fig. 4 point 3.

The AP sends RECVD_PROBE message to SDN Controller after receiving the probe request from the user station. This probe request must have the same SSID as the AP. The SDN Controller receives the RECVD_PROBE message and decides about the next step. If the association process is continuing the SDN Controller sends ADD_LVAP message to the AP. On the other hand, the SDN Controller ignores information about the received probe request on the AP. The ADD_LVAP message doesn't contain user station IP address in the first step (see Fig. 4 point 1). The IP address for user station is assigned by DHCP server immediately after user station reaches associated state. All DHCP messages are transferred via the SDN Controller to the DHCP server. The AP and the SDN Controller add the IP address to the user LVAP.

The handover decision is done on a basis of reported statistics from the APs via PUBLISH message. The PUBLISH message contains a signal strength of user station. In our case the signal strength is evaluation parameter for handover decision. The client is moved to new AP after old AP signal strength is below threshold and new AP signal strength is above this threshold. When the SDN Controller decides to make handover, the LVAP is migrated from the old AP to the new AP. This migration of LVAP is done by removing the LVAP from the old AP and adding the LVAP to the new AP. The removing LVAP process is done by REMOVE_LVAP message and the adding LVAP is done by ADD_LVAP message.

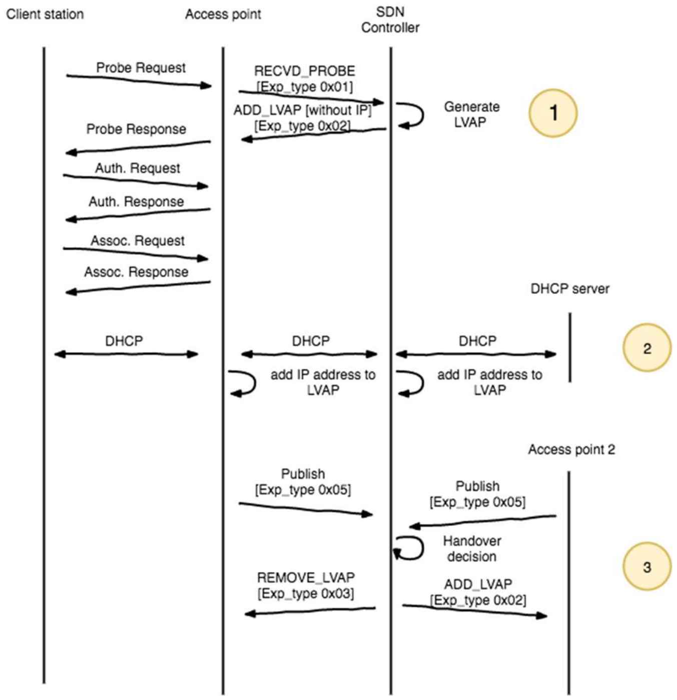


Fig. 4 Communication protocol with handover call flow

III. Implementation

In this section we describe implementation of our solution to SDN networks. As mentioned above we used original implementation of Odin, which was extended according to our goals described in Section II.

As can be seen in Fig. 5, there are currently two network monitoring interfaces, main interface mon0 and auxiliary interface mon1. The whole operation of these interfaces is monitored by Odin agent. For that reason, the agent can perform special actions according to contents of 802.11 frames.

