Telecommun Syst
DOI 10.1007/s11235-016-0264-1

SDN-based wireless user management system

Eduardo Luengo[1] · **Shahram Shah Heydari**[1] · **Khalil El-Khatib**[1]

Abstract The degradation of end-to-end quality of service (QoS) on mobile users is becoming a common issue for IEEE 802.11 infrastructure-based networks in crowded areas. This research tackles the issue by employing an SDN framework on an integrated wireless/wired environment. Thereby, we present the development and implementation of a system that performs user management by analyzing the network load from the OpenFlow statistics, as well as the wireless information collected from the associated users. In order to analyse the behaviour of the proposed user migration algorithm, we evaluate the system under scenarios with different traffic load and user session duration. From the experiments, we observed that in several cases wireless users get a considerable QoS improvement at the application layer (up to 30% improvement in throughput and up to 20% in delay in our simulations) once the system is activated. Based on the results, we present an analysis on how and when user migration in multi-access point IEEE 802.11 networks can be most effective.

Keywords Wireless SDN · Wireless user management · OpenFlow · Load balancing

1 Introduction

Wireless technologies have revolutionized human life by allowing suitable, fast and reliable ways of communication anywhere. In particular, the ever-increasing presence of public/private Wireless Local Area Networks (WLAN) allows high-speed connections to transient users with portable devices such as laptops, phones and tablets. These WLAN networks are commonly implemented with the IEEE 802.11 standard which was originally designed to provide convenient high-speed wireless communications to a reduced number of computers by utilizing unlicensed radio-frequency spectrum [1]. Since then, the standard have been revised in order to enhance technical aspects related to modulation techniques, frequency range of operation, and spatial diversity capabilities. However, the predominant adoption of the standard, which was originally developed for confined environments, and the high demands of traffic generated by wireless users raise concerns in terms of performance and scalability. In this regards, common issues encountered in crowded IEEE 802.11 networks are related to user management, user mobility, radio frequency interference from IEEE 802.11 and non-IEEE 802.11 sources, security and quality of service (QoS) [2].

Public crowded areas usually have several Access Points (AP) with spare capacity for wireless users to connect to. These APs may belong to the same administrator or they may be part of a cooperative framework of different administrators. However, regardless of the network capacity distributed among the available APs in the area, wireless users experience poor connection services. The reason is that a single AP is usually supporting large volume of traffic from several wireless users while other APs remain idle. Hence, congestion occurs when the overloaded AP can no longer accommodate all the traffic demands from the associated users. From a user's perspective, this leads to the degradation of the QoS, affecting the end-to-end throughput, delay, jitter and packet-loss of services and applications (i.e., video/audio streaming) [3].

The fact that many wireless users connects to the same AP is due to the association mechanism of the IEEE 802.11.

✉ Shahram Shah Heydari
Shahram.heydari@uoit.ca

[1] University of Ontario Institute of Technology, 2000 Simcoe Street North, Oshawa, ON, Canada

Published online: 06 January 2017

The standard allows wireless user devices to decide which AP to associate with, based on partial information such as the strongest signal-to-noise ratio and theoretical transmit rate. An example is a scenario where there is an empty room with overlapping wireless coverage provided by distributed APs. In this case, as IEEE 802.11 users enter the room, it is likely that users will associate to the closest AP to the entrance door (due to its highest signal strength), despite the available capacity on other APs. Hence, the independent AP selection by the user without a complete view of the network topology and status, leads to the allocation of large amount of traffic in some APs, while others remain idle. Thus, the mere presence of under-loaded APs in the vicinity does not solve the problem.

In order to tackle the issue, a user management system is required. User management solutions distribute the user traffic load across nearby APs so that network congestion can be avoided. A number of commercial solutions as well as methods based on traditional wireless architecture have been proposed for this purpose. However, these approaches typically require either proprietary protocols and implementations, or the use of management agents and protocols (i.e., SNMP) that makes the architecture more complex to operate.

More recently, the applicability of Software-Defined Networking (SDN) in IEEE 802.11 networks has been gaining popularity. One reason for this approach is to discover simple, flexible and standardized solutions to deal with the traditional IEEE 802.11 management issues that we mentioned before. In an SDN architecture, the data and control planes are separated and the control plane is managed through an SDN controller. The SDN controller allows communication with high-level components (e.g. network management applications) through *northbound* interface, and with lower-level components (e.g. switches) through *southbound* interface. Usually, an SDN controller has knowledge of the entire topology and network status by retrieving traffic measurements from the network devices. This information serves as an input to make dynamic traffic handling decisions and to install forwarding rules in network devices. The interaction between the SDN controller and the network devices (SDN switches) is carried out by means of a standardized southbound SDN protocol such as OpenFlow. Among its capabilities, the OpenFlow protocol allows the collection of statistics and the configuration of quality of service (QoS) and traffic shaping [4]. Moreover, the northbound interface of an SDN controller is usually developed as Representational State Transfer (REST) APIs. In this way, most of the network complexity is hidden, allowing the upper services/applications to get information from the network and manage certain aspects of the traffic in a flexible manner.

Considering the fact that an SDN architecture maintains a general view of the network topology, IEEE 802.11 network performance issues can now be tackled from a different perspective than with traditional wireless solutions. This paper presents the design and analysis of a dynamic wireless user management system for infrastructure-based IEEE 802.11 networks. The novel approach uses the SDN architecture in order to mitigate load-balancing issues in IEEE 802.11 networks, without introducing modifications to the southbound protocol (OpenFlow). The proposed system collects statistics from the SDN network as well as wireless information from associated users. Based on analysis of the collected data, the system detects traffic load asymmetry on APs, and consequently makes user migration decisions to distribute the load among nearby APs. The aim of this solution is to improve the end-to-end QoS of wireless users by maximizing the throughput and, at the same time, reducing the delay, jitter, packet-loss, and the disconnection time elapsed during migrations.

The main contribution of this paper includes the following:

- System architecture: A description of the components involved in the system, definition of their roles and interactions between them.
- Algorithm: The design and implementation of a wireless user migration algorithm to mitigate the load asymmetry detected by the system.
- Emulation: A detailed description regarding technical aspects of the real environment used to test the system
- Performance evaluation: A study of the improvements once the user management approach is applied. In addition, this paper analyses the impact of the user migration algorithm on the QoS.

The remainder of this paper is organized as follows. Section 2 describes the prior related works in the field. Section 3 explains the design and requirements of the system. Section 4 presents the proposed algorithm. Section 5 presents the emulation environment and the results obtained from the experiments. Finally, Sect. 6 concludes the paper and discusses future works.

2 Related works

The prior work in the field of wireless user management has primarily focused on load balancing among APs through user migration. Throughout the years, state-of-the-art solutions have been suggested to solve the load-balancing issue in IEEE 802.11 infrastructure-based networks. Many solutions employed traditional networking architectures while more recently the SDN-based solutions have been considered. All user management approaches include three main components: the definition and monitoring of metrics, an algorithm to determine the asymmetry and the causes, and finally, a mechanism to distribute the load among other nodes

of the network. Moreover, approaches can be preventive or reactive in response to unbalanced conditions in the network. The former usually utilizes access control policies to reject new associations in over-loaded APs. The latter dynamically accommodates the current load of the associated users by means of association mechanisms applied from the AP or centralized wireless controller.

2.1 User-driven approaches

These approaches provided users with detailed wireless information, other than the signal strength, in order to accomplish a smarter connection choice from available APs. For example, Virgil [5] is presented as a user-level application running on the user device, associating to every AP in the vicinity and performing a series of tests to determine bandwidth and round-trip-time values to predefined servers in the Internet. The application then selects the AP with the highest performance. This approach resembles a brute-force method in which network resources are wasted while wireless users find their best AP to finally associate with.

Another example of user-driven approach is the Application Layer Load Distribution protocol (ALDP) [6] in which an SNMP server assists users in the task of collecting wireless metrics. The architecture requires that both APs and user devices run an SNMP agent. The SNMP server provides users with information regarding the degree of congestion on each AP, so that users could re-associate to a different AP if needed. Both residual bandwidth and number of users are sent to the user device, which utilizes that information to compute a normalized value of the remaining capacity on each available AP. In order for ALDP to work, each user is required to know the IP address of the management server in advance (or learn from any entity on the network) and establish a registration. The solution is suitable on small WLAN scenarios. However, the amount of traffic generated by user registrations would not be negligible in crowded environments.

