ORCHESTRA: Virtualized and Programmable Orchestration of Heterogeneous WLANs

Ensar Zeljković*, Tom De Schepper*, Patrick Bosch*, Ian Vermeulen, Jetmir Haxhibeqiri[†],
Jeroen Hoebeke[†], Jeroen Famaey*, and Steven Latré*[†]
*University of Antwerp - imec, IDLab, Department of Mathematics and Computer Science, Belgium
[†]Ghent University - imec, IDLab, Department of Information Technology, Belgium

Abstract—local area networks (LANs) are employed by a plethora of heterogeneous consumer devices, equipped with the ability to connect to the Internet using a variety of different wireless network technologies. Existing solutions and the lower layers of the OSI stack are unfit to cope with this heterogeneity. For instance, dynamical inter-technology switching is user- of application-based We propose the ORCHESTRA framework to manage the different devices in heterogeneous wireless local area networks (WLANs) and introduce capabilities such as packet-level dynamic and intelligent handovers (both inter- and intra-technology), load balancing, replication, and scheduling. The framework consists of a controller that is capable of communicating with both existing Software-Defined Networking (SDN) and Network Function Virtualization (NFV) controllers and with devices containing a newly introduced virtual Medium Access Control (MAC) layer. We show that the virtual MAC enables transparent and real-time inter-technology handovers and that our solution scales up to two thousands of clients.

I. INTRODUCTION

Today's LANs consist of a plethora of heterogeneous consumer devices, equipped with the ability to connect to the Internet using a variety of different network technologies (e.g., LTE, Bluetooth, IEEE 802.11n, and IEEE 802.11ac). Over the next few years, the diversity among devices and technologies is expected to expand further with the rise of all kinds of Internet of Things (IoT) devices, multimedia services, and the availability of new technologies such as glslte-A, 60 GHz Wi-Fi, visible light communications, and Bluetooth 5.0. These, mostly wireless, devices and applications have stringent and diverse quality requirements (e.g., high throughput for 3D video applications) and are very sensitive to network disruptions and degradations (e.g., high latency, congestion, or link failures). On the other hand, every technology has specific characteristics in terms of, among others, maximum throughput, latency and range.

Managing this complex puzzle of heterogeneous devices and technologies at the same time, while providing the desired Quality of Service (QoS) is currently not possible. The current lower layers of the OSI stack (in particular the datalink up to the transport layer) are unfit to cope with this heterogeneity. Switching between technologies or load balancing is delegated to the application layer, or even worse, to the user. This leads to a very static and sub-optimal management of these wireless networks, making it impossible to automatically react in a timely fashion to dynamic network changes (e.g., disruption or changed. number of devices). While existing solutions, such as the IEEE 1905.1 standard and Multipath Transmission Control Protocol (MPTCP), do allow for dynamic flow redirection or simultaneous interface usage for a single flow, respectively, the necessary coordinated intelligence and level of control is missing [1, 2]. All of this indicates the need for intelligent and dynamic inter-technology and access point (AP) routing and interface selection, which enables optimizations such as multi-path routing, load balancing, and dynamic path reconfiguration, aiding to unlock the network's full potential.

In this paper, we introduce a software-defined framework that relies on network virtualization to cope with the above described heterogeneous challenges and is able to support inter-technology management. This framework is called ORCHESTRA, as it orchestrates between technologies. It consists of two major parts: a virtual layer (OVL) and a ORCHESTRA controller. The OVL unifies the MAC of the different supported technologies on a single device, providing a single socket for connectivity to both the application layers and the ORCHESTRA controller. An SDN-based approach is used where a set of policies can be defined to control the MAC behavior on a packet level. The controller is capable of managing both OVL-enabled or legacy devices across the entire network, and decides, amongst others, on technology and path selection, AP assignment, and channel access. This framework allows for the implementation of the previously mentioned optimizations like intra- and inter-technology handovers, load balancing and dynamic path reconfiguration.

The contributions of this paper are fourfold: First, we introduce the novel ORCHESTRA controller that handles the decision making and is capable of communicating with all devices within a heterogeneous WLAN, including legacy devices. Second, we propose a virtual layer, called OVL, that is more intelligent and lightweight then current state-of-the-art. Third, we formulate the problem of dividing traffic and devices across different technologies and APs as an Lineair Programming (LP). Novelties are the consideration of both multiple technologies and APs and the usage of real-time monitoring information, instead of assuming full knowledge across the entire network. Fourth, we demonstrate the capabilities of ORCHESTRA through a real-life prototype and emulations.

