Journal of Modern Transportation (2013) 21:58–65

www.springer.com/40534

A software-defined networking approach for handover management with real-time video in WLANs

Peter Dely · Andreas Kassler · Lawrence Chow · Nicholas Bambos · Nico Bayer · Hans Einsiedler · Christoph Peylo · Daniel Mellado · Miguel Sanchez

Received: 15 November 2012 / Revised: 18 February 2013 / Accepted: 17 March 2013

Abstract To achieve high performance and reliability in video streaming over wireless local area networks (WLANs), one must jointly consider both optimized association to access points (APs) and handover management based on dynamic scanning of alternate APs. In this article, we propose a new architecture within the software-defined networking (SDN) framework, which allows stations to be connected to several APs simultaneously and to switch fast between them. We evaluate our system in a real-time testbed and demonstrate that our SDN-based handover mechanism significantly reduces the number and duration of video freeze events and allows for smaller playout buffers.

Keywords Openflow · Handover · Video streaming

1 Introduction

The upcoming wireless local area network (WLAN) standards IEEE 802.11ac and 802.11ad will offer PHY layer rates of several Gbit/s using high-order coding and modulation schemes, wide channels, and high carrier frequencies. To achieve those high rates, short- and high-quality radio links and a dense network deployment are necessary. As a consequence, even with mobility at moderate speeds, handovers between access points (APs) need to be executed frequently to insure that a mobile station is always using a close-by AP with a high signal quality. Thus, mobility management will play a more pronounced role in future WLANs.

Several IEEE standards for mobility management have been published. For example, IEEE 802.21 [1] describes signaling messages exchanged to trigger handovers between WLANs and other networks. IEEE 802.11r [2] specifies how the authentication process and the encryption key negotiation during a handover can be accelerated. While those standards offer some messaging primitives to control handovers, they are not sufficient to enable seamless mobility in WLANs. For example, those standards do not specify when and how to detect new APs, when to schedule a handover, and how to control the wired distribution system to route the traffic to the right AP.

Those open issues, however, are a key to enable seamless mobility in WLANs. In particular, video streaming and conferencing applications require fast handovers, since such applications use small playout buffers and a too long a handover duration would result in a frozen video. To this end, we propose to allow a station to be associated to several APs simultaneously and to use the concept of software-defined networks (SDNs) [3] and the Openflow protocol [4] to steer the handover. SDNs allow us to integrate application characteristics into the handover decision in an elegant way. As the main contribution of this article, we describe an SDN-based system architecture for fast handovers in WLANs and evaluate this architecture in a WLAN testbed using video streaming applications.

P. Dely (✉) · A. Kassler
Computer Science Department, Karlstad University, 65188 Karlstad, Sweden
e-mail: peter.dely@kau.se

L. Chow · N. Bambos
Department of Management Science and Engineering, Stanford University, Stanford, CA 94305, USA

N. Bayer · H. Einsiedler · C. Peylo
Telekom Innovation Laboratories, Deutsche Telekom, Berlin, Germany

D. Mellado · M. Sanchez
The Charles III University of Madrid, Madrid, Spain

Optimization of WLAN handovers has been examined in several studies before. For example, Refs. [5] and [6] investigated how to optimize the scanning procedure for new APs. Those methods are orthogonal to our proposal which targets the actual handover and can be used to improve the scanning duration. Nah et al. [7] proposed a method for scheduling the scanning procedure in WLANs, while considering the buffer level of the video player. In comparison to this study, their evaluation is based on network simulation, and the re-configuration of the wired backhaul network is not considered.

The use of SDN/OpenFlow for improving streaming video has been explored [8]. However, in contrast to our study, Ref. [8] stipulates that multiple interfaces are required at the station and explores the concurrent use of different radio technologies (WLAN and WiMAX). Scanning for new APs is not explicitly addressed in this study. CloudMAC [9] is a recent proposal to distribute the MAC processing in WLANs using SDNs. CloudMAC also allows implementing fast handovers between APs, but is mainly targeting network initiated handovers, while this article uses client initiated handovers.

References [10] and [11] formulate and solve optimization problems to compute the best moment for performing a handover. While Ref. [10] uses an application independent formulation, which aims to maximize the long-term throughput, Ref. [11] explicitly considers the characteristics of streaming video and the playout buffer.