2.2 Network-driven approaches

Alternatively, some user management approaches were positioned inside the network. In [3] a network-driven load-balancing scheme is presented. The architecture consists of overlapped APs configured in different IEEE 802.11 channels and equipped with a Load Balancing Agent (LBA). Basically, APs share load information between them (using a wired network) and perform analysis to determine whether user management actions were required or not. This user migration approach is reactive but also preventive in nature because it rejects upcoming association on over-loaded APs. The results show improvements on user's packet delay and throughput in the long term when load balance is activated. However for migrated users, the data traffic is interrupted for approximately three seconds while the user was re-associating with the new AP. This increased delay is due to the unassisted handoff mechanism utilized.

Sawma et al. [7] presents a network-driven load-balancing algorithm called ALBA (Autonomic Load Balancing Algorithm). The architecture requires a central server to collect measurement from the wireless environment such as the channel utilization, the spatial distribution of users and APs, as well as the QoS profile of each user. The load-balancing algorithm migrates existing users from a given AP to other nearby APs in order to provide enough capacity to a user that was previously experiencing low performance. The algorithm was tested in an overlapping IEEE 802.11b network with wireless users generating UDP VoIP traffic. The results of this work show enhancement in terms of global end-to-end delay (55% improvement) and throughput on the network (13.6% improvement).

A more recent user management approach is introduced in Le et al. [8] based on a centralized server and multiple overlapping WLAN APs. In this architecture, the centralized server performs AP association/disassociation for wireless users based on the results of two algorithms. The first algorithm focuses on a proactive scheme that assigned each user to the lightest AP in the area. The second algorithm works as a reactive user management algorithm to reduce the load on the over-loaded APs by moving their users to APs with minimum loads. The results show improvement on user throughput when the approach was compared with the traditional IEEE 802.11 mechanism. In addition, measurements of Jain's fairness index return values closer to 1. This study did not include results of other network performance metrics such as delay, jitter and packet loss.

2.3 Standardization for 802.11-based approaches

There has been some standardization efforts related to the concept of centralized architecture and user management in IEEE 802.11 networks. The CAPWAP [9] is an IETF protocol for control and provisioning of WLAN APs using a central controller. Commercial solutions based on the CAPWAP protocol have been developed in a proprietary manner, and they often lack interoperability among other CAPWAP vendor solutions [10].

The IEEE 802.11k amendment [1] has been developed as a standard way to provide detailed reports of physical (radio) and data link layer of IEEE 802.11 environments. IEEE 802.11k allows local and remote measurements from wireless users, as well as, measurements on different channel than the one the user was connected to, and it also offers metrics that could be used to mitigate load asymmetry in the network [11]. While IEEE 802.11k has brought new capabilities to acquire detailed information about the wireless

environment, it requires changes on both the AP and user device software to support new frame format and procedures [12,20], and is not always supported in commercial wireless cards. In addition, the traffic overhead generated by user reports might affect the network throughput in scenarios with a large number of wireless users [12].

2.4 SDN in wireless networks

While SDN was originally designed for data centers and enterprise networks, researchers have been analyzing its potential to solve control and management issues in other environments. As such, the research on the integration of SDN in IEEE 802.11 networks has emerged as an alternative approach to overcome the limitation of current network infrastructures.

OpenRoads [13] is an open-source platform developed by Stanford University for integrated wired and wireless (IEEE 802.11 and 802.16) SDN network. OpenRoads focuses on slicing a production network in order to perform independent and concurrent test on the same infrastructure. User-level software is used to equip access points (APs) with Openflow and tunneling capabilities.

Suresh et al. [14] presents Odin, a programmable SDN framework for WLANs. The architecture consists of OpenWRT APs and a Floodlight SDN controller, which collects wireless statistics and parameters from each AP (i.e., signal strength, bit rate, last packet timestamp, etc), by employing a custom-designed southbound protocol. One of the key components introduced by Odin is the Light Virtual Access Point (LVAP) that allows a wireless user to stay associated with a unique virtual BSSID, while moving along the overlapping APs zone. This permits users to switch from one physical AP to another seamlessly (without service disruption). A mobility application on Odin collects the user‘s signal strength values from all LVAPs.to allow for selection of the AP with the highest signal strength for that user. However, the LVAP overhead per associated user in Odin is approximately 80 bytes, which could limit scalability. Furthermore, the verification of the destination address received from associated users (BSSID) might lead to the incorrect generation of ACKs messages, and consequently, these messages might be destined to BSSIDs that were not hosted by the LVAP in the first place. Lastly, though Odin allows seamless handoffs, APs need to be configured on the same IEEE 802.11 channel. Otherwise, Odin would require the support of IEEE 802.11h; an amendment developed for the 5 GHZ frequency spectrum.

Dely et al. [15] develops the CloudMAC architecture for WLANs using SDN technologies. The architecture consists of OpenWRT APs, OpenFlow switches, a POX SDN controller and VMs running virtual instances of IEEE 802.11 access points. The concept behind CloudMAC is to withdraw part of the MAC layer functionality from the physical APs and perform these MAC layer operations on virtual APs (running on standard servers) in the cloud. This approach differs from Odin in that in CloudMAC the physical APs serve just as relaying elements between wireless users and virtual APs. However, physical APs perform some strict-time operations independently, such as ACKs and frame retransmissions. One advantage of CloudMAC is the flexibility to accommodate users on different channel in the same AP.As in Odin, this feature relied on the IEEE 802.11h amendment. No analysis on network latency was provided. It is worth mentioning that CloudMAC provids a smooth integration of wireless and wired SDN elements because, in contrast to Odin architecture, CloudMAC does not require an agent (additional software development) in the APs nor proprietary southbound API to handle the wireless issues.

Ethanol [16] is another SDN architecture for IEEE 802.11 infrastructure-based networks. It is designed to provide the following features: mobility, user management, security, QoS and localization of wireless users, the collection of wireless links statistics, and the virtualization of SDN wired/wireless deployments. Ethanol consists of a SDN controller (POX) and commercial home APs running user-level OpenFlow software (Pantou 1.0). Each AP is equipped with an additional software module that allows the collection of IEEE 802.11 wireless links statistics and QoS management from the SDN controller. Ethanol architecture supports QoS programmability in a per-flow basis by utilizing a modified version of Pantou (OpenFlow software for commercial APs) that gives the SDN controller complete management of QoS parameters. In this way, Ethanol provides evidence that the OpenFlow protocol could be used to enforce QoS policies in IEEE 802.11 networks. This first development of Ethanol controller did not consider the definition of a northbound API, which discourages upper services/applications to collect IEEE 802.11 information from the controller on a standardized manner.

Some other researches have focused on extending the OpenFlow protocol for IEEE 802.11 networks. Patro et al. [17] develops the Coordination framework for Open APs (COAP), which is implemented with SOHO APs utilizing OpenWRT and the Floodlight SDN controller. The information exchanged between the SDN controller and APs includes the airtime utilization, user and neighboring APs statistics and beacon statistics among others. By identifying the traffic type on the wireless channel, the SDN controller could apply a throttled or slotted technique so that high priority traffic is not affected by low priority traffic. The SDN controller could keep track of previous wireless information regarding the AP surroundings (from IEEE 802.11 and non-IEEE 802.11 sources).

Kim et al. [18] present the OpenFlow AP architecture (OFAP) to provide seamless handoff and improve network throughput by mitigating the performance anomaly issue

in overlapped wireless environments. As part of the OFAP architecture, a wireless extension for the OpenFlow protocol was developed. This approach does not require any changes to user devices. As in Odin, users associate to a virtual AP instead of a physical one. In order to estimate the data rate of a given user to migrate, the controller periodically collects the user signal strength levels and data rates from all APs, and computes the correlation between user signal strength values and user data rates. As a result, the controller selects the most convenient AP for the user to camp on based on the current user data rate. However, the OFAP architecture requires all APs to be configured with same BSSID, SSID and IEEE 802.11a channel, so that users perceive that only one AP was presented. Though that configuration scheme might be applicable to some cases, it is not the most commonly scenario found in crowded places where BSSID, SSID and channels may vary from one network administrator to another. Finally, this research dis not provide any study on the impact of traffic load in the network due to the monitoring traffic, considering that each AP was required to compute and report the signal strength to the SDN controller periodically.