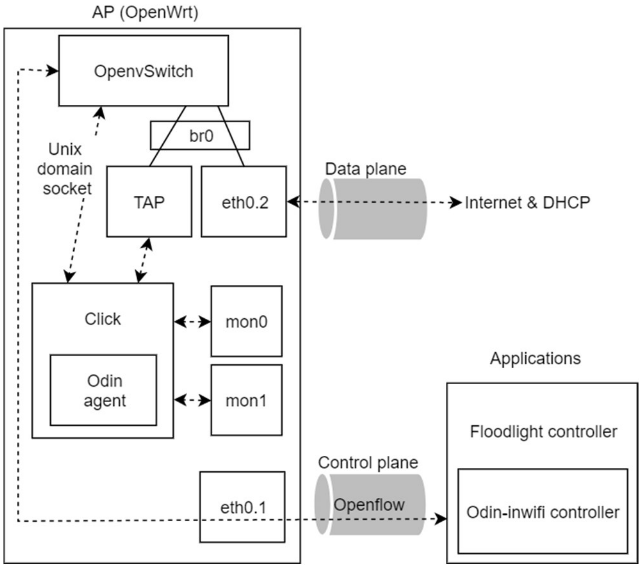


Fig. 5 Our solution design

As soon as the agent wants to connect to the Controller, it sends the packets through a unix domain socket to the Open vSwitch. An optional physical split-up between the data and the control plane has been proposed: data plane information is directed to the Internet (as well as DHCP messages) through the eth0.2, while control plane uses the second interface eth0.1. This contains OpenFlow packets, since both Odin protocol streams are embedded into OpenFlow as experimenter. As the Controller needs to know the assigned IP addresses, we also send DHCP packets to the control plane. In conclusion, this functionality allows Controller to define OpenFlow rules for each client.

IV. Testbed

This section describes the hardware and software requirements for delay, jitter (packet delay variation), packet loss rate and throughput measurements.

A. Experimental Setup

Our experimental setup for measurements consisted of one laptop, one desktop and two SOHO routers. The laptop was a 3.0 GHz Intel Core i7 running Ubuntu 14.04 LTS and functioning as a client with two D-Link DWA-127 USB wireless NICs powered with MediaTek MT7601U chip. The desktop with Odin Controller and DHCP server was a 2.5 GHz Intel Core i5 running also Ubuntu 14.04 LTS. The routers with Odin agent were MikroTik RB951G-2HnD with Atheros AR9344 chip and edited driver ath9k_htc. There was used our improved version of OpenWRT with added Open vSwitch to support OpenFlow protocol. The network element of wired network part was represented by HUB. We know that in real network topology, there is no hub. Our goal was to see performance of pure SDN architecture with wireless elements avoiding influence of other peripherals like switch, router, etc. We also wanted to evaluate our VAP implementation. We captured all data traffic with Wireshark 2.2[1] and the traffic was generated by D-ITG[2] or iperf. The testbed is shown in Fig. 6.

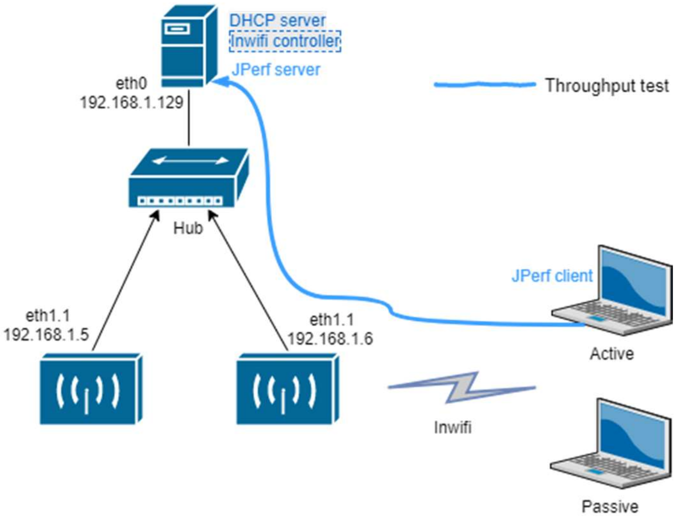


Fig. 6 Testbed

The experiments were conducted under 802.11g wireless standard on channel 11 with maximal theoretical throughput up to 54 Mb/s. In most tests, there was a large amount of external interference from other wireless networks, which can be seen in Fig. 7. Tests were performed in an indoor environment in order to achieve realistic test's conditions.

