Sādhanā (2019)44:243
https://doi.org/10.1007/s12046-019-1229-3

Testbed evaluation of a seamless handover mechanism for an SDN-based enterprise WLAN

ARKADEEP SEN and KRISHNA M SIVALINGAM*

Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India
e-mail: arkadeep.sen87@gmail.com; krishna.sivalingam@gmail.com; skrishnam@iitm.ac.in

MS received 20 July 2019; revised 29 October 2019; accepted 30 October 2019

Abstract. Many modern-day applications require seamless connectivity to provide a good quality experience to mobile users. Thus, mobility management mechanism becomes a crucial aspect of all the wireless technologies used to connect to the Internet. In this letter, we present the testbed implementation details of a Network Address Translation (NAT)-based, client-unaware, seamless handover mechanism for a Software-Defined Enterprise WLAN framework. The results from the testbed implementation corroborate that the handover mechanism can provide uninterrupted connectivity during a handover process.

Keywords. Enterprise WLANs; Mobility Management; Software-Defined Networking.

1. Introduction

Mobility management in enterprise WLANs is of two types: mobile device driven or distributed [1, 2] and network driven or centralized [3, 4]. In [5] and [6], details are provided about other related works on mobility management in enterprise WLANs. We do not present additional details about these works here. The mobile-device-driven handover mechanism is used in traditional WLAN. This type of handover mechanism suffers from the disconnection of the on-going sessions during handover, as the mobile device first disconnects from the currently connected access point (AP) and then connects with an appropriate AP. This can be avoided by a network-driven handover mechanism; however, the packets destined for the roaming mobile device should be re-routed to the new AP after handover or else they might get dropped.

In this letter, we present the testbed implementation details of a centralized, NAT-based, client-unaware, seamless handover mechanism, proposed in [5], which provides uninterrupted connectivity during handovers. The design of the centralized enterprise WLAN framework is enabled by Software-Defined Networking (SDN) principles. This design provides a global view of the entire WLAN at the logically central SDN Controller. The NAT-based handover mechanism [5] has been proposed using the SDN-based enterprise WLAN framework proposed in [6]. The handover mechanism periodically checks if any mobile device is moving away from its connected AP and, if so, then it initiates handover for all those roaming mobile devices. For correct packet delivery after the handover, the Controller updates NAT entries corresponding to the roaming mobile device at the appropriate network devices. Since the handover is detected beforehand and an appropriate AP is also chosen, the handover delay is significantly reduced. The SDN framework is implemented in a testbed environment, along with the NAT-based handover mechanism. The framework is extended with additional functionalities to support the NAT-based handover mechanism. We have also modified the handover mechanism to immediately detect and initiate handover for each roaming mobile device rather than doing it periodically.

2. Testbed implementation

This section explains the implementation of the prototype of the SDN-based enterprise WLAN framework [6], along with the NAT-based handover process [5]. The source code of the testbed implementation is publicly available at https://github.com/arkadeepsen/Handover.

2.1 *SDN-based enterprise WLAN framework*

Figure 1 presents the network architecture of the testbed implementation of the SDN-based framework. All the APs in the enterprise WLAN connect with the SDN Controller using OpenFlow protocol. The SDN Controller is called the Wireless Controller (WiC) as it orchestrates the entire WLAN. The MAC management functionalities are split between the APs and the WiC. Since the APs do not provide all the functionalities, they are called Light APs. As a result of such a design, the WiC will have a global view of

*For correspondence
Published online: 30 November 2019

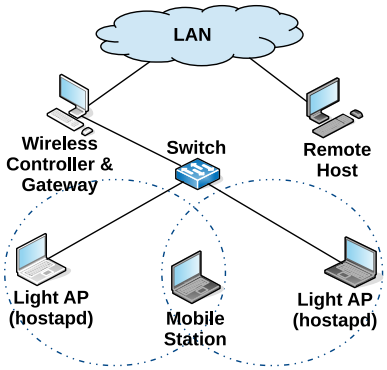


Figure. 1. Network architecture.