2.5 Research gap and contributions of this paper

The common limitation factor of the traditional 802.11 load balancing approaches was the lack of flexibility and integration regarding wireless and wired networks. Else, they required the presence of a particular IEEE 802.11 amendment and/or new network entity in the network, or the customization of existing network protocol to mitigate the problem. Thereby, these solutions made the network architecture more complex and disjoined.

On the other hand, SDN promises to address IEEE 802.11 issues from the network perspective in a standardized and flexible manner. Though some efforts regarding user management with SDN architecture were reviewed earlier, the network metrics utilized in these approaches (number of users associated to an over-loaded AP) were not adequate to reflect the load imbalance in data networks and the differences between user traffics.

This paper aims at developing a user management system for IEEE 802.11 networks by using the OpenFlow statistics to detect network asymmetry. Besides, this approach provides an approach to address broader IEEE 802.11 issues (i.e., mobility management, security, etc.), rather than tackle the particular load balancing problem. The proposed approach does not require protocol/standard customization in order to collect wireless information to mitigate the issue. Moreover, our design is not limited to an enterprise environments like commercial solutions, in which all APs belong to the same administration. Rather, it is conceived to address the issue in scenarios where multiple APs from the same or different services providers may be present. In these scenarios, where the network load may be distributed among APs of different service providers a reactive approach is more realistic.

Moreover, instead of utilizing the number of users as a load metric, the proposed algorithm uses APs and users' traffic rate, as well as, the users session duration to distribute load across APs. Our proposed system assists wireless user during migrations to ensure that the user performance is not significantly affected by unnecessary delay during the migration process. Lastly, due to the fact that using the same IEEE 802.11 channel on all APs might lead to a physical wireless channel congestion in crowded scenarios, the system presented in this work does not limit APs to be configured on the same channel as Odin and OFAP do.

3 System architecture

3.1 System overview

The proposed approach is a Wireless User Management System (WUMS) for WLAN networks under an integrated SDN architecture. The WUMS corrects unbalanced load conditions by moving users that generate excessive load on a given WLAN AP, to other under-loaded APs in the vicinity. By doing so, the WUMS maximizes the network throughput in order to improve the end-to-end QoS for wireless users.

The WUMS is a reactive mechanism in nature. Essentially, it monitors the network periodically for network asymmetry. Once an unbalanced condition arises, the WUMS determines the amount of load that must be redistributed in order to counteract the effect. However, moving the excessive load in a WLAN network means migrating the wireless users that are causing the issue. Hence, the WUMS selects some wireless users on the over-loaded AP and migrates them to other under-loaded APs. Eventually, the balanced condition is re-established in the network.

The WUMS requires a mixture of wired and wireless network information for its operation. On one hand, it periodically contacts the SDN controller via the northbound REST API to collect wired network information. However, in order to learn about the wireless environment, the WUMS also acquires wireless information from the associated users by contacting them directly.

The WUMS provides some assistance to wireless users during migration. As a result, the disconnection gap experienced by wireless users during the migration process can be reduced. This is mainly because the tasks related to the selection of the target AP (i.e., network scanning) are performed while the wireless user is still associated to the over-loaded AP. Therefore, wireless users do not need to waste time in scanning and re-association attempts in order to connect to another AP.

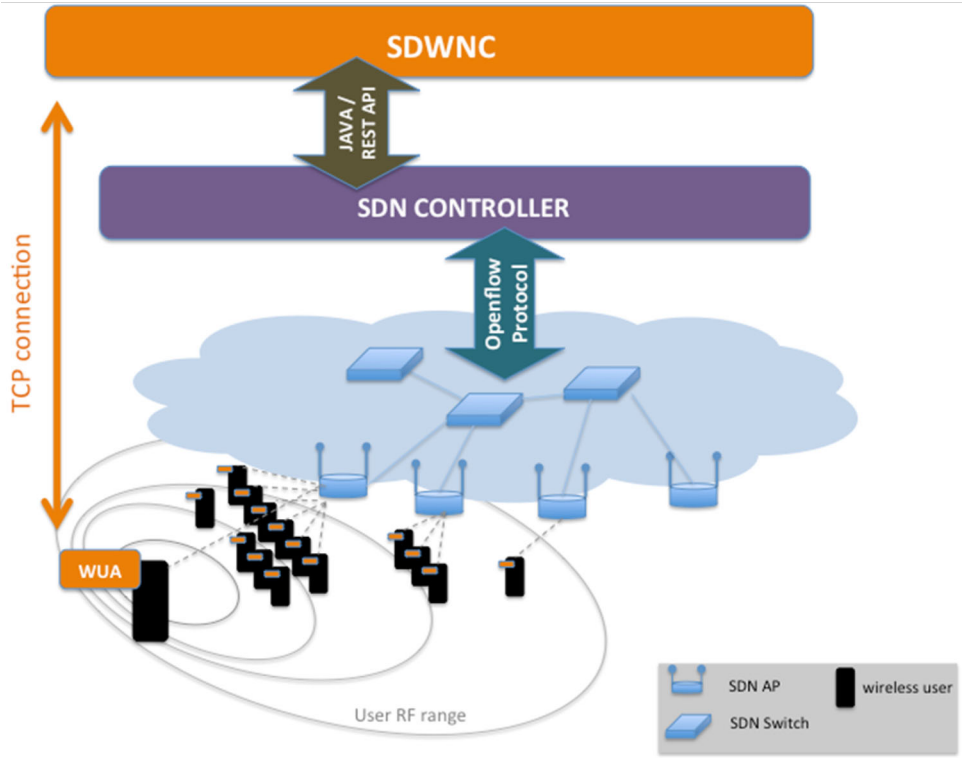


Fig. 1 WUMS architecture

The WUMS consists of two components working together. One is the Software Defined Wireless Controller (SDWNC) running on top of the SDN controller. The other one is the Wireless User Application (WUA), which is installed on every wireless user device. Figure 1 depicts the SDN environment for this WUMS.

The SDN network consists of switches and APs that are equipped with OpenFlow protocol. In this way, an SDN controller manages the forwarding decisions in the network. On the other hand, SDWNC is a central component in the architecture that communicate with both the SDN controller via the northbound interface (REST API) and with the multiple WUAs using TCP connections. Finally, WUAs inform the SDWNC about wireless information collected on their Radio Frequency (RF) range.

3.2 Assumptions and design considerations

- *Reactive user migration approach* The WUMS was developed as a reactive mechanism for unbalanced conditions in SDN wired/wireless networks. Hence, wireless users must be associated to APs in order for the WUMS to perform user migration.
- The WUMS allows administration of more than one SDN controller domain. The WUMS was initially conceived to collect network information from several SDN controllers, though this work considers only one SDN controller.
- The WUMS requires connectivity with the SDN controller and the wireless users at the same time.
- In order to achieve reliable control message exchange, the WUMS uses the connection-oriented protocol TCP.
- The WUMS monitors the load of those APs that belongs to the SDN topology only. It does not monitor non-OpenFlow devices.
- In this work, the WUMS considers only the traffic rate in the downlink direction, essentially assuming that the uplink traffic is, for all practical reasons, negligible comparing to downlink traffic.
- The WUMS does not consider inactive users. Inactive users are wireless users that have been associated with an AP for a long time, but they are not sending/receiving traffic load.
- IEEE 802.11n is utilized between APs and wireless users. No modifications to the IEEE 802.11 standard was required in WUMS
- According to the IEEE 802.11 standard [1], the wireless networks in this work are categorized as infrastructure-based
- APs are configured with distinct SSIDs in order to emulate crowded scenarios in which several networks are advertised
- APs operate in different wireless channels on the range of 2.4 Ghz. Each AP has one wireless interface where users associate with. SSIDs are all visible to wireless users.