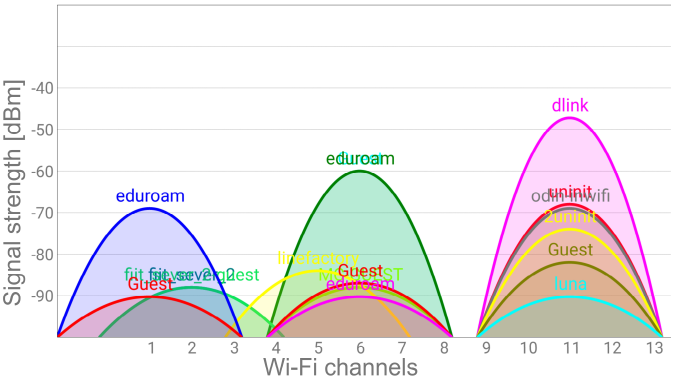


Fig. 7 WiFi signal strength in our test enviroment

B. Test cases

Test requirements were designed to show performance of our architecture. Measurements were mostly performed on a multimedia streams because they are sensitive to the delay and packet loss. Each test duration was 100 seconds, and there were several forced handovers of the client. Handovers were set according to the pedestrian model [14] and to Cisco study on Location-Aware WLAN [15]. In pedestrian model walking in a passage has velocity around 1.0 ± 0.02 m/s. From Cisco study the distance between APs should be between 11 to 22 meters in most indoor locations for real-time applications. But they also mention the overlap of wireless cells, which should be around 20%. According to our tests, the longest distance from AP where we had a Cisco mentioned minimum signal level -67 dBm suitable for real-time application was 17 meters. Applying these numbers mathematically, we have calculated an inter-access point distance to 30 meters and an estimated cell size of 17 meters. The overlap is 24%. All parameters are perfectly corresponding to theoretically needed numbers according to before mentioned Cisco study. When we consider previous calculations of distances between APs and also take into account the velocity of standard person in a passage, we can set the handover intervals. In our tests we have set them to 15 seconds. The handover is accomplished with forced move of VAP between APs. The handovers were repeated six times and the measurements eight times to exclude possible outliers and obtain results reflecting sufficient statistics.

As a first test case we have chosen a First-Person Shooter (FPS) game Quake 3 as a multimedia stream, which characterizes a good sample of a demanding real-time service. We used D-ITG to simulate a Quake 3 scenario. The goal was to achieve handover duration less than 50 ms. Such delay does not influence even the interactive delay sensitive applications [16].

[1] https://www.wireshark.org/

[2] http://traffic.comics.unina.it/software/ITG/

The second scenario is basically performance evaluation performed with JPerf, which is widely used tool for network performance measurements. The server side was at the desktop with SDN controller and client side was on active laptop (see Fig. 6).

Test scenarios were designed in order to used hardware, that it will serve reliably, and hardware load does not affect the performance of the network adapter.

V. Performance measurements

In the following section we present the results for the handover time, delay, packet loss and throughput.

A. *Quake 3 scenario*

In the first test there are two devices connected to the network, one generating data flows of interactive multiplayer FPS game Quake 3. The traffic is generated using D-ITG and the other connected device appears as passive. We set the application to do force handover between APs each 15 seconds (red lines in Fig. 9). We wanted easily comparable and reproducible scenario. We found project Wi-5 which is working also in domain of SDN and WLAN and they are extending Odin solution as we are. According to work [17] from Wi-5 project we used the same D-ITG model for Quake3 so we could straightly compare our achieved results with them. In Fig. 8 we can see that in referenced article on their improved Odin topology the delay did not occur because of the handover (8 handovers) and is generally below 35 ms. In their case the average delay is 15 ms, the jitter is 5.5 ms and there is also a packet loss rate 3.25%. They have used subjective quality estimator Mean Opinion Score (MOS) published in [18], which is based on delay, jitter and packet loss. The results are 3.73 in LAN scenario and 3.58 in Intra-region scenario.