the entire WLAN. The Light APs and the WiC are implemented on Linux-based PCs. The Gateway router connects the enterprise WLAN to the outside network (LAN in this case). It is also SDN enabled and uses OpenFlow protocol to connect with the WiC. The Gateway router is implemented on the Linux system on which the WiC runs. The Light APs are connected to the Gateway router and the WiC through a switch. The WiC and the Gateway can also be implemented on two different systems, in which case both the systems should be connected to the switch. A Linux-based laptop is used as the mobile station. All the Light APs are configured to operate on the same channel. They are also configured with the same Service Set Identifier (SSID) and Basic Service Set Identifier (BSSID). Due to such configurations, the mobile devices will not be able to differentiate between the Light APs and it will seem that only a single AP exists. The WiC decides which Light AP will serve a mobile device based on the Probe Request frames initially sent by the mobile device when it searches for the available APs. This Light AP is designated as the Home AP of the mobile device. Any further communication with the mobile device happens with this Light AP. During a handover, when a mobile station (STA) is migrated from one AP to another by the WiC, the STA will not be able to detect the migration because of the configurations and it will seem to the STA that it is still connected to the same AP. Since the STA will be unaware of the change in connectivity, the on-going application sessions running on the STA will not encounter any disconnection during the handover.

2.2 *Implementation details*

Table 1 summarizes the additions/changes made to the different software used for the implementation of the SDN-based framework and the NAT-based handover mechanism.

2.2a *Light APs*: A user space program called *hostapd* [7] is used to create software APs on laptops. AP and authentication server functionalities are provided by *hostapd*. As DHCP server functionality is not provided by *hostapd*, a tool called *dnsmasq* [10] is used for setting up DHCP server and DNS cacher on the Light APs. Thus, using *hostapd* and *dnsmasq*, along with *iptables* (for forwarding packets to the NAT module), a Linux box can be turned into a software-based WiFi AP. Packets arriving at the Light APs are queued up by specifying *iptables* rules. These packets are then handled using the API provided by a userspace library called *libnetfilter_queue* [11]. For every packet received on the wireless interface, an OpenFlow Experimenter message is sent to the WiC containing the source and destination IP addresses and port numbers. The WiC assigns a unique port number for the destination IP address and port number pair and sends the corresponding NAT entry to the Light AP via another OpenFlow Experimenter message. Every subsequent packet in the same flow will match this NAT entry. After applying NAT, each packet will have the Light AP's IP address as the source IP address and the unique port number as the source port number. The WiC stores the entries by mapping the STA address to all its NAT entries. Packets arriving at the interface of the Light AP connected to the switch are matched with the NAT entries. If a packet matches an entry, then NAT is applied on the packet appropriately by consulting the matching entry.

To support the *OpenFlow* protocol and to connect to the WiC, an extra module is added to *hostapd*. This module sets up a TCP connection with the WiC and handles all the incoming OpenFlow messages from the WiC as per the OpenFlow specification. This module also sends appropriate Experimenter messages to the WiC for various events related to the WLAN services. NAT functionality is also taken care of by this module. The *Click* modular router [8] is also run at the Light AP to capture all the data frames that have the destination MAC address as the common BSSID, configured on all the Light APs. Once captured, the received signal strengths of these data frames are written to files, which are read by the module added to *hostapd* and are sent to the WiC via OpenFlow Experimenter messages. Based on these received signal strengths, the WiC takes the handover decisions.

2.2b *Gateway router*: A Linux system is used to operate as the Gateway router with NAT functionality. For every new packet received on the interface connected to the switch, a new NAT entry is created and stored. Every subsequent packet in the same flow will match this NAT entry and the destination IP address of those packets will be changed to the Gateway router's IP address. The Gateway router also connects with the WiC using the OpenFlow protocol. The WiC instructs the Gateway to change the value of the source IP address field of certain NAT entries during the handover process. Packets arriving at the interface connected to the LAN are matched with the NAT entries. If a packet matches an entry, then NAT is applied on the packet appropriately by consulting the matching entry.

Table 1.. Summary of software additions/changes.