The remainder of the article is organized as follows: In Sect. 2, we describe in detail the proposed system architecture. Section 3 outlines the implementation of the architecture. In Sect. 4, we present evaluation results. We wrap-up the article in Sect. 5.

2 System architecture

2.1 Overview

Figure 1 provides an overview of the system architecture. The system consists of a streaming server acting as the video source. The streaming server is connected to WLAN APs via an OpenFlow switch. A controller dynamically configures the forwarding table of the OpenFlow switch to direct traffic from the streaming server to the station via the right AP.

2.2 Mobile station

As Fig. 2 shows, the mobile station is equipped with one IEEE 802.11 WLAN card and optionally with one dedicated card for scanning for new APs. On top of the physical card, several virtual WLAN cards are created. Each virtual WLAN card has a unique MAC address and can be associated to one AP. As one station has multiple virtual

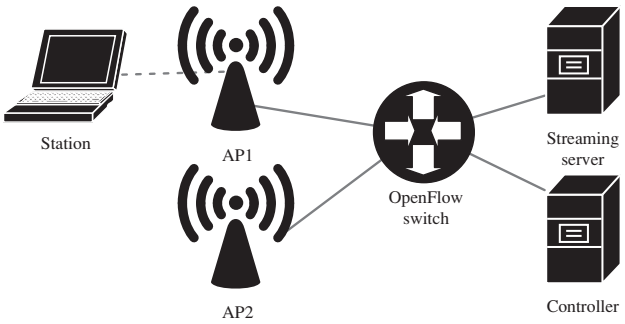

Fig. 1 Overall system architecture

WLAN cards, it can be connected to several APs simultaneously. It is not required that all APs are operated on the same channel.

The virtual WLAN cards are connected to a software switch, the OpenFlow switching element. In addition, the software switch contains a super virtual WLAN device and a transmission buffer. The super virtual WLAN is the interface to the OS networking stack and has a device-wide IP and MAC address. The transmission buffer can be set into a blocked state, in which all data packets (not control packets) from higher layers are stored in the buffer, but not forwarded to the OpenFlow switching element. If the transmission buffer is in the unblocked state, then it simply passes packets down to the switching element. Blocking traffic at the transmission buffer insures that no packets are lost when the station is performing a handover. Packets that are lost during the wireless transmission (e.g., due to bit errors) are retransmitted using normal operations of the IEEE 802.11 MAC protocol. Retransmissions are always targeted to the same AP.

The switching element contains rules that specify (1) which virtual WLAN card outgoing packets should be forwarded to and (2) what source MAC address should be used for outgoing packets. Those rules are configured from the switching controller application using OpenFlow. As each virtual WLAN card is associated to one AP, choosing a virtual WLAN card for transmission results in a transmission to a specific AP. Rewriting of the source MAC address is required, since MAC frames generated by the operating system carry the MAC address of the super virtual WLAN device as source. To be transparent to the underlying network, the MAC address of the respective virtual WLAN card is used instead.

The station contains the handover manager which consists of a scheduler and a scanning module. The scheduler decides when to transmit or receive frames via which AP and when to scan for new APs. In other words, the scheduler decides when to perform a handover. In addition, the handover manager generates the signaling messages to reconfigure the routing in the backhaul

