HAL open science

A NAT Based Seamless Handover for Software Defined Enterprise WLANs

Arkadeep Sen, Krishna M. Sivalingam

To cite this version:

Arkadeep Sen, Krishna M. Sivalingam. A NAT Based Seamless Handover for Software Defined Enterprise WLANs. 17th International Conference on Wired/Wireless Internet Communication (WWIC), Jun 2019, Bologna, Italy. pp.78-90, 10.1007/978-3-030-30523-9_7 . hal-02881741

HAL Id: hal-02881741

https://inria.hal.science/hal-02881741v1

Submitted on 26 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A NAT based seamless handover for Software Defined Enterprise WLANs ⋆

Arkadeep Sen and Krishna M. Sivalingam

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

Abstract. Various applications used by mobile users today need seamless connectivity for providing a good quality user experience. Enterprise Wireless Local Area Network (WLAN) is one of the technologies used by mobile devices to connect to the Internet in several environments. For providing seamless connectivity and communication, mobility management becomes an important aspect of such deployments. In this paper, we propose a client-unaware handover process for NAT (Network Address Translation) operation mode of the access points in a Software Defined Networking (SDN) based enterprise WLAN framework. The proposed mechanism has been implemented in simulation and the results show that the proposed mobility management mechanism is able to achieve seamless handover and provide uninterrupted connectivity and communication to the mobile devices.

Keywords: Enterprise Wireless LANs · Mobility Management · Software Defined Networks.

1 Introduction

With the exponential increase in the number of mobile devices and applications, the mobile data traffic for the Internet has increased manifold. The mobile users use a myriad of mobile applications on a day-to-day basis and many such applications require seamless connectivity for a good quality user experience. Mobility management, thus, becomes a very important aspect of any wireless technology.

Enterprise wireless local area network (WLAN) technology provides Internet connectivity to mobile devices in several environments, such as building, campus. The deployment of such a network comes with its own set of requirements such as network security, load balancing, mobility management, etc. In this paper, we delve into the mobility management aspect of the enterprise WLANs.

⋆ This work was supported in part of by India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services (IU-ATC). This work was also supported by an Institute Research & Development Awards (IRDA) grant from IIT Madras (2017-2020) and a Department of Science & Technology (DST) grant (EMR/2016/003016) from Government of India (2017-2020).

In WLANs, the connection initiation and termination is taken care of by the mobile devices [2]. As a result, the handover process is mobile device driven. When the mobile device moves away from the current AP and the signal strength is not strong enough for communication, the mobile device disconnects from the current AP. It then searches for a suitable AP and finally connects with a new AP. This results in an interruption in the communication. Though IEEE 802.11r [1] reduces the handover delay, disruption in communication will still be there.

Thus, mobility management in WLANs is distributed in nature and because of this mobile devices encounter unnecessary disconnection while moving across the entire coverage area. A centralized mobility management mechanism, having a global view of the entire WLAN, can solve this problem. The network can track the movement of mobile devices and can also detect an imminent handover. Instead of the mobile devices, the network can initiate the handover and, as it has a global view, it can also choose the AP which the mobile device should connect with after the handover. This process will significantly reduce the handover delay, and the ongoing communication will not get interrupted.

The advent of SDN [12–16] has paved a way for centralized network design and control. This has been achieved by decoupling the data plane and the control plane of a switch (router) and placing the control plane in a centralized controller. Such a design allows the controller to have a global view of the entire network. SDN also enables network programmability by allowing the deployment of customized control applications at the controller. To reap the benefits of such a design, we have proposed a client-unaware, seamless handover process for NAT (Network Address Translation) operation mode of the access points (AP) in a Software Defined Networking (SDN) based enterprise WLAN framework. In this paper, we have considered the SDN based framework proposed in [17]. In [17], a non-NAT handover process is also proposed which creates tunnels in order to correctly deliver packets to the roaming STA after the handover. We have only used the SDN based framework in this paper. Additionally, we have extended the framework by adding features in order to aid the proposed NAT based handover mechanism. The additional features are providing a unique transport layer port number across all Light APs for the NAT entry of a flow by the SDN Controller, and tracking as well as storing of the NAT entries for the flows of each STA at the SDN Controller. The proposed handover process has been implemented in OMNeT++ simulator [19] and the results from the simulation show that the handover process is able to achieve seamless handover and provide uninterrupted communication to the mobile devices.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 presents the proposed handover mechanism. Section 4 describes the simulation-based performance results. Section 5 concludes the paper.