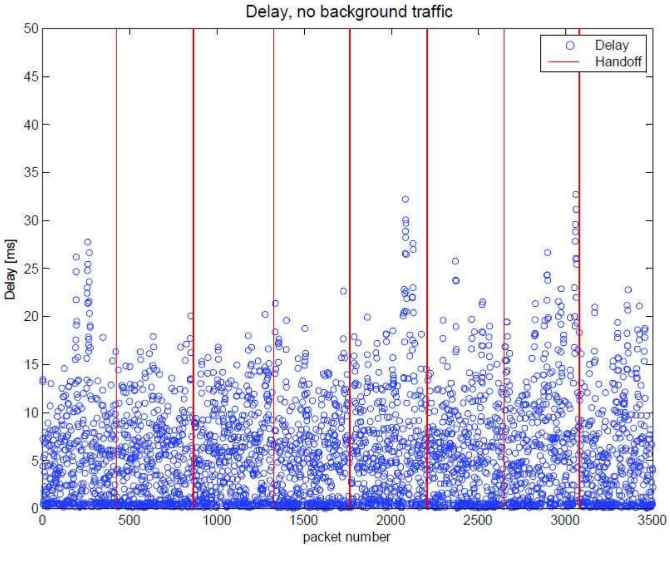


Fig. 8 Quake3 delay in referenced article [17]

Afterwards we repeated similar test described above on our solution. We have performed 8 measurements with 100 seconds duration. Across all the tests we have achieved average delay of 8.3 ms, jitter 1.5 ms and packet loss 0.54 %. In the Fig. 9 there are depicted results from one of the measurements. Analyzing the plot and log we know the maximum delay is 113 ms, average delay is 10.6 ms, jitter is 1.4 ms and 0.24 % of packets were dropped. We can also observe that the handover does not introduce any notable delay and all the delay peaks are probably caused by external interferences.

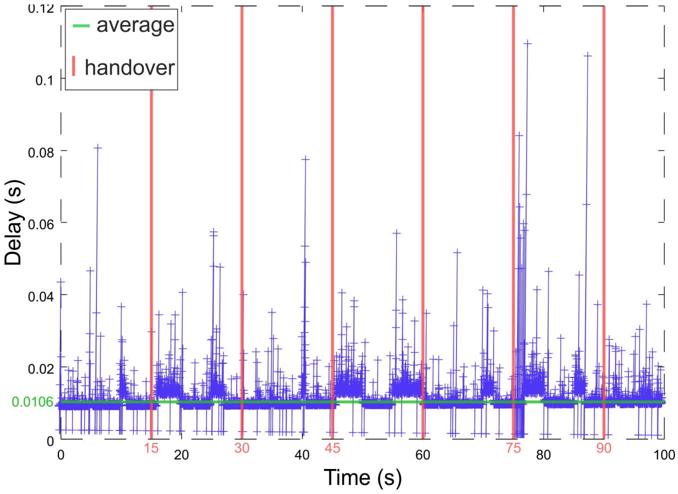


Fig. 9 Quake3 in the proposed solution

When we insert our averaged results to MOS formula for Quake model [18] it returns 4.18 (the scale is 0 to 5, the higher is better). According to literature, for VoIP a value above 3.5 is considered acceptable, some articles say that value 3 may be good, but players will go to another server if MOS is about 2 [19]. Note that the model was researched for Quake IV and we use it for Quake 3, but there are no significant differences between them in network part. To confirm our solution, we compare our results to results from Wi-5 work [17]. Our average delay is better, but there are more peaks a little bit higher. The jitter and packet loss are significantly better. After putting the results into MOS formula we have 4.18 compared to 3.73 from Wi-5 which is also considerably better. The results of the first experiment show that it is possible to maintain an acceptable level of subjective quality during a handover in a scenario in which the end device moves between different APs during an online game traffic. Therefore, selection of LVAP-based approach seems to be a good choice.