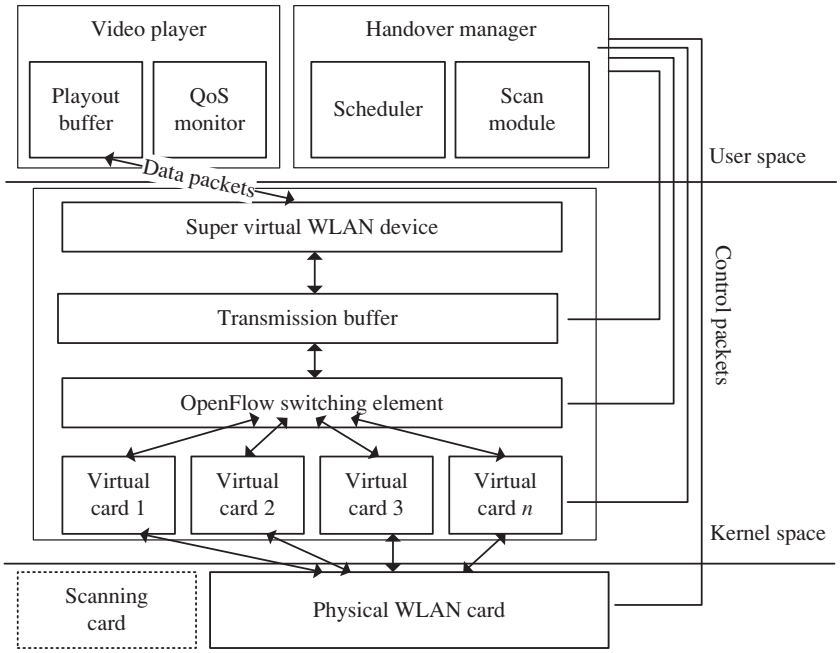

Fig. 2 Station architecture

network for downstream packets (see Sect. 2.5). The decision when to schedule a handover or a scan for new APs can use information from the Quality of Service (QoS) monitor and the playout buffer of the video player application. For example, a handover could be scheduled when the video buffer level is getting low or the QoS level is degrading.

2.3 Pre-authentication and pre-association

A station can associate to an AP outside its coverage area using out-of-band signaling. Using an existing wireless connection and the wired backhaul, it can exchange association and authentication messages with the control server, which relays them to the AP. This mechanism allows a station to associate to all APs of a limited geographic area, before the station arrives at this area. As we will discuss in Sect. 2.5, this allows handovers to be accelerated. Note, that our approach of pre-authentication and pre-association is fundamentally different from IEEE 802.11r, as we allow exchanging messages via an AP which is not in reach using the connection of a close-by AP. This can be performed before the handover. Therefore, the number of authentication message exchanged is not a performance limiting factor in our scheme and the authentication does not increase the duration of the actual handover.

2.4 Scanning for new APs

Before a station can initiate a handover to a new AP, it needs to detect it. To detect new APs, the scan module in the handover manager performs a standard IEEE 802.11 active scanning procedure, i.e., it broadcasts probe request frames. APs on the respective channel answer with probe response messages. The scan procedure can either be performed on the regular WLAN card, which is also used for data communication, or on a dedicated scanning card. If the regular WLAN card is used for scanning, then a trade-off between the scan frequency and the achievable throughput and QoS emerges. A scan for new APs needs to be scheduled often enough to detect new APs fast enough. However, too frequent scanning would result in a reduction of throughput and QoS, as during scanning no data can be transferred.

A dedicated scanning card would avoid this trade-off, but increase hardware costs. Future mobile devices, such as smart phones and tablets, are presumably equipped with several flexible radios. With such devices, a dedicated scanning card would be a viable option. In Sect. 4, we will evaluate the trade-off between scanning frequency and AP detection speed as well as the benefit of a dedicated scanning card.

2.5 Handover process

Based on the scanning results, the scheduler decides when to initiate a handover to which AP. In the current version, the scheduler triggers a handover, if the Received Signal Strength Indicator (RSSI) of any AP is larger than the RSSI of the current AP plus a hysteresis margin. Figure 3 shows a sequence diagram of the messages exchanged during a handover. The station puts its transmission buffer into the blocked state. Hereafter, the station transmits a “Handover Initiate” message to its currently used AP (AP1), which

forwards the message to the controller. The controller (1) answers to the station with an ACK message, (2) sends a pause message to the switch and (3) using the OpenFlow protocol updates the forwarding table of the switch to send future downstream data packets via AP2 and to forward upstream traffic coming from the station via AP2 toward the streaming server.

As soon as the station has received the ACK message, it tunes its WLAN card to the channel of the new AP. In case no ACK is received within a timeout period, the station retries up to five times. If no retry is successful, for example, because AP1 is already out of reach, then the station nevertheless attempts to change the channel and sends a "Handover Initiate" message via AP2. The OpenFlow switch, just like the station, contains a per station transmission buffer, which is set into the blocking state upon the reception of the pause transmission command. Once the station has tuned its card to the new channel, it sends a "Handover Complete" message to the controller via AP2 and unblocks its transmission buffer. The controller then resumes the downlink traffic by unblocking its transmission buffer.

The handover process does not require exchanging any association or authentication messages, as pre-authentication and pre-association insures that the station is already associated to the new AP before the actual handover is started.