2 Related Work

In this section, we discuss some of the proposed mobility management schemes for WLANs which try to reduce the delay during a handover process.

Handover management mechanisms for WLANs are proposed in [4, 6, 7, 9, 10]; however, the handover processes described in the papers are initiated by the mobile devices. As a result of this type of handover process, the communication of the ongoing sessions is disconnected and it can resume only after the handover is completed.

Mobility management mechanisms proposed in [5, 8, 18, 20, 21] are initiated by the network instead of the mobile devices. In [8], a mechanism is presented to migrate all the associated mobile devices of an AP to another one. However, this mechanism does not take into consideration that some of the mobile devices may not be in the coverage area of the second AP, thus resulting in disconnection of those mobile devices after the migration. Moreover, the proposed mechanism does not support the handover of an individual mobile device.

The Odin project [18] uses a Light Virtual AP (LVAP) for representing each mobile device. Each LVAP is configured with a unique basic service set identifier (BSSID). It is stored at the physical AP with which the mobile device is connected. In the paper, handover of a mobile device is handled by removing its LVAP from one AP and adding the LVAP to another one AP. Though this mechanism achieves client-unaware handover, it does not re-route the packets, which have the roaming mobile device as the destination, from the old AP to the new AP. As a result, the on-going communication may get disconnected, in certain situations, until the communication is re-established by the application, running on the mobile device. The paper also proposes unicasting of Beacon frames to each mobile device. For large WLANs, this will increase the wireless traffic leading to collisions with other frames.

In [20], a similar LVAP based approach is discussed. A handover mechanism is proposed which takes into consideration the signal strength, the load on the APs and the location of the mobile devices during taking a decision on the handover of the mobile devices. However, [20] has similar shortcomings as that of [18].

In [21], an SDN based framework is proposed, where the SDN controller detects an imminent handover and modifies flows at the old and new APs appropriately. The controller then proactively adds flows to the SDN enabled wired backbone to correctly deliver the data packets having the mobile device as the destination. This proactive route update may become a bottleneck for large WLANs. This is because of the added overhead of storing all the alternate paths at the controller and the number of flow modification messages required to be send for the route update.

In [5] Croitoru et al. propose that to have seamless mobility, the mobile devices should connect with all the available APs and the traffic should be split across them by using the Multi-path TCP (MPTCP) protocol. As the paper only takes care of TCP traffic, UDP traffic will not have an uninterrupted communication during handover. Moreover, the paper suggests multiple client-side modifications so that the throughput does not get severely affected in some scenarios. As a result, the clients without such modifications will suffer from low throughput.

In our proposed solution, we make modifications at the AP side only. The mobile devices do not require any changes. The SDN controller, in the proposed solution, detects imminent handovers and instructs the APs to initiate the handover processes. The data packets having the mobile devices as the destination are re-routed to the new APs by updating NAT entries after the handovers are completed. The proposed solution does

not require the complete wired backbone to be SDN enabled. Moreover, the number of NAT entry updates remains the same even when scaling to large WLANs. Thus, we efficiently address the shortcomings of the above-mentioned work in our proposed solution.

3 Proposed Handover Mechanism for SDN based enterprise WLAN framework

This section presents the proposed handover mechanism for SDN based enterprise WLAN. We have considered the SDN based enterprise WLAN framework proposed in [17]. Figure 1 depicts the SDN framework for the proposed NAT based handover mechanism.

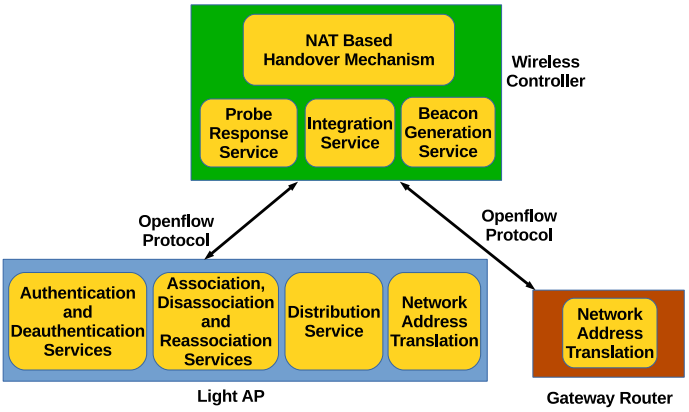


Fig. 1. Software Defined Enterprise WLAN framework.