B. *Throughput scenario*

iPerf sever and iPerf client (with GUI from JPerf) were used for second scenario to simulate constant traffic at highest possible bandwidth. UDP buffer size was set to 160 KB, which is default value at iPerf server. According to Cisco study [20] the average highest achievable throughput is 25 Mb/s on 802.11g at application layer, which was ours target for end to end measurement. We measure the peak congestion with highest throughput, we set UDP bandwidth on iPerf to 54Mb/s according to maximum raw data throughput of 802.11g standard. The results shown in Fig. 10 are from four different measurements on D-Link DWA-127 wireless dongle. The handover was enforced each 15 seconds according to pedestrian scenario and the test duration was 100 seconds. The Fig. 10 shows minimal throughput decrease affected by handover process. The average throughput is 24.7 MB/s and on each of the measurements is as follows: 23.2 Mb/s, 25.9 Mb/s, 24.5 Mb/s, 25.1 Mb/s. The results show that our implementation almost does not affect maximal state of the technology throughputs at AP.

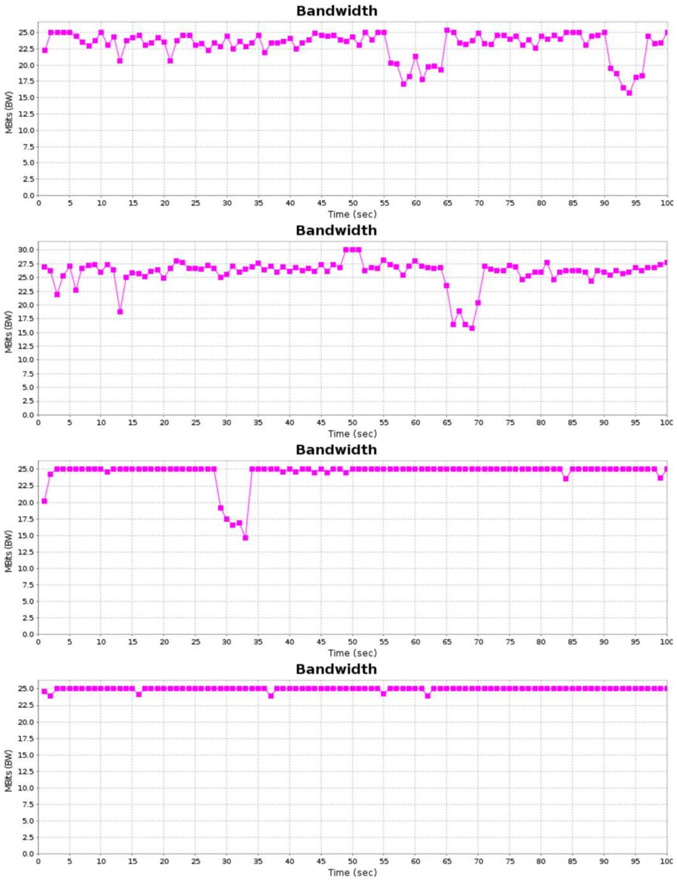


Fig. 10 Maximum throughout measurements

VI. Conclusions

In this article, we presented the improved architecture derived from Odin. According to SDN principles, we proposed the enhanced architecture what resulted to merged control channels for wireless and wired part of the network to one control channel. We proposed the extension of OpenFlow protocol for ensuring the wireless functionality and also changes in AP and SDN Controller components.

We have presented properties of the proposed solution by measurements in real implementation using two distinct scenarios. Achieved results were compared to that obtained by original Odin. The first test was a FPS game, which is a demanding real-time application and also evaluated the results with MOS value, which furthermore confirmed suitability of our solution. The second test have shown maximum achievable throughput. The tests were performed in real life environment with a large amount of external interference from other wireless networks, because we have chosen a real environment instead of environment without distracting influences to show all real results and usability in existing environments. According to Fig. 8 and Fig. 9 our handover solution does not make any significant impact on delay duration. According to results, also considering throughputs depicted in Fig. 10, we can state with satisfaction that our solution is significantly better than Odin because it is architecturally cleaner, has the same qualities and better results and finally introduces more SDN principles.