3 Implementation

We have implemented the proposed system in Linux. The station uses the virtual WLAN card function provided by ath5k and ath9k drivers. By default, Linux does not allow virtual interfaces to be used on different channels, which we changed accordingly. Furthermore, we add an interface to the WLAN driver, which allows initiating a fast channel switch. The pre-authentication/association scheme is implemented by extending the mac80211-subsystem and hostapd. For example, a new control command was added to hostapd, which allows external applications to create an association for a given station MAC address and capability list (PHY rates, 40 MHz channels etc.). Similarly, the mac80211-subsystem was modified to create an association without exchanging IEEE 802.11 authentication/association frames over the air.

The OpenFlow switching element is based on OpenVSwitch [12]. The "Super virtual WLAN card" is a bridge device connected to OpenVSwitch. All user-space applications (handover manager, video player, control server) were written in Python. The scanning module uses a raw-socket to inject probe messages into the WLAN card driver. The video player application is based on the Gstreamer framework and has a socket interface, which allows the handover manager to query its current buffer level and QoS statistics such as frame drop rate. We use VLC as streaming server and HTTP streaming.

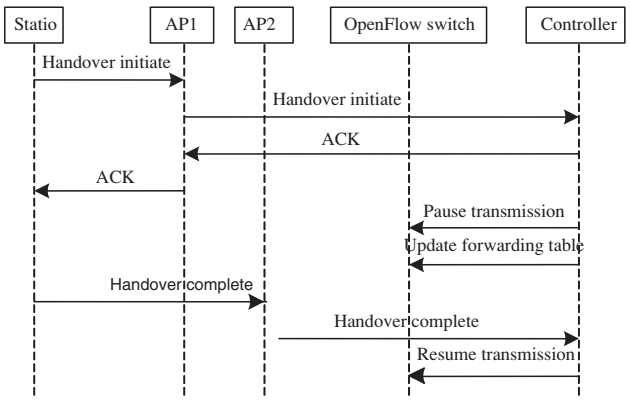

Fig. 3 Sequence of messages exchanged during a handover from AP1 to AP2

4 Evaluation

We have evaluated the proposed system in a WLAN testbed deployed in an office building in downtown Berlin (see Fig. 4).

The testbed consists of 8 embedded systems with a Dual Core Atom D525 1.80 GHz CPU, which act as APs. Some of the APs are deployed in offices and meeting rooms (e.g., AP1), while others are located on corridors (e.g., AP5). The APs are positioned so that a mobile node in the corridor at any time is within range of at least one AP.

Each AP is equipped with two wireless cards, of which only one is used in our experiments. The cards are based on the Atheros AR5418 IEEE 802.11abgn chipset. To reduce interference from other WLANs deployed in the area, the network is operated in the less occupied 5 GHz band. The APs are connected via Ethernet and a Generic Routing Encapsulation (GRE) tunnel to a virtual machine host, in which the OpenFlow switch, the streaming server and the control server are hosted. The OpenFlow switch uses OpenVSwitch.

4.1 Micro-benchmarks

4.1.1 AP scanning duration

The mobile station needs to scan for new APs occasionally to detect new target APs for a handover. While scanning, the station needs to process a list of channels and send out

Fig. 4 Map of the testbed

probe request messages on each channel. After sending a probe request message, the station needs to remain on the channel for a while to wait for probe response messages. We remain on the channel for 5 ms, which turned out to be a good compromise between the probability of missing a probe response and the duration of the scan. In our experiments, switching from one channel to another channel requires on average 1.2 ms if we empty the transmission WLAN queue before a switch and recalibrate the card after the switch ("Normal channel switching"). If we just tune the synthesizer to the new frequency ("Fast channel switching"), then the channel switch takes on average 1 ms. Figure 5 plots the duration for scanning 2–8 channels. The scan duration increases linearly with the number of channels to scan. Each extra channel adds about 8 ms (1.2 ms for switching, 5 ms for waiting and listening and 1.8 ms for processing the frames) to the total scan duration. To reduce the scan duration in a real deployment, the station could only scan channels which are used by the network operator (often channels 1, 6 and 11). The total scan durations for the normal and fast channel switch mode are almost identical as the majority of the time is spent on listening for probes and other processing activities, and the actual channel switch does not play a major role.