In the framework, all the access points (APs) are SDN enabled. They all connect to the SDN Controller called the Wireless Controller (WiC) using the OpenFlow [16] protocol. The APs are called Light APs as the MAC management functionalities are split between the APs and the WiC. The WiC handles the Authentication and Deauthentication Services, the Association, Disassociation and Reassociation Services, and the Distribution Service and the Light APs handle the Probe Response Service, the Integration Service, and the Beacon Generation Service. All the Light APs are configured to operate on the same channel. They are also configured with the same SSID and BSSID. Consequently, it will seem to the mobile stations (STAs) that only single AP is available. All the configurations are done centrally through the WiC.

The WiC has a global view of the entire enterprise WLAN because of the splitting of the MAC management functionalities. As a result of this design, the WiC can track and detect any imminent handover. Thus, the WiC can initiate the handover processes for the roaming STAs. Moreover, as the STAs are unaware of the availability of multiple APs, even after the completion of the handover, the STA will not be able to detect any

change in its connectivity. As a result, the on-going sessions at the STAs will continue uninterrupted even during the handover. Additional features, such as providing a unique transport layer port number across all Light APs for the NAT entry of a flow by the WiC, and tracking as well as storing of the NAT entries for the flows of each STA at the WiC, are added to the SDN based framework proposed in [17]. These additional features aid in the operations of the proposed NAT based handover mechanism.

NAT based Handover Mechanism In this handover process, the WiC periodically checks whether any STAs are moving away from their corresponding Home APs and if so then the WiC initiates handover for all those STAs. The Light AP, with which an STA is currently connected, is called the Home AP of the STA. For this handover process, the Gateway router, which connects the enterprise WLAN with the Internet, should also be OpenFlow compliant and should connect with the WiC. The Light APs, as well as the Gateway router, implement the NAT functionality.

Whenever a Home AP receives the first packet of a new flow from an STA, it contacts the WiC for a unique port number across all the Light APs for the NAT entry of the flow. The WiC will assign a unique port number for the flow and inform the Home AP about it. The WiC will also map the STA address to the NAT entry containing the source and destination IP addresses, the source and destination port numbers and the unique port number. The Home AP will also store the same NAT entry containing all the information. It will then apply NAT on the packets of the flow by changing the source IP address of the Home AP's IP address and the source port number to the unique port number.

Whenever a data frame from any STA is received by a Light AP, including its Home AP, the signal strength of the frame is reported to the WiC by the Light APs. For each STA, the WiC keeps track of the Light AP, which is currently receiving the maximum signal strength from that STA.

The WiC periodically calls the handover process, described in algorithm 1, and checks the handover criterion for each STA. That is, if the Home AP of the STA is not the Light AP (max AP) currently receiving the maximum signal strength from that STA, then the handover process for the STA is initiated.

If there exist any NAT entries corresponding to the STA at the WiC, it will send OpenFlow Experimenter messages to add them to the max AP. This will ensure any incoming packets from the Gateway router, belonging to any existing flows corresponding to the STA, will be properly routed by the max AP by applying NAT.

After this, the STA has to be migrated to the max AP from the Home AP without directly involving the STA. For achieving this, the Home AP and the max AP will be instructed by the WiC to change the state of the STA to *Not Authenticated* and *Associated* state respectively. The max AP now becomes the Home AP of the STA. As all the Light APs are configured with same SSID and BSSID, the STA will not detect any change in its connectivity after these operations.

The WiC then will send OpenFlow Experimenter messages to the Gateway router to update the NAT entries corresponding to the STA. The source IP address of those NAT entries will be changed to the IP address of the max AP. After the changes are made, if any packet, belonging to any existing flows corresponding to the STA, comes to the

Algorithm 1 NAT based Handover Mechanism

```
procedure HANDOVER
    for each STA do
        if STA→HomeAP ≠ STA→maxAP then
            for each NAT entry for STA do
                Add NAT entry to STA→maxAP
            end for
            Change STA state to NOT_AUTHENTICATED in STA→HomeAP
            Change STA state to ASSOCIATED in STA→maxAP
            Set STA→HomeAP to STA→maxAP
            for each NAT entry for STA do
                At the Gateway update source address of NAT entry to
                address of STA→maxAP
            end for
        end if
    end for
end procedure
```

Gateway router from a remote host, the Gateway router will route the packet to the max AP which, in turn, will route the packet to the STA by appropriately applying NAT.