Software	Additions/changes
hostapd [7]	Added *OpenFlow* support and NAT functionality; modified IEEE 802.11 MAC services according to the SDN-based framework
Click router [8]	Added module to get the received signal strengths of captured data frames and write them in files used by hostapd
Gateway router NAT module	Added *OpenFlow* support and NAT functionality with required modifications
Floodlight Controller [9]	Added module to handle Experimenter OpenFlow messages from Light APs and Gateway router, and the handover process for NAT operation mode of the APs

2.2c *WiC*: The WiC is implemented using the Floodlight OpenFlow Controller [9]. The WiC communicates with the Light APs and the Gateway router using the OpenFlow protocol. To implement the SDN-based framework, we added a module to the Floodlight Controller to handle all the Experimenter messages sent by the Light APs and the Gateway router. This module stores all the information about the mobile devices connected with the Light APs, including the NAT entries. The handover mechanism is also implemented in this module.

2.3 *NAT-based handover mechanism*

The handover process is triggered whenever the WiC detects that an STA is moving away from its corresponding Home AP. Home AP is the Light AP with which the STA is currently connected.

The signalling required for the NAT-based handover mechanism is described in figure 2. All the Light APs keep track of the last received signal strength from an STA. If a non-Home AP of the STA receives a data frame from the STA, then it will check whether the signal strength of the received data frame is greater than the last received signal strength from the STA. If so, then the Light AP will send the signal strength information to the WiC. Upon receiving this information, the WiC will ask the Home AP of the STA to report the last received signal strength from the STA. The Home AP will comply with the request by sending the same to the WiC.

After receiving the information from the Home AP of the STA, the WiC will check the handover criterion. The handover is initiated if the last received signal strength from the STA is greater at the non-Home AP as compared with that at the Home AP of the STA. Subsequently, all the NAT entries, if any, corresponding to the STA will be added to the new AP. This is done so that the new AP can properly forward all the incoming and outgoing packets of the existing flows corresponding to the STA by applying NAT.

The WiC will then inform the Home AP to remove the association of the STA and set its state to *NOT AUTHENTICATED*. The new AP will now be designated as the Home AP of the STA. The WiC will inform the new AP to add the STA and set its status to *ASSOCIATED*.

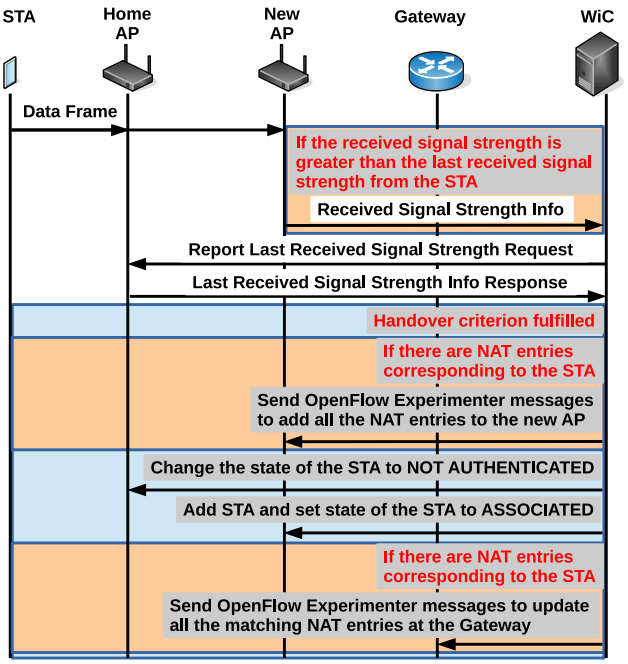


Figure. 2. Handover process for NAT operation mode.

The WiC will next update all the NAT entries, if any, at the Gateway router corresponding to the STA. The source IP address of those NAT entries will be changed to the IP address of the new Home AP of the STA. For any incoming packets destined for the STA, the Gateway router will now apply NAT by changing the destination IP address of those packets to the new Home AP and forward the packets to it. All the messaging among the Light APs, the Gateway and the WiC is accomplished by sending appropriate OpenFlow Experimenter messages.