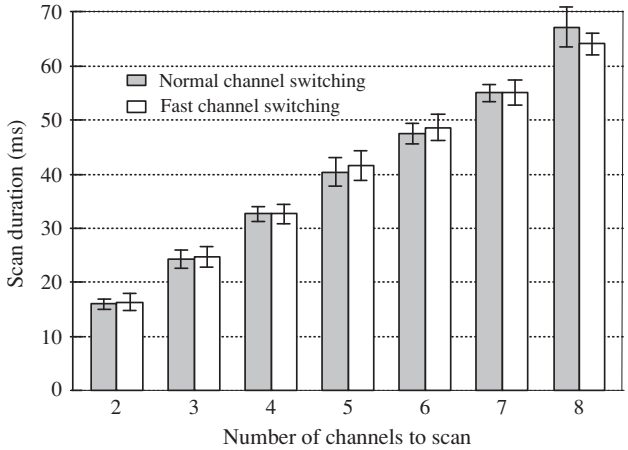

Fig. 5 Duration of a scan for new APs

4.1.2 *Handover duration*

As discussed in Sect. 2.5, a handover consists of a series of messages to be exchanged. We call the time between the transmission of the handover initiate message and the reception of the ACK message the ACK duration. The total handover duration spans the period from the transmission of the handover initiate to the transmission of the handover complete message. We measured the duration taken by our optimized handover for 100 handovers and plot the Empirical Cumulative Distribution Function in Fig. 6. The average ACK and handover durations are 9.4 and 34.7 ms, respectively. For a small fraction, the handover duration exceeds 50 ms. In those cases, if one of the control messages (e.g., the handover initiate) is lost, then it needs to be retransmitted after a timeout. In contrast, Ref. [13] reports handover times of 6–9 s for IEEE 802.11i secured WLANs.

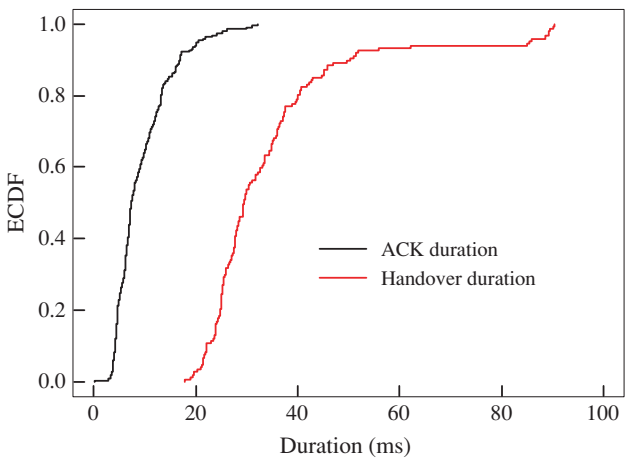

Fig. 6 Empirical Cumulative Distribution Function of handover and ACK duration

4.1.3 Frequency of handovers

We walked along the corridors marked with the red line in Fig. 4 ten times and recorded the signal strength of the APs every 0.5 s. The decision to trigger a handover is based on the RSSI of the APs. If any AP has a higher signal strength than the current AP plus a hysteresis margin, then a handover is performed. As Fig. 7 shows, by increasing the hysteresis margin, the average time between two handovers increases. If the hysteresis margin is 0, then a handover is triggered on average every 3.6 s.

Increasing the hysteresis margin reduces the number of handovers, but comes at price: the station is using optimal APs less frequently. With a hysteresis margin of 20 dB, the station uses the best AP only 60 % of the time, while it stays connected to suboptimal APs 40 % of the time.

It depends on the application characteristics, whether a handover is beneficial or not. For example, low bandwidth application might not benefit from the higher throughput after handover to a better AP, but might suffer from the disruption the handover can cause. This shows that an intelligent handover scheduler is necessary and simple hysteresis schemes might not be sufficient.

4.2 Video streaming

Next, we compare how well our proposed system supports streaming video. Streaming video is an application that is gaining popularity rapidly and poses high requirements on the performance of the handovers. For our experiments, we streamed a pre-recorded MPEG-2 video using HTTP-streaming to the client laptop. A person is walking with the laptop five times from point A to B and back (see Fig. 4), while streaming the video. One walk from point A to B takes approximately 90 s. The video player has a playout buffer of 400 ms and logs the buffer fill level every 20 ms to a file. If within 400 ms, then no new video frames are received, and the video freezes until the buffer is filled up again. Such video freezes impair the perceived quality and thus should be avoided.

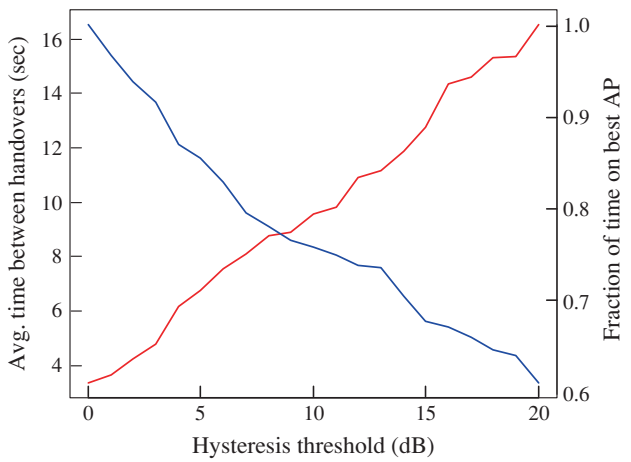

Fig. 7 Average time between two handovers and the loss of signal quality due to a higher hysteresis

In the following, we compare (1) Linux with wpa_supplicant, (2) Linux with wpa_supplicant and optimized scanning, (3) our system with a dedicated scanning card, and (4) our system without a dedicated scanning card. The optimized scanning procedure in (2) only scans on the four channels which are actually used by our network, while the normal scanning procedure of (1) scans all channels. In our system, without a dedicated scanning card, we scan for a new AP every 2 s, while with a dedicated scanning card, we initiate a new scan every 500 ms. For all experiments, the hysteresis threshold was 0 dB.

4.2.1 Smoothness of video playout

Table 1 shows that with the standard Linux system 28.3 % and 12.48 % of the time the video is in freeze mode. With our proposed system, the video is frozen only 0.73 % and 3.25 %, respectively. Interestingly, with the dedicated scan card, the freeze time is slightly higher than without. This can be mainly attributed to two factors: first, with a dedicated scanning card, updates on the APs' quality are obtained more frequently, and hence handovers happen more often. Such behavior could be avoided with a more intelligent handover scheduler. Second, in our experiments with the dedicated scanning card, two relatively long freeze events occurred as result of failed handovers, which have large impact on the average video freeze time. This phenomenon can also be observed from Fig. 8, which shows the distribution of the freeze event durations. Figure 8 further shows that the standard Linux implementation leads to many long freeze events, which have severe impact on the user perception. In particular, if the channel scan is not optimized, freeze events last 10 s or more. With our system, however, the number of freeze events is small and the duration is usually short and thus not very disturbing to the viewer.

4.2.2 Necessity of a dedicated scanning card

The total number of freeze events is lower if a dedicated scanning card is in operation, while the total length of freeze events is higher. With our optimized scanning procedure, the benefits in terms of the total number of freeze events are marginal and also the subjective quality differences experienced during the measurements were low. On the one hand, a dedicated scanning card might decrease the system complexity, as the question when to schedule a new scan is easier to be answered then. On the other hand, a

Table 1 Video freeze events

Mode	Freeze time fraction (%)	Total number of freeze events	Maximum freeze time (s)	Mean freeze time (s)
Proposal/no dedicated scan card	0.73	15	2.24	0.46
Proposal/dedicated scan card	3.25	11	6.24	2.24
Linux/optimized scanning	12.48	55	8.43	2.01
Linux/standard scanning	28.30	79	32.06	2.75

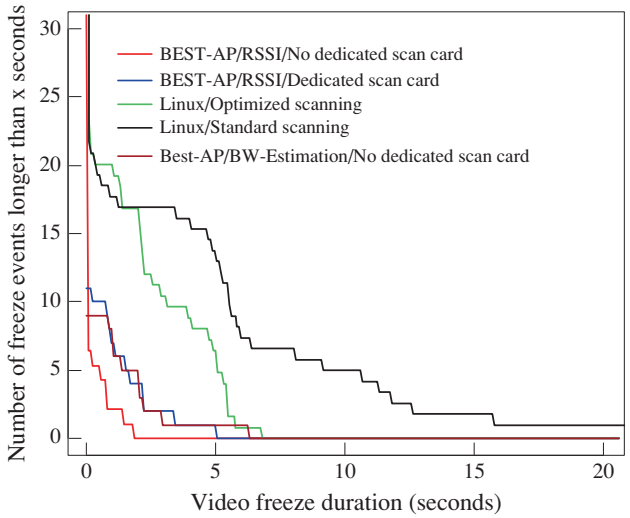

Fig. 8 Distribution of freeze event durations

dedicated scanning card increases hardware cost and energy consumption. Considering all those factors, a dedicated scanning card is only useful, if it allows being integrated into the system with low effort, for example, through software defined radios.

4.2.3 Optimal playout buffer size

The playout buffer adds delay, which should be kept small, in particular for real-time applications such as video conferencing. Decreasing the playout buffer size would increase the number of freeze events, while increasing the buffer size would offer more protection against freeze events. Figure 8 shows, that to offer protection against all freeze events, the buffer size should be approximately 3–5 s large. Such large buffers are suitable for on-demand video, but the large delay would render real-time applications unusable. However, even with the 400 ms buffer used in our experiments, the number of noticeable freeze events is relatively low and the perceived quality is good.

4.2.4 Maximum possible station velocity

The proposed architecture is targeted to WLAN environments. As the WLAN PHY is not designed for fast moving users (e.g., in cars), high velocities are not possible. For example, Ref. [14] showed that even at moderate speeds of 50–60 km/h the packet error rate of an IEEE 802.11a channel exceeds 10 %. Besides the PHY, also communication distance of typical WLANs is a limiting factor. Considering a straight movement with 60 km/h and a communication range of 30 m, a handover would have to be performed at least every 3.5 s. Thus, the proposed system is only suitable for pedestrian-type speeds.

5 Conclusions

In this article, we have presented and evaluated an SDN-based architecture for optimizing handovers in WLANs. We have investigated as regards when and how to detect new APs using an optimized scanning procedure, when to schedule a handover, and how to use OpenFlow to control the wired distribution system to route the traffic to the right AP. The evaluation has shown that our architecture significantly improves the quality of streaming video over WLANs. As future study, we plan to investigate how to couple the handover decision more tightly with the video player application. This would allow for more intelligent scheduling of scanning or handovers.

References

1. IEEE Std 802.21-2008 (2009) IEEE standard for local and metropolitan area networks—part 21: media independent handover
2. IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008) (2008)
3. Open Networking Foundation (2012) Software-defined networking: the new norm for networks. http://www.opennetworking.org
4. The OpenFlow Consortium (2012) Openflow switch specification 1.1
5. Mishra A, Shin M, Arbaush WA (2004) Context caching using neighbor graphs for fast handoffs in a wireless network. In: Proceedings of INFOCOM 2004, Hongkong
6. Wu HT, Tan K, Zhang YG et al (2007) Proactive scan: fast handoff with smart triggers for 802.11 wireless WLAN. In: Proceedings of INFOCOM 2007, Alaska
7. Nah JW, Chun SM, Wang S et al (2009) Adaptive handover method with application-awareness for multimedia streaming service in wireless Lan. In: Proceedings of ICOIN 2009, Chiang Mai
8. Yap KK, Huang TY, Kobayashi M et al (2009) Lossless handover with n-casting between wifi-wimax on openroads. In: Proceedings of ACM MOBICOM 2009, Beijing
9. Vestin J, Dely P, Kassler A et al (2012) Cloudmac: towards software defined WLANs. Mobile Comput Commun Rev 16(4): 42–45

10. Dely P, Kassler A, Bayer N et al (2012) Optimization of WLAN associations considering handover costs. EURASIP J Wirel Commun Netw (1):1–14
11. Chow L, Collins B, Bambos N et al (2012) Playout-buffer aware hand-off control for wireless video streaming. In: Proceedings of GLOBECOM 2012, Anaheim
12. Pfaff B, Pettit J, Koponen T et al (2009) Extending networking into the virtualization layer. In: Proceedings of ACM HotNets, New York
13. Martinovic I, Zdarsky FA, Bachorek A et al (2007) Measurement and analysis of handover latencies in IEEE 802.11i secured networks. In: Proceedings of EW2007, Paris
14. Sibecas S, Corral C, Emami S et al (2002) On the suitability of 802.11a/RA for high-mobility DSRC. In: Proceedings of IEEE vehicular technology conference, Birmingham