- Wireless users are transient or stationary, and they are randomly distributed in different locations within the APs radio frequency range, aiming at emulating real scenarios such as coffee shops, hotspots, airports and other crowded places
- The SSID of a given AP has the same ID as the Mac address of the AP WLAN interface. In this way, the WUMS can link the AP information coming from the SDN environment (AP Mac Address) with the one collected from the wireless environment (SSID).
- The SDN controller must provide a northbound REST API interface for the WUMS to interact with the SDN network
- The SDN controller works in dynamic forwarding mode. This means that, based on the network status and traffic demands, the SDN controller install, remove and update flows on each SDN wired and wireless switch in an automatic manner.
- OpenFlow 1.3 was used as the SDN southbound protocol.

3.3 Software defined wireless network controller

The Software Defined Wireless Network Controller (SDWNC) is the principal component of the WUMS. This application is in charge of monitoring the network, detecting the asymmetry and redistributing users associated to over-loaded APs to other under-loaded AP in the vicinity. In general, the SDWNC receives wired and wireless network information via TCP connections either with the SDN controller or WUAs. The advantage of choosing this scheme is the flexibility towards various implementations. In fact, no modifications are required in the wired and wireless network protocols and standards to provide the WUMS with the required network information. Particularly, the SDWNC contacts the SDN controller to get the topology information and traffic rate measurements from the APs and associated users. However, IEEE 802.11 network information is not supported by current open-source SDN controllers. For this reason, the SDWNC collects the current association information and the available wireless networks that are seen by the wireless user from the WUA directly.

A major part of the SDWNC is an algorithm that performs wireless users migrations. Although the details of the algorithm will be explained in the next section (User Migration Algorithm), the following describes all the tasks and control messages exchanged between the SDWNC, SDN controller and WUAs in order to obtain the network information required by the algorithm.

Topology information: Usually, an SDN controller maintains an updated network topology database to make forwarding decisions. The SDWNC requires this source of information in order to perform all the user management operations.

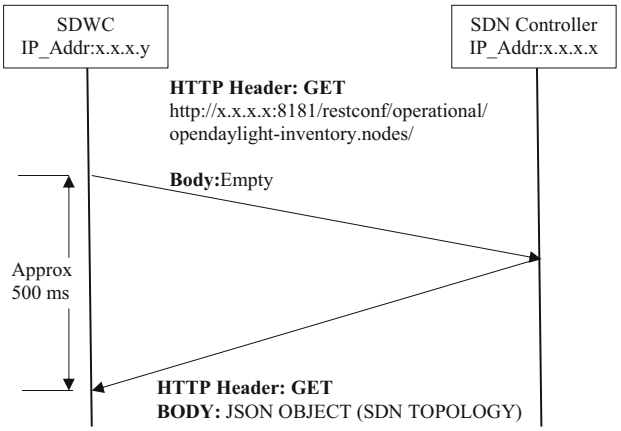


Fig. 2 Message diagram of SDN topology information

In order to get the required information, the SDWNC contacts the SDN controller via the northbound RESTCONF API interface. This interface supports HTTP operations such as OPTIONS, GET, PUT, POST and DELETE Moreover, the RESTCONF API allows access to the SDN controller database to configure and retrieve information [19]. Therefore, upon starting, the SDWNC sends a request to the SDN controller. The reply contains the topology database of the SDN controller, which provides the SDWNC the required data to retrieve contact and status information from the APs. Finally, the SDWNC builds an AP list to store MAC Address, IP addresses, OpenFlow IDs, and OpenFlow ports IDs from all the APs in the SDN topology. Figure 2 shows the messages exchanged between the SDWNC and SDN controller during this phase.

Monitoring: Once the SDWNC builds the AP list of the SDN topology, the monitoring process starts. During this phase, the SDWNC contacts the SDN controller to get the current traffic load information of the network. Essentially, the SDWNC gets OpenFlow port statistics of each AP of the topology. These statistics contains network status indicators that can be used to calculate the current traffic load on a given AP.

User analysis: This phase identifies the users' load and session duration on the over-loaded AP. In order to get this information, the SDWNC queries the SDN controller for OpenFlow flow statistics related with the wireless users associated with the over-loaded AP. After taking a number of samples, the SDWNC identifies the flows that belong to each user.

Decision-making: During this phase, a set of wireless users are chosen for migration. The phase considers the average traffic load of all APs, as well as, the traffic load and session duration of each user in the over-loaded AP. As a result of this analysis, the SDWNC provides a sorted list of wireless users to migrate. The sorted list includes the IP and Mac address

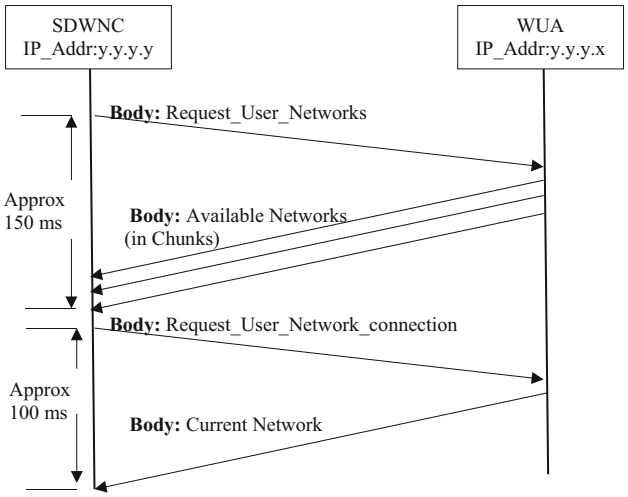


Fig. 3 Message diagram of available networks and current network information

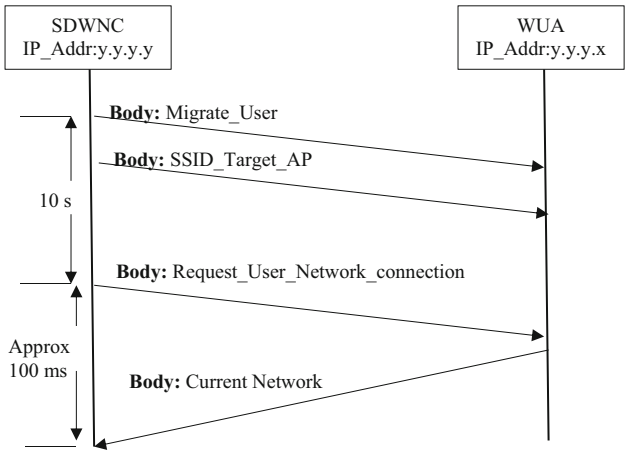


Fig. 4 Message diagram of user migration and confirmation after migration

of the selected users for migration. There is no additional information required to perform this analysis, and no control message is exchanged. A detailed description of this part is provided in the Sect. 4.

Contacting wireless users: Using the user list from the previous phase, the SDWNC contact the WUA on each user. This is to retrieve information of the wireless environment. The information exchanged includes the current association and the available networks seen by the user. Figure 3 shows the messages exchanged between the SDWNC and the WUA (of a particular user) during this phase.

Data processing: The SDWNC processes the responses received from the WUA in order to obtain the following information: SSID, AP MAC ADDR (BSSID), signal strength, and wireless channel. Due to the fact that there is no additional information required to perform these tasks, there are no control messages exchanged in this phase.

User migration and confirmation: The WUMS assists the wireless users during migration. In this process, the SDWNC opens a TCP connection with the WUA to send the target AP credentials. Once the credentials are sent to the WUA, the SDWNC wait for predefined period of time (10 s in our experiments) to contact the WUA again. Hence, the SDWNC verifies that the wireless user is currently associated to the target AP. The waiting gap provides the WUA with enough time to:

- execute the command to switch the user device to the target AP
- wait for the completion of the re-association and authentication process
- wait until wireless connectivity resumes (in the target AP)
- execute the "current networks" command that will returned the information of the target AP

It was observed during our experiments that setting the waiting gap too small (e.g. less than 10 s), causes the wireless card utility (in the user device) to halt or work erratically (when the WUA executes the current networks command after migration). However, as it will be shown in Sect. 5C, the actual disconnection time occurs independently of this issue, and it takes considerable less time than the waiting gap mentioned here.