3. Testbed evaluation

This section presents the results of the experiments conducted using the testbed described in the previous section. All *hostapd* instances are configured to operate in IEEE 802.11g mode. In the experiments, instantaneous throughput is measured for TCP and UDP applications. Two different scenarios are considered for the

experimentation: sender application configured on the STA while receiver application configured on the remote host, and vice versa. The sender applications transmit packets at a rate of 80 kbps (1000 bytes every 100 ms) and send a maximum of 200,000 bytes. The performance of the NAT-based handover for the SDN-based framework is compared to that of the handover in traditional enterprise WLAN. For the traditional enterprise WLAN, the unmodified version of *hostapd* is used along with *dnsmasq* and default NAT functionality of *iptables*.

Figures 3 and 4 present the instantaneous throughput at the receiver application for the scenarios mentioned earlier. In all the cases, the handover process starts at around 5 s and ends at around 10 s.

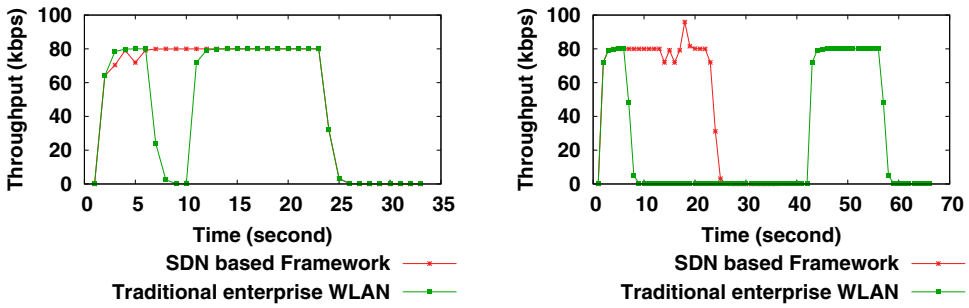
As seen in figures 3a and 4a, the on-going UDP communications get interrupted during the handover, for the case of the handover process for the traditional enterprise WLAN. Once the handover process is over, only then the communication resumes.

For the case of the sender application running on the STA, the STA will send the packets to the new AP after the handover. The new AP will then forward the packet to the remote host via the Gateway router. For the case of the handover process for the traditional WLAN, the communication will get interrupted during the handover, since the STA will disconnect from the previous AP and connect to the new AP.

When the sender application runs on the remote host, in the case of the handover process for traditional WLAN, after the handover, the packets from the sender application will still be forwarded to the previous AP by the Gateway router and will get dropped. The UDP receiver application running on the STA will detect a break in communication. To reconnect, it will then send a request packet to the UDP sender application via the new AP. When the Gateway router forwards the request packet to the remote host, it will update the corresponding NAT entry. Henceforth, any UDP packet sent from the remote host to the STA will be forwarded to the new AP by the Gateway router.

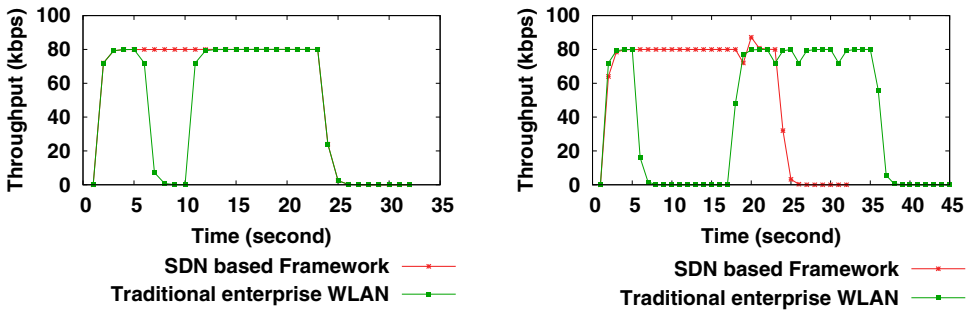
Since, in both the scenarios, the UDP sender application cannot detect any break in communication, it will keep on sending packets during the handover process, and these packets will get dropped. The finish time of the UDP sender application does not change because of this reason.