4 Simulation-Based Performance Study

The proposed handover process for NAT operation mode of APs has been implemented in the OMNeT++ simulator [19] (version 5.0). The INET framework [3] (version 3.4.0), which is an open-source OMNeT++ model suite for wired, wireless and mobile networks, and an OpenFlow extension to the INET framework [11] are used for the implementation.

4.1 Simulation Setup

The STAs and the APs are all configured to operate in IEEE 802.11n mode on the 2.4 GHz frequency band at a maximum data rate of 600 Mbps. All the APs connect to the remote hosts via the Gateway router. The STAs always move through the coverage area of the APs. Each STA randomly chooses a direction (right or left) towards which it will move. The STAs move in the chosen direction at a speed of 10 m/s until they reach an end of the simulation area and then start moving in the opposite direction with the same speed. This type of mobility model is chosen to ensure the occurrence of at least one handover per STA during the simulation.

We have compared the performance of the proposed handover process with the handover processes of the traditional enterprise WLAN and the Odin framework [18]. For the purpose of simplifying the routing process, static routes are pre-installed in all the network elements wherever required.

UDP and TCP applications are set up on the STAs and the remote hosts. Throughput, packet delivery ratio, and delay are measured for the performance study. Each STA

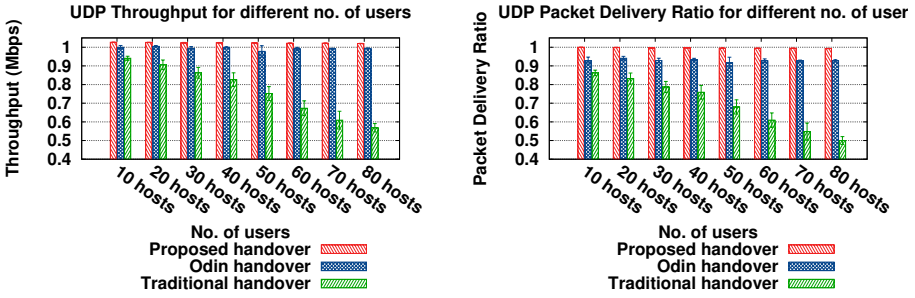
sends a request to a remote host and the remote host sends traffic to the STA at the rate of 1.024 Mbps. In one simulation scenario, the STAs send UDP traffic request and in another one, the STAs send TCP traffic request. Each STA sends the request at a randomly chosen time between 5 seconds and 10 seconds of simulation time. The remote hosts, after receiving the request, keep sending the traffic till the end of the total simulation time. The sender application continues to send packets even if there is a disconnection during the handover process. If the receiver application detects that there is a disconnection, to reconnect, it sends a request packet to the sender application. The request is sent again because in the cases of traditional enterprise WLAN and Odin framework, there might occur disconnections during handovers. If the receiver application does not send a request packet after detecting a disconnection, then the subsequent packets sent by the remote host will not reach the STA. All the experiments are run for a total of 45 simulation seconds to ensure the occurrence of at least one handover per STA during the simulation.

4.2 Performance Evaluation Results for fixed number of APs

The parameters used for running the next set of simulations are summarized in table 1.

Table 1. Simulation parameters for fixed no. of APs

Parameter	Value
No. of APs	10
No. of mobile stations	varies from 10 to 80
Speed of mobile stations	10 m/s
Traffic at each mobile station	1 UDP application of 1.024 Mbps 1 TCP application of 1.024 Mbps



(a) Average UDP Throughput. (b) Average UDP Packet Delivery Ratio.

Fig. 2. Throughput and Packet Delivery Ratio for UDP applications for varying number of users.