In the future work we would like to focus on optimization of handover planning and load balancing between APs. We will extend OpenFlow protocol by new OpenFlow actions, matches and packet type for 802.11 and they will be verified by Petri nets. These new features will increase manageability of wireless part of the network.

Acknowledgment

This work is a partial result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF. It is also a part of APVV-15-0731 project and VEGA 1/0836/16, KEGA 011STU-4/2017. The authors would like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

References

[1] "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016 (Revision IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[2] S. Feirer and T. Sauter, "Seamless handover in industrial WLAN using IEEE 802.11k," *2017 IEEE 26th Int. Symp. Ind. Electron.*, pp. 1234–1239, 2017.

[3] H. Ahmed and H. Hassanein, "A performance study of roaming in wireless local area networks based on IEEE 802.11r," in *2008 24th Biennial Symposium on Communications*, 2008, pp. 253–257.

[4] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 93–102, Apr. 2003.

[5] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[6] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," 2012.

[7] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in *Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12*, 2012, vol. HotSDN '12, pp. 115–120.

[8] T. Lei, Z. Lu, X. Wen, X. Zhao, and L. Wang, "SWAN: An SDN based campus WLAN framework," in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, 2014, pp. 1–5.

[9] D. Zhao, M. Zhu, and M. Xu, "Supporting 'One Big AP' illusion in

enterprise WLAN: An SDN-based solution," in *2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, 2014, pp. 1–6.

[10] R. Bencel, K. Kost'al, I. Kotuliak, and M. Ries, "Common SDN control channel for seamless handover in 802.11," in *2018 Wireless Days (WD)*, 2018, pp. 34–36.

[11] A. K. Rangisetti, H. B. Baldaniya, P. K. B, and B. R. Tamma, "Load-aware hand-offs in software defined wireless LANs," in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2014, pp. 685–690.

[12] Open Networking Foundation, "OpenFlow Switch Specification Version 1.5.1," *OpenFlow Switch Specif.*, pp. 1–283, 2015.

[13] L. P. Suresh, "Programming the enterprise WLAN: an SDN approach," Universidade Técnica de Lisboa, 2012.

[14] R. Löhner, "On the modeling of pedestrian motion," *Appl. Math. Model.*, vol. 34, no. 2, pp. 366–382, Feb. 2010.

[15] Cisco Systems, "Chapter: Best Practices—Location-Aware WLAN Design Considerations," in *Wi-Fi Location-Based Services 4.1 Design Guide*, 2014, p. 206.

[16] ITU-T, "G.114 One-way transmission time," *Ser. G Transm. Syst. MEDIA, Digit. Syst. NETWORKS Int. Teleph. Connect. circuits – Gen. Recomm. Transm. Qual. an entire Int. Teleph. Connect.*, pp. 1–20, 2003.

[17] J. Saldana, J. L. de la Cruz, L. Sequeira, J. Fernandez-Navajas, and J. Ruiz-Mas, "Can a Wi-Fi WLAN support a first person shooter?," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 2015, pp. 1–3.

[18] A. F. Wattimena, R. E. Kooij, J. M. van Vugt, and O. K. Ahmed, "Predicting the perceived quality of a first person shooter: the Quake IV G-model," in *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games - NetGames '06*, 2006, pp. 1–4.

[19] M. Dick, O. Wellnitz, and L. Wolf, "Analysis of factors affecting players' performance and perception in multiplayer games," in *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games - NetGames '05*, 2005, pp. 1–7.

[20] J. Florwick, J. Whiteaker, A. C. Amrod, and J. Woodhams, *Wireless LAN Design Guide for High Density Client Environments in Higher Education-Cisco Guide*. San José: Cisco Press, 2017.