Figure 4 shows the messages exchanged between the SDWNC and the WUA (of a particular user) during this phase. The SDWNC sends a "Request_User_Network_connection" message to the WUA. The SDWNC processes the response received from the WUA and gets the SSID in which the user is currently associate to. Lastly, the SDWNC checks the SSID received in the response with the SSID sent previously to the WUA (on the target AP credentials message). In this way, the SDWNC verifies whether the migration of successful or not.

3.4 Wireless user application

The Wireless User Application (WUA) is part of the WUMS and runs on every wireless device. It makes the WUMS aware of the wireless environment and provides support to the SDWNC during user migrations. The WUA is a lightweight Java-based software, and performs its actions upon request. It listens for incoming requests on a predefined TCP port (10007 in our experiments). As shown in Figs. 3 and 4, connections are always initialized by the SDWNC. Every time the WUA received a message request from the SDWNC, it executes commands from the wireless card utility of the user device.

The WUA plays an important role during user migrations. Once the WUMS determines the target AP for a given migrating user, the WUA receives the credentials to re-associate to another AP and performs the user migration. These credentials may include the target SSID and password for authentication and encrypted connections. Once the WUA switches the wireless user to the new AP, the SDWNC sends a request regarding the current association details of the wireless user ("*Request_User_Network_connection*" message). Consequently, the WUA executes the corresponding command internally and sends the output to the SDWNC. On the other side of the communication, the SDWNC gets the new SSID from the message, and uses it to verify that the user was successfully migrated to the target AP.

The assisted method provided by the WUMS does not rely on the traditional IEEE 802.11 approach, in which the user device is totally in charge of the re-association process. Although, the assisted method still introduces a minor connection disruption when the WUA moves a user to another AP, it is considerably smaller in comparison with the traditional approach. This is because the scanning process and re-association attempts, that are traditionally done when a user dissociate to a given AP, are avoided during the disassociation and re-association period.

4 User migration algorithm

The purpose of the User Migration Algorithm (UMA) is to control user migration across APs in order to improve the QoS in an integrated wireless SDN environment. The UMA resides in the SDWNC and works in a reactive manner to detect and correct unbalanced load conditions in the network. The user management decisions are based on the information gathered from both the wired and wireless SDN networks. In particular, the UMA evaluates the load in the network by measuring the traffic rate and session duration of all flows that each AP is supporting at the moment. When the current load of a given AP is greater than the average load of all APs, the UMA detects the asymmetry in the network and makes decisions of which users should be migrated and to which AP. Consequently, it re-distributes the load to other APs with vacant capacity.

The UMA was developed under the following set of assumptions:

- In order for the UMA to perform user management operations, traffic rate values of WLAN ports must be above a certain pre-defined threshold. This is to avoid the activation of the UMA when the traffic load is negligible for the network.
- Whenever an unbalanced condition occurs, the UMA analyze the over-loaded APs one at a time.
- There must be sufficient capacity available on nearby APs, in order to absorb the exceeded traffic load. Otherwise, balancing the load will not result in a positive effect on the wireless users' QoS.

As explained in the previous section, the WUMS collects real-time information from the SDN controller. This information serves as the input metrics to select users for migration. These metrics are the U_h and the *user session duration.* The U_h is defined as the average traffic rate of the user in the downlink direction. Although the WUMS gets information from both directions (uplink and downlink), the study is conducted to analyze the U_h in the downlink direction. This provides a direct way of measuring the load contribution of a user in the wireless network. In this way, the UMA can differentiate the load generated by each wireless user on the overloaded AP. The *user session duration* was considered to provide a second criteria for migrating decision. The *user session duration* provides information about wireless user behavior. Thereby, the UMA considers wireless users with recent sessions as eligible for migration, avoiding the connection disruptions on wireless users with long time session. Lastly, the UMA uses wireless information from the selected users to decide on the target AP where the user should be moved to. Basically, in order to perform the task, the UMA requires the SSID and signal strength information of nearby APs from the wireless user.

In order to determine whether the network is balanced or not, the UMA calculates the network *balancing factor* as following [3]:

$$\beta = \frac{\left(\sum_{i=1}^{I} P_i\right)^2}{I * \left(\sum_{i=1}^{I} P_i^{2'}\right)} \tag{1}$$

where I is the total number of AP found in the SDN topology and P_i is the Average port rate of each AP.

The closer the value of β is to one, the more balanced the network is. In our experiments, we used a target value of 0.9 to achieve a desirable level of balancing.

The UMA algorithm calculates the average port rate of the Access Point network to determine which APs are overloaded and which ones are under-loaded. Once the overloaded APs are determined, the user analysis on each AP starts by sorting the users based on their session duration, with the recently connected clients first. The algorithm favors migrating the new users to minimized disruptions to long- established sessions. Starting from the top of the list, the algorithm assesses each user for migration by evaluating the impact of its removal on the port load of the overloaded AP. The following condition controls the total amount of user load

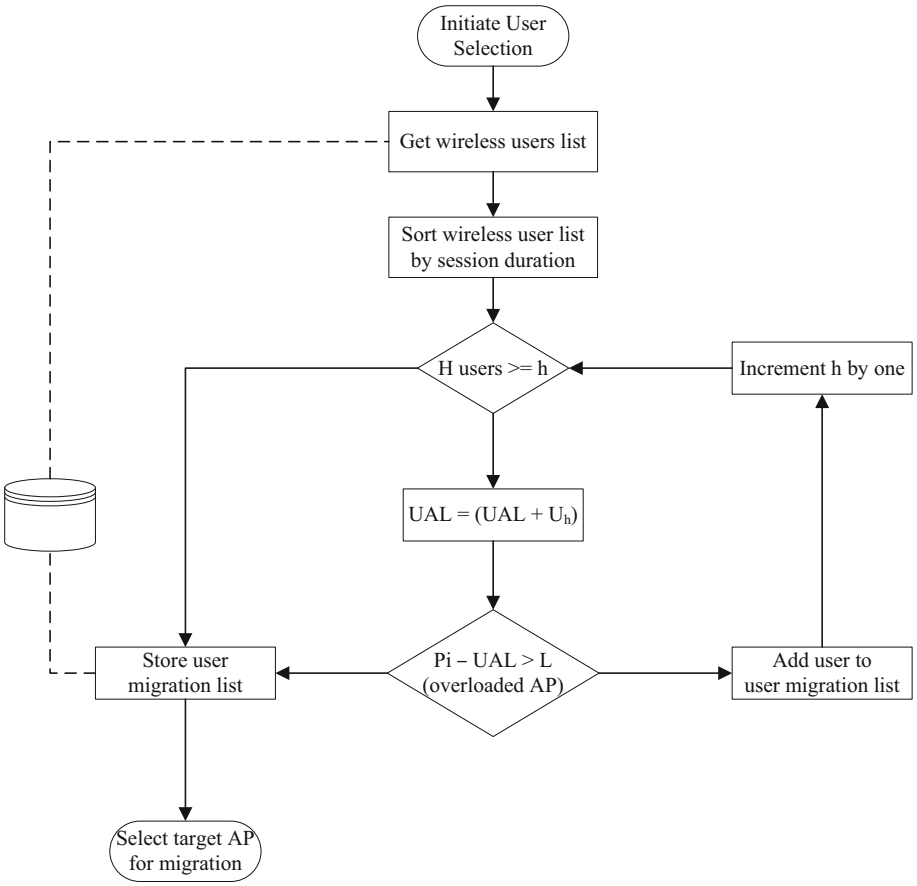


Fig. 5 Flow chart of process to select candidate users for migration

that the UMA will consider to migrate:

$$\left(P_{i(overloadedAP)} - UAL\right) > \bar{L} \tag{2}$$

In which P represents the average port rate of the overloaded AP, $\bar{L}$ is the average port rate of the entire network, and *UAL* indicates the total traffic rate of the users that have been marked for migration up to the current iteration of the algorithm. If the condition is true, the load on the over-loaded AP is still above network average. Hence, the UMA confirms the current user for migration and picks the next user to evaluate. However, when the condition is no longer valid, adding the current user would reduce the AP port rate (P_i) below $\bar{L}$. In this case, the UMA rejects the current user for migration, updates the *UAL* variable to the previous state (before adding the current user load U_h), and moves to the next user. Each time a user is marked for migration, its traffic load is deducted from the overloaded AP. Once the traffic load for this AP is reduced to the network average rate, the algorithm stops user analysis and returns the list of marked users for migration. Figure 5 shows the flowchart of the above process.