The remainder of this paper is structured as follows. We start by giving an overview of the current state of the art in Section II. Next, we describe the ORCHESTRA and OVL architecture and functionality in Section III, while the load

balancing problem and algorithm are stated in Section IV. Section V discusses the results. Finally, conclusions and future research directions are provided in Section VI.

II. RELATED WORK

In this section we will discuss existing work on inter- and intra-technology handovers and load balancing. The idea of introducing a layer between the existing MAC layers and the network layer, similar to the presented OVL, has also been proposed in the IEEE 1905.1 standard [1]. Devices that are compliant to the IEEE 1905.1 standard have an abstract layer hiding the underlying diversity in MAC technologies. With a unique virtual MAC address representing each device on the network, it is possible to transparently switch flows between multiple heterogeneous interfaces. In contrast to our packet-based solution, the IEEE 1905.1 standard only grants a flow-level control over the network. The implementation of the IEEE 1905.1 standard results in a simplified set-up, configuration and operation of network devices with heterogeneous technologies. However, it only supports Ethernet, Wi-Fi, power-line home plug and Multimedia over Coax (MoCA). Although IEEE 1905.1 specifies the features to enable dynamic flow redirection (e.g., the protocols), it does not define the algorithms for selecting suitable paths per flow. To this extent, load balancing algorithms have been proposed, including ones that focus on energy efficiency [3, 4, 5, 6]. However, existing work mostly focuses on theoretical models that assume the detailed knowledge of flow throughput requirements and dynamic network conditions. For instance, the specific nature of wireless networks (e.g., interference, link quality variability) is ignored. This was addressed in previous work which relies on real-time monitoring information [7]. However, it does not take into account multiple APs which is crucial for realistic state-of-the-art networks.

Some alternatives for a virtual MAC based on SDN have been recently proposed that bring SDN into LANs [8, 9, 10, 11]. These alternatives rely, for instance, on an OpenFlow OpenFlow (OF) controller (e.g. Ryu) and Open vSwitch (OVS) [12, 13]. The OVS reports in real-time monitored flow information (e.g., counter for the number of packets and bytes) to the controller by making use of OF stats request and reply messages. The controller then decides to handover a subset of the flows to an other technology if desired by changing the outgoing OF port. This can, for instance, be needed when more flows arrive in the network and the throughput of current flows start decreasing. Another SDN solution worth mentioning, is the 5G-EmPOWER framework that focuses on virtualized network functions in wireless networks [14]. In line with the thought of NFV, it moves intelligence from an AP to a controller. Currently the focus lays on the following control aspects: wireless clients state management, resource allocation, network monitoring, and network reconfiguration.

Related research can be found in so-called heterogeneous networks (HetNets) or 5G networks, where load balancing, dividing connections across different technologies and handovers across LTE/UMTS, Wi-Fi, and WiMAX have been investigated. Most research proposes technology specific solutions that are capable of performing handovers or load balancing across only two of these technologies (e.g., LTE and Wi-Fi or Wi-Fi and WiMAX) [15]. The decision to perform a handover is made centrally by the base station and different decision strategies have been proposed using, among others, utility functions, multiple attributes decision making, Markov chains, game theory, and user location [15, 16]. These strategies take only a limited number of parameters into account, with Received Signal Strength Indicator (RSSI) and signal to noise ratio (SNR) being the most popular ones [17, 18]. Open issues include, for instance, the development of more generic solutions, better support for mobility, the use of multi-criteria decision functions, supporting different QoS classes and the increase of QoS during or after handovers [19]. Current solutions are technology specific and do not take actual application or QoS parameters and objectives into account, making them unsuitable for use in with QoS-sensitive or mission critical applications.

A complementary piece of research, worth mentioning is MPTCP [2, 20]. This TCP extension enables the transmission and reception of data concurrently on multiple interfaces. While MPTCP shares its goal of improving QoS and network resource utilization with our ORCHESTRA framework, it focuses only on the alternative paths between two hosts and not on a network-wide scale

To summarize, current presented architectures contain only a single approach, either with an OVL or a SDN-based setup. Supporting both options would result in a faster aoptable and more extensive solution that allows for the usage of different network optimizations. One of the most impactful optimizations is load balancing. Current load balancing algorithms lack practical applicability, assume full knowledge over the network and only take into account a limited number of technologies. Our presented architecture and algorithm will offer significant improvements in all of these areas.

III. ORCHESTRA