As seen in figures 3b and 4b, the TCP sessions get interrupted during the handover, for the case of the handover process for the traditional enterprise WLAN. However, the TCP applications take more time to establish a new session and the communication resumes only after that. The TCP sender and receiver applications detect a break in the communication during handover and they close the session in both the scenarios. After this, the TCP client, which runs on the STA, tries to establish a new session with the TCP server application. The longer it takes for the new



(a) UDP Throughput. **(b)** TCP Throughput.

Figure. 3. Throughput when the sender application runs on the mobile node or station.



(a) UDP Throughput. **(b)** TCP Throughput.

Figure. 4. Throughput when the sender application runs on the remote host.

session to get established after the handover, the longer will be the break in communication. Only after a new session is established, the sender application, in both the scenarios, resumes sending the remaining packets. Due to this reason, the finish time of the TCP sender application increases in both the scenarios.

In the case of the handover process for the SDN-based framework, the on-going sessions continue uninterrupted during the handover in both the scenarios. Moreover, there is no drop in the instantaneous throughput at the receiver application even during the handover process, for both TCP and UDP applications. The WiC moves the association of the STA from the previous AP to the new AP during the handover, in both the scenarios. The WiC also updates the corresponding NAT entries at the new AP as well as the Gateway router. As a result, the communication continues uninterrupted for both the TCP and UDP applications and the finish time of all the sender applications remains the same in both the scenarios.

Thus, it can be seen that the NAT-based handover mechanism for the SDN-based framework is able to provide seamless mobility for both scenarios. The testbed results show that such constant throughput during handover is, in fact, achievable.

4. Conclusions

In this letter, we presented details of the testbed implementation of a centralized, client-unaware, seamless, NAT-based handover mechanism for an SDN-based enterprise WLAN framework. The handover mechanism is initiated by the SDN Controller when it detects an imminent handover of a mobile device. After the handover, the SDN Controller updates NAT entries at appropriate network devices for the successful delivery of packets. The testbed evaluation results show that the handover method is able to achieve seamless mobility and the on-going sessions continue uninterrupted even during the handovers.

Acknowledgements

This work was supported in part by a Mid-Career Institute Research and Development Award (IRDA) Grant from IIT Madras (2017–2020) and the Department of Science and Technology (DST) Grant (EMR/ 2016/003016) from the Government of India (2017–2020).

References

[1] Chen W, Yen L, Chuo C, Heish T and Tseng C 2017 SDN-enabled session continuity for wireless networks. In: *Proceedings of IEEE ICC*, pp. 1–6
[2] Feirer S and Sauter T 2017 Seamless handover in industrial WLAN using IEEE 802.11k. In: *Proceedings of IEEE ISIE*, pp. 1234–1239
[3] Zeljković E, Marquez-Barja J M, Kassler A, Riggio R and Latré S 2018 Proactive access point driven handovers in IEEE 802.11 networks. In: *Proceedings of CNSM*, pp. 261–267
[4] Zhao D, Zhu M and Xu M 2014 SDWLAN: a flexible architecture of enterprise WLAN for client-unaware fast AP handoff. In: *Proceedings of ICCCNT*, pp. 1–6
[5] Sen A and Sivalingam K M 2019 A NAT based seamless handover for Software Defined Enterprise WLANs. In: *Proceedings of IFIP WWIC*, pp. 78–90
[6] Sen A and Sivalingam K M 2015 An SDN framework for seamless mobility in enterprise WLANs. In: *Proceedings of IEEE PIMRC*, pp. 1985–1990
[7] hostapd. http://w1.fi/hostapd/
[8] The Click modular router. https://github.com/kohler/click
[9] Project Floodlight. http://www.projectfloodlight.org/floodlight/
[10] dnsmasq. http://www.thekelleys.org.uk/dnsmasq/doc.html
[11] Welte H libnetfilter_queue. http://www.netfilter.org/projects/libnetfilter_queue/.