After compiling the list of candidates for migration, the target AP for each user is selected from the list of under-loaded APs (determined in the balancing factor analysis step) based on their signal strength for that particular user, which is provided by the WUA client process running on that user device. The SDWNC sends the target AP credentials to the WUA. After 10 s, the UMA verifies that the migration was successful, by comparing the current SSID (sent by the WUA) string with SSID string that belongs to the target AP. The UMA returns to the monitoring process when all users from the migration list have been successfully moved.

5 Performance analysis

5.1 Emulation model

We implemented our WUMS system on a testbed consisting of an OpenDayLight SDN controller and three 802.11 APs running Openflow-enhanced OpenWRT operating systems connected over a Gigabit/s Ethernet LAN. The user traffic was generated by setting up virtual media and web servers on the SDN test bed. The SDWNC module was installed on the same virtual server as the SDN controller. Figure 6 illustrates the architecture of the test bed.

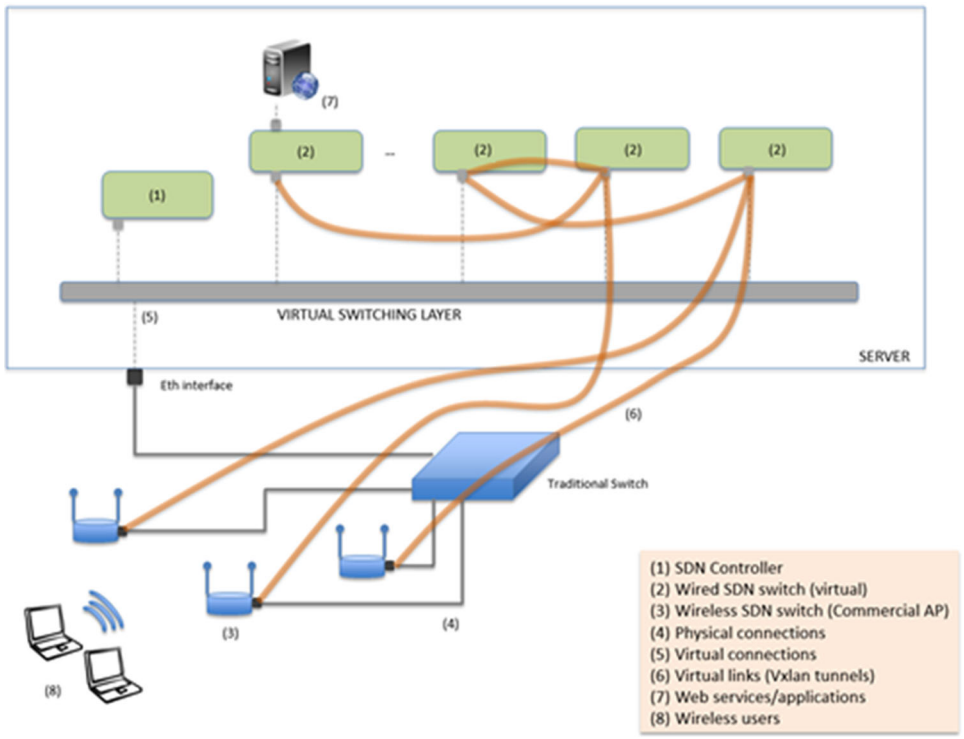


Fig. 6 Emulated software-defined wireless network

Table 1 User categories

User type	Application type	Traffic rate
Heavy	UDP streaming traffic	≥2 Mbps
Light	TCP web traffic	<2 Mbps

Wireless Users were emulated as virtual machine instances on laptops, with each VM accessing the APs through its dedicated USB wireless dingle. The users were classified according to Table 1.

A bandwidth limitation policy was configured on every AP of the test bed in order to test the WUMS and APs under congestion conditions, and to obtain network trace files manageable in size for later analysis. The policy affects user traffic (UDP and ICMP). However, it does not have any effect on TCP, which is used for the signaling traffic between SDWNC and WUA.

5.2 UMA performance results

In our first experiment, a total of 11 wireless users, each running a copy of WUA client, were connected to the APs. APs were configured with distinct SSIDs in order to emulate crowded scenarios in which several networks are advertised. APs were configured to operate in different wireless channels on the range of 2.4 GHz. This is to avoid the issues that result when the wireless medium is crowded, and thus concentrate the study in the analysis of load redistribution among APs.

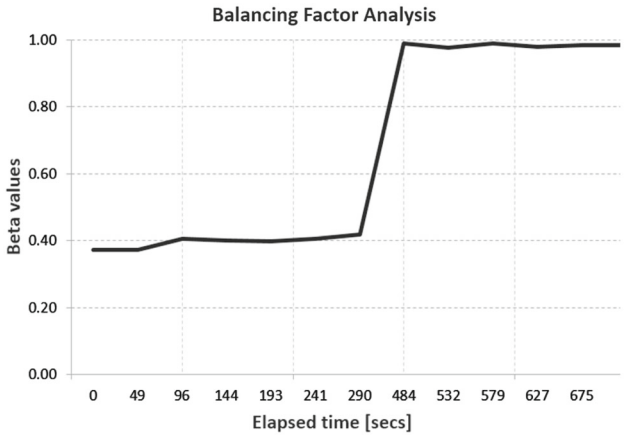


Fig. 7 Balancing factor indicator before and after the WUMS migrate the wireless users

Figure 7 shows the change in the *balancing factor* upon applying the UMA. From 0 to 290 s, the WUMS reports balancing factor values closer to 0.4, which clearly reflects asymmetry in the network. Next, from 290 to 484 s (194 s), the WUMS performs all the load-management tasks to accommodate the user load in the network. After migrations, the new balancing factor values computed by the WUMS are closer to 1. These results evidence that the WUMS performed the redistribution of the load successfully, and that the network is balanced.

The final load redistribution exhibits a smooth allocation of the user traffic among nearby APs. This is because the