Figures 2a and 2b present the average UDP throughput and packet delivery ratio experienced by the STAs. The UDP throughput and packet delivery ratio decrease as the number of STAs increases because, as the number of STAs increases the chance of interference and collision increases, resulting in higher packet loss. The average UDP throughput for the proposed handover process reaches 1.024 Mbps for 10 STAs and the reduction in throughput is almost negligible as the number of STAs increases from 10 to 80. From the packet delivery ratio plot, we can see that for the proposed handover process negligible loss is experienced for the case of 10 STAs and as the number of STAs increases, the packet delivery ratio remains almost same. This shows that the proposed handover process is able to provide seamless handover for varying number of STAs and as a result, high throughput and high packet delivery ratio can be achieved. For the case of Odin framework, though for most of the cases the average throughput is almost 1 Mbps, the packet delivery ratio varies between 92% and 94%. This shows that though the handover process in Odin is able to reduce the handover delay, it fails to provide seamless handover, which results in packet loss during the handover. For the traditional enterprise WLAN framework, the throughput considerably reduces as the number of STAs increases. Even for the case of 10 STAs, the average throughput is below 1 Mbps. Similar results can be seen from the packet delivery ratio plot. The packet delivery ratio, for the case of 80 STAs, reduces to almost almost 50%. This shows that during handover there is high packet loss in the case of the traditional handover process and it increases with the increase in the number of STAs.

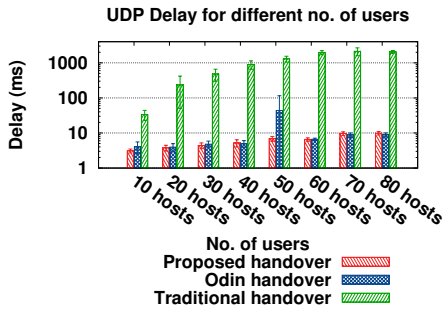
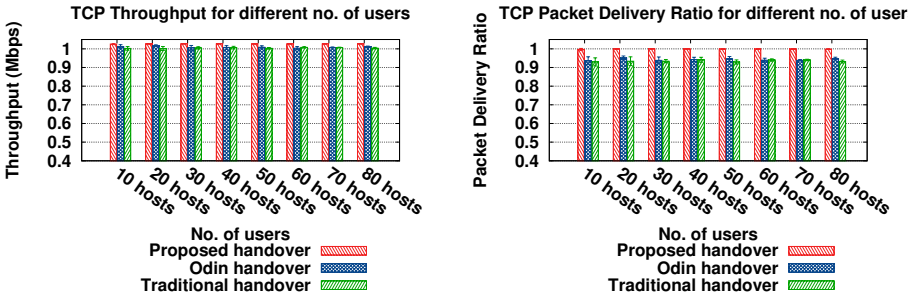


Fig. 3. Average UDP Delay for varying number of users.

Figure 3 presents the per packet average UDP delay experienced by the STAs. It can be seen that the average delay increases as the number of STAs increases. Except for the case of 50 STAs, the average delay experienced by the STAs for the proposed handover process and the handover process in Odin framework are comparable (ranges between 3 ms and 10 ms). For the case of 50 STAs, the average delay experienced for the handover process in Odin framework (approx. 44 ms) is much higher than that of the proposed handover process (approx. 6 ms). However, for the case of traditional enterprise WLAN, the delay experienced is significantly high compared to that of the other two handover processes and ranges between 33 ms and 2.13 s. This shows that

both the proposed handover process and the handover process for the Odin framework reduce the handover delay because of which the average delay experienced by the UDP applications is less than that of the handover process for traditional enterprise WLAN.



(a) Average TCP Throughput. (b) Average TCP Packet Delivery Ratio.

Fig. 4. Throughput and Packet Delivery Ratio for TCP applications for varying number of users.

Figures 4a and 4b present the average TCP throughput and packet delivery ratio per application experienced by the STAs. The TCP throughput and packet delivery ratio remain high for all the handover processes as TCP retransmits all the dropped packets, due to handover and interference, within a very short period of time. As a result, the overall average throughput and packet delivery ratio remain high in spite of packet loss during handover. We can see that even though the average TCP throughput for all the handover processes remains high (around 1 Mbps), the average throughput achieved for the proposed handover process is highest for all the cases. For all of the cases, it achieves a throughput of 1.024 Mbps. We can see that the packet delivery ratio achieved for the proposed handover process is also highest for all the cases with negligible packet loss. This shows that the proposed handover process is able to provide seamless handover for varying number of STAs and as a result, high throughput and high packet delivery ratio can be achieved. For the case of handover processes for the Odin framework and the traditional enterprise framework, even though they are able to achieve high throughput, still they suffer from packet loss which happens during the handover process (packet delivery ratio ranges between 93% and 95%).

Figure 5 presents the per packet average TCP delay experienced by the STAs. The average delay increases as the number of STAs increases for the case of the proposed handover process and the handover process for the Odin framework, but the average delay for the proposed handover process is always less than that of Odin framework. This shows that though both handover processes reduce the handover delay, the handover process for the Odin framework does not achieve seamless handover which results in TCP retransmitting all the dropped packets. This, in turn, increases the total number of packets in the system thus resulting in higher average delay (more buffering). The average delay experienced in the case of the proposed handover process is also less than that of the handover process for traditional enterprise WLAN for all the cases. This shows

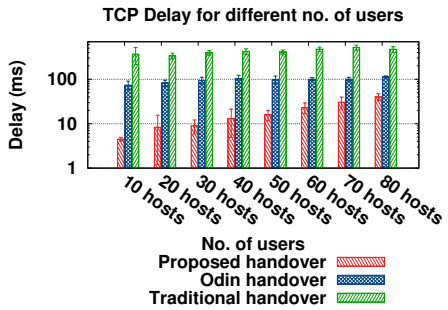


Fig. 5. Average TCP Delay for varying number of users.

that the handover delay is reduced by the proposed handover process and as a result, the TCP applications experience less delay.

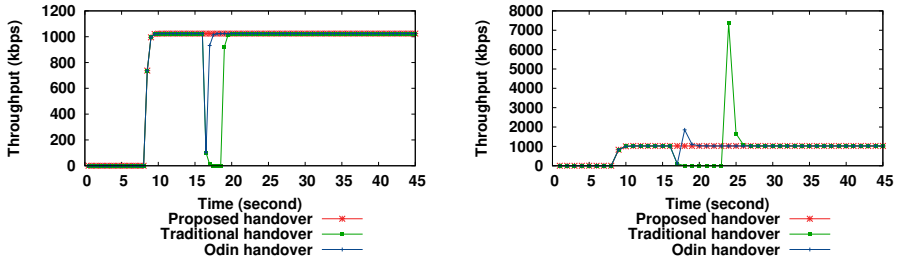
4.3 Instantaneous Throughput Results

The parameters used for running the next set of simulations are summarized in table 2.

Table 2. Simulation parameters for checking instantaneous throughput.

Parameter	Value
No. of APs	4
No. of mobile stations	10
Speed of mobile stations	10 m/s
Traffic at each mobile station	1 UDP application of 1.024 Mbps 1 TCP application of 1.024 Mbps

Figures 6a and 6b present the instantaneous throughput of UDP and TCP applications respectively over the complete simulation time experienced by a randomly selected mobile station for all the three handover processes. The case of 4 APs and 10 STAs is chosen as in this case there will be little to no interference and we can clearly understand the effect of handover on throughput. For both UDP and TCP applications, there are interruptions during the handover processes for traditional enterprise WLAN and Odin framework but the communication in Odin framework resumes quickly compared to that of the traditional WLAN. This shows that even though Odin framework is able to reduce the handover delay, there still will be an interruption in the communication during the handover as the packets are not re-routed to the new AP after the completion of the handover process. The throughput of the TCP application spikes up just after the handover in the traditional enterprise WLAN and Odin framework. This is because, as soon as the connection is re-established all the packets, which were dropped during the handover, are retransmitted by the TCP layer at the remote host. However, for the UDP application, all the packets, sent during the handover, get dropped. Only



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

Fig. 6. UDP and TCP Throughput experienced by a random mobile station.

when the connection is re-established after the handover, the UDP packets get delivered to the STA.

For the proposed handover process, the communication continues uninterrupted and the throughput, for both types of applications, remains high even during the handover. This shows that the proposed handover process is able to reduce the handover delay and also provide seamless handover to the roaming STA.

The simulation-based performance evaluation results show that the handover process is able to seamlessly handover the STAs with negligible handover delay. We can also see that the handover process is capable of serving a high number of STAs (10 to 80) with more network load and is still able to achieve high throughput with low packet loss. Thus, it can be said that the proposed handover process is well suited for enterprise WLAN environment.