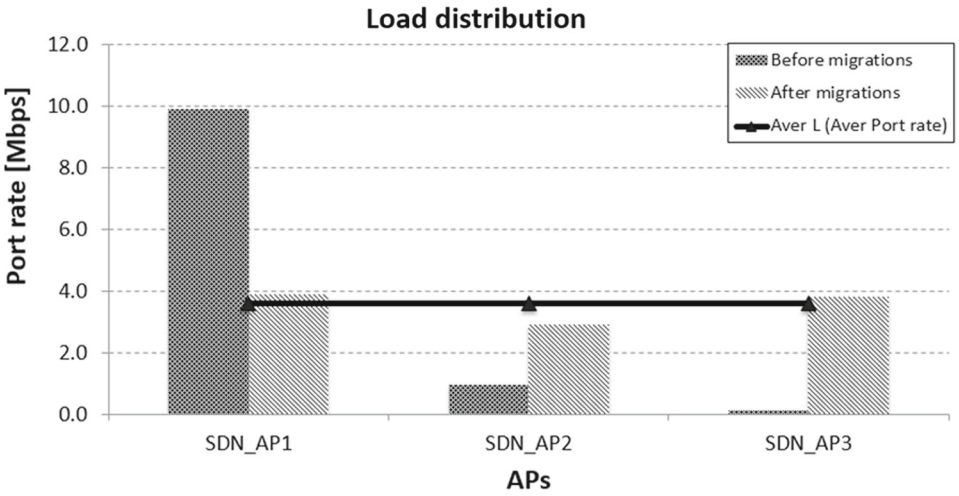


Fig. 8 Load distribution on each AP during the test

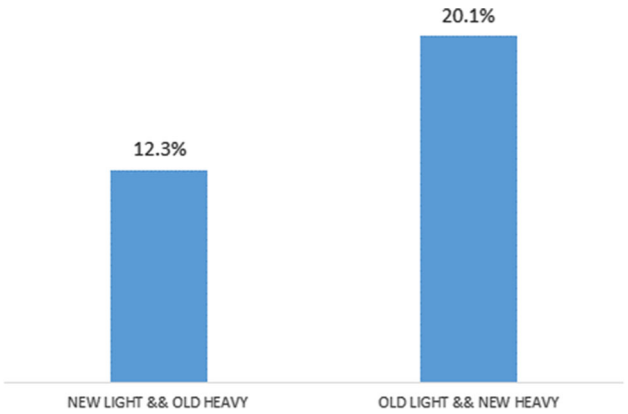


Fig. 9 UDP throughput improvement

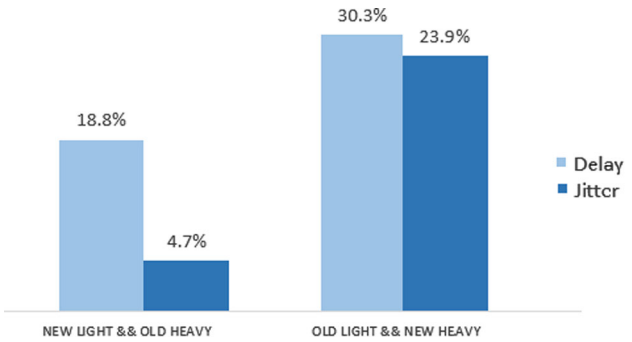


Fig. 10 Delay and jitter reduction

WUMS checks the current load in the under-loaded APs, before migrating the user. Figure 8 provides the overall port rate (P_i) supported by each AP during the experiment, before and after user migrations. The results show that, as expected, the WUMS redistributes the wireless users by migrating more users to the two under-loaded APs.

5.3 User load sensitivity scenarios

We also performed another set of experiments, considering scenarios with various mixture of heavy and light users. In these experiments, a total of ten wireless users were connected to the APs. The scenarios were divided based on session duration times into two groups: Scenarios in which the new users were heavy, and other scenarios where the new users were light. In each case, we considered a mixture of 80% heavy and 20% light users (80H/20L scenario), with an additional background traffic (BT) equal to 50% of the AP capacity in order to study the impact of congestion scenarios on the end-to-end QoS.

Our results indicated significant UDP throughput improvement, in particular in the scenarios where new users generated heavy traffic (Fig. 9). The 80H/20L scenarios provided a noticeable improvement in UDP delay and jitter (Fig. 10). In these scenarios, the total traffic load exceeds the bandwidth limitation policy configured in SDN_AP1, so congestion occurs frequently. In particular, the 80H/20L scenarios returns major improvements in jitter and delay when the UMA deals with Old Light/New Heavy (OL&NH) users. This is because moving the heavy users first and leaving light users in the AP helps alleviating congestion while it contributes in the network balancing tasks in a more efficient manner. On the other hand, in the case of New Light/Old Heavy (NL&OH), the UMA starts migrating the light users first, which does not provide significant improvement until heavy users are migrated. Consequently, in order to solve the imbalance condition in the network, the WUMS performs additional user migrations. In fact, the OL&NH scenario resulted in less migrations than the NL&OH scenario.

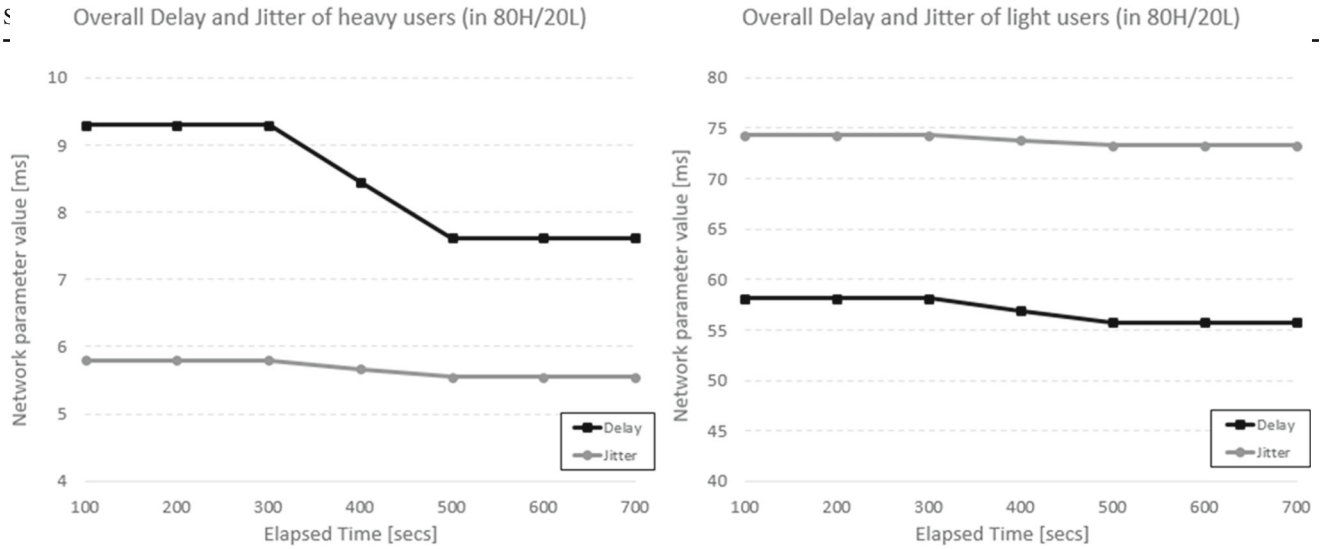
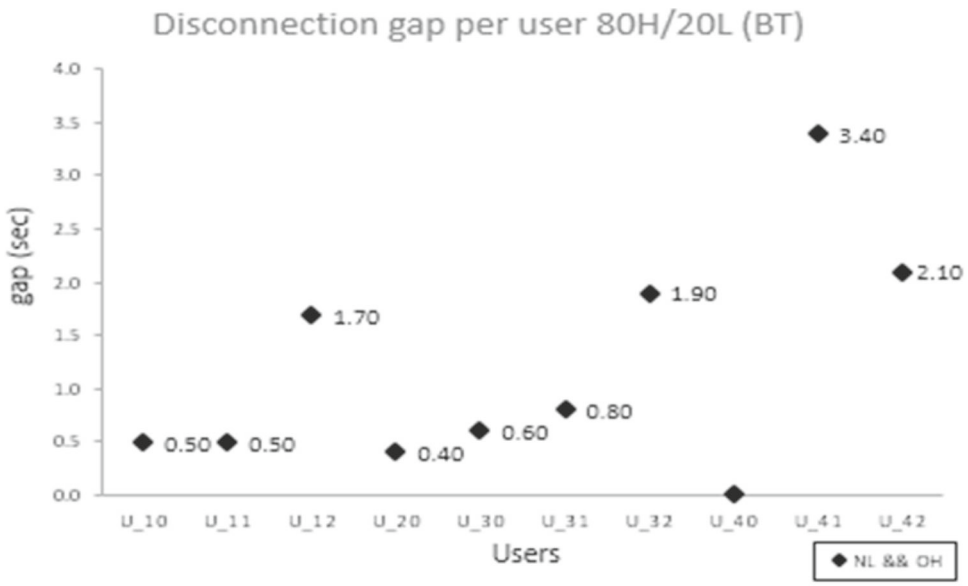


Fig. 11 Delay and jitter results versus time

Fig. 12 Disconnection gap per user



Another aspects to note is the dissimilar UDP delay and jitter improvement noticed on heavy and light users. Figure 11 depicts the values of heavy and light users in the 80H/20L scenario versus time as migration is conducted.

Heavy users receives video streaming (MPEG video) while light users receives audio streaming (MP3 audio). The video streaming traffic is more sensitive to delay and jitter than the audio streaming traffic. This makes heavy users more sensible to network changes than light users. In addition, the amount of traffic generated by a heavy user is 10 times greater than the one generated by a light user. When there is congestion in the AP, the traffic that belongs to a heavy user cannot be accommodated at the rate that the video application requires (approximately 2 Mbps). However, the traffic rate from a light user (200 kbps) can be transmitted in short bursts, making it more resilient to congestion. Therefore, the improvements achieved in delay and jitter were more evident on heavy users than in light users.

We also measured the disconnection time gaps, which indicate the connection disruptions while the users are being migrated from one AP to another. Our results indicate minimal disconnection gaps with WUMS. This is due to the assisted approach provided by the WUMS during user migrations. Without this approach, wireless users would experience extra delay due to network scanning and several attempts of re-association with other AP in the vicinity. In fact, the overall disconnection time of users in all scenarios is between 0.5 and 1.4 s (Fig. 12).

6 Conclusion

In this paper an OpenFlow-based design was proposed for a wireless user management system (WUMS) in an integrated wireless SDN environment. The implementation and the results clearly show that the performance of wireless users can be enhanced by utilizing the WUMS under a SDN architecture. The proactive user-management approach demonstrated improved results in congested scenarios, especially, when migrating the heavy users in the first place. While this system was used here primarily for load-balancing in a multi-AP wireless environment, it has the potential to be used for any other user management policy, due to its SDN-based architecture which allows centralized monitoring of traffic flows. The use of OpenFlow as the main communication protocol allowed for a flexible and standard operation between the controller and the wireless access points.

Acknowledgements This research was supported through funding from Natural Sciences and Engineering Research Council of Canada (NSERC).

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 1: Radio Resource Measurement of Wireless LANs Institute of Electrical & Electronics Engineers (IEEE).
2. High-Density Wi-Fi Design Principles. Aruba Networks. Available from http://docs.aerohive.com/pdfs/Aerohive-Whitepaper-Hi-Density%20Principles.pdf. Accessed on 10/16/2016.
3. Velayos, H., Aleo, V., & Karlsson, G. (2004, June). Load balancing in overlapping wireless LAN cells. In *2004 IEEE international conference on communications* (Vol. 7, pp. 3833–3836).
4. *OpenFlow Switch Specification, Version 1.5.1 (Protocol 0x06).* Open Netwokring Foundation (2014).
5. Nicholson, A. J., Chawathe, Y., Chen, M. Y., Noble, B. D., & Wetherall, D. (2006, June). Improved access point selection. In *Proceedings of the 4th international conference on mobile systems, applications and services* (pp. 233–245). ACM.
6. Yen, L. H., & Yeh, T. T. (2006, May). SNMP-based approach to load distribution in IEEE 802.11 networks. In *2006 IEEE 63rd vehicular technology conference* (Vol. 3, pp. 1196–1200). IEEE.
7. Sawma, G., Aib, I., Ben-El-Kezadri, R., & Pujolle, G. (2008, April). ALBA: An autonomic load balancing algorithm for IEEE 802.11 wireless networks. In *NOMS 2008–2008 IEEE network operations and management symposium* (pp. 891–894). IEEE.
8. Le, Y., Ma, L., Yu, H., Cheng, X., Cui, Y., Al-Rodhaan, M. A., & Al-Dhelaan, A. (2011, August). Load balancing access point association schemes for ieee 802.11 wireless networks. In *International conference on wireless algorithms, systems, and applications* (pp. 271–279). Springer, Berlin.
9. Stanley, D., Calhoun, P., & Montemurro, M. (2009). *Control and provisioning of wireless access points (CAPWAP) protocol specification, 2009*. IETF RFC 5415.
10. Aruba Networks Position Statement on CAPWAP. Aruba Networks (Dec. 3, 2015). Available from http://community.arubanetworks.com/aruba/attachments/aruba/115/422/1/CAPWAP+Position.pdf. Accessed on 10/16/2016.
11. Villegas, E. G., Ferre, R. V., & Aspas, J. P. (2006, June). Load balancing in WLANs through IEEE 802.11 k mechanisms. In *11th IEEE symposium on computers and communications* (ISCC'06) (pp. 844–850). IEEE.
12. Ryou, J. B. (2011). *Adaptive load balancing metric for WLANs*. Doctoral dissertation, Oregon State University.
13. Yap, K. K., Kobayashi, M., Underhill, D., Seetharaman, S., Kazemian, P., & McKeown, N. (2009, September). The stanford openroads deployment. In *Proceedings of the 4th ACM international workshop on experimental evaluation and characterization* (pp. 59–66). ACM.
14. Suresh, L., Schulz-Zander, J., Merz, R., Feldmann, A., & Vazao, T. (2012, August). Towards programmable enterprise WLANS with Odin. In *Proceedings of the first workshop on hot topics in software defined networks* (pp. 115–120). ACM.
15. Dely, P., Vestin, J., Kassler, A., Bayer, N., Einsiedler, H., & Peylo, C. (2012, December). Cloudmac—An Openflow based architecture for 802.11 MAC layer processing in the cloud. In *2012 IEEE Globecom workshops* (pp. 186–191). IEEE.
16. Moura, H., Bessa, G. V., Vieira, M. A., & Macedo, D. F. (2015, May). Ethanol: Software defined networking for 802.11 wireless networks. In *2015 IFIP/IEEE international symposium on integrated network management* (IM) (pp. 388–396). IEEE.
17. Patro, A., & Banerjee, S. (2015). COAP: A software-defined approach for home WLAN management through an open API.

ACM SIGMOBILE Mobile Computing and Communications Review, *18*(3), 32–40.
18. Kim, W. S., Chung, S. H., Ahn, C. W., & Do, M. R. (2014, May). Seamless handoff and performance anomaly reduction schemes based on OpenFlow access points. In *2014 28th international conference on advanced information networking and applications workshops* (WAINA) (pp. 316–321). IEEE.
19. *OpenDaylight platform, MD-SAL RESTCONF API*. (Dec. 3, 2015). Available from https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Restconf. Accessed on 10/16/2016.
20. Puthalath, L. S. (2012). *Programming the enterprise WLAN: An SDN approach*. Doctoral dissertation, Instituto Superior Técnico.

Eduardo Luengo received the bachelor of engineering's degree in Electronics and Telecommunications from the Pontifical Catholic University of Argentina, Buenos Aires, Argentina in 2008. Since then, he worked in the Telecommunications industry for 6 years, performing technical tasks related to wired and wireless networks at service providers and consulting. In 2014, he started his MASc in Computer Engineering at University of Ontario Institute of Technology (UOIT), Oshawa, Ontario, Canada. His master's thesis, "A load-balancing algorithm for Software Defined 802.11 Infrastructure-Based Networks", led to the publication of several papers. After his graduation in 2016, he worked as a Research Associate at UOIT in the topic of QoE on video streaming services. In August 2016, he joined a cyber security company in Toronto, Canada, where he is currently a Technical Analyst.

Shahram Shah Heydari is an Associate Professor at the University of Ontario Institute of Technology (UOIT), Ontario, Canada. Prior to joining the UOIT in 2007, he was a System Engineer and Member of Scientific Staff at Nortel Networks where he worked on element management in ultra high speed IP/MPLS routers, performance modeling of automatically switched optical networks (ASON), and proprietary voice-over-IP transport control protocols. His main research interests include network design and planning, software-defined networking, critical infrastructure resilience and security, and sensor networks. He received his B.Sc. and M.Sc. degrees in Electronic Engineering from Sharif University of Technology (Iran), M.A.Sc. degree from Concordia University, Montreal, and Ph.D. degree from University of Ottawa, Canada.

Khalil El-Khatib is an associate professor at the University of Ontario Institute of Technology (UOIT). Prior to UOIT, he was an assistant professor at the University of Western Ontario until July 2006. Between the years of 1992 and 1994, he worked as a research assistant in the computer science Dept. at AUB. In 1996, he joined the High Capacity Division at Nortel Networks as a software designer. From Feb. 2002, he worked of Research Officer in the Network Computing Group (lately renamed the Information Security Group) at the National Research Council of Canada for two years, and continued to be affiliated with the group for another two years. His current research interests include:

- Smart Communities for Smart Cities
- Big data and security analytics
- Security and privacy issues in wireless sensor network, mobile wireless ad hoc networks, and vehicular ad hoc networks,
- Smart grid security,
- Cloud computing,
- Biometrics,
- Ubiquitous computing environments,
- E-health.