5 Conclusions

Enterprise WLAN is used by mobile devices to connect to the Internet in various environments. Mobility management becomes an important aspect in an enterprise WLAN for providing seamless connectivity to the various mobile applications running on the mobile devices. In this paper, we have proposed a NAT based seamless mobility management mechanism for an SDN based enterprise WLAN framework. The performance study shows that the proposed mobility management mechanism is able to provide seamless, uninterrupted connectivity during handovers. Thus, high throughput and packet delivery ratio is achieved with the delay being within reasonable limits. As part of future work, this work can be extended to studies in a large-scale enterprise or campus environment to fully understand the potential benefits.

References

1. IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS) Transition. IEEE Std

802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008) pp. 1–126 (July 2008). https://doi.org/10.1109/IEEESTD.2008.4573292
2. 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 of IEEE Std 802.11-2012) pp. 1–3534 (Dec 2016). https://doi.org/10.1109/IEEESTD.2016.7786995
3. INET Framework. `https://inet.omnetpp.org/` (Feb 2019)
4. Chen, W., Yen, L., Chuo, C., Heish, T., Tseng, C.: SDN-enabled session continuity for wireless networks. In: Proc. of IEEE ICC. pp. 1–6. IEEE (2017)
5. Croitoru, A., Niculescu, D., Raiciu, C.: Towards Wifi Mobility without Fast Handover. In: Proc. of NSDI. pp. 219–234 (2015)
6. Dai, Y., Li, F., Li, H., Wu, Q.: A core-stateless ip mobility management scheme based on openflow protocol. In: Proc. of IEEE IWCMC. pp. 1117–1122. IEEE (2016)
7. Dely, P., Kassler, A., Chow, L., Bambos, N., Bayer, N., Einsiedler, H., Peylo, C., Mellado, D., Sanchez, M.: A software-defined networking approach for handover management with real-time video in wlans. Journal of Modern Transportation **21**(1), 58–65 (2013)
8. 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: Proc. of IEEE Globecom Workshops. pp. 186–191. IEEE (2012)
9. Feirer, S., Sauter, T.: Seamless handover in industrial WLAN using IEEE 802.11k. In: Proc. of IEEE ISIE. pp. 1234–1239. IEEE (2017)
10. Guimarães, C., Corujo, D., Aguiar, R.L., Silva, F., Frosi, P.: Empowering software defined wireless networks through media independent handover management. In: Proc. of IEEE Globecom. pp. 2204–2209. IEEE (2013)
11. Klein, D., Jarschel, M.: An OpenFlow extension for the OMNeT++ INET framework. In: Proc. of the International Conference on Simulation Tools and Techniques. pp. 322–329 (2013)
12. Kobayashi, M., Seetharaman, S., Parulkar, G., Appenzeller, G., Little, J., van Reijendam, J., Weissmann, P., McKeown, N.: Maturing of OpenFlow and Software-defined Networking through deployments. Elsevier Computer Networks **61**, 151–175 (2014)
13. Masoudi, R., Ghaffari, A.: Software defined networks: A survey. Elsevier Journal of Network and Computer Applications **67**, 1–25 (2016)
14. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review **38**(2), 69–74 (2008)
15. Nunes, B., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. IEEE Communications Surveys Tutorials **16**(3), 1617–1634 (2014)
16. Open Networking Foundation: Openflow specifications (Dec 2014)
17. Sen, A., Sivalingam, K.M.: An SDN framework for seamless mobility in enterprise WLANs. In: Proc. of IEEE PIMRC. pp. 1985–1990 (2015)
18. Suresh, L., Schulz-Zander, J., Merz, R., Feldmann, A., Vazao, T.: Towards programmable enterprise WLANS with Odin. In: Proc. of ACM HotSDN workshop. pp. 115–120 (2012)
19. Varga, A., Hornig, R.: An overview of the OMNeT++ simulation environment. In: Proc. of International conference on Simulation tools and techniques for communications, networks and systems & workshops. p. 60 (2008)
20. Zeljković, E., Marquez-Barja, J.M., Kassler, A., Riggio, R., Latré, S.: Proactive Access Point Driven Handovers in IEEE 802.11 Networks. In: Proc. of CNSM. pp. 261–267. IEEE (2018)
21. Zhao, D., Zhu, M., Xu, M.: SDWLAN: A Flexible Architecture of Enterprise WLAN for client-unaware fast AP handoff. In: Proc. of ICCCNT. pp. 1–6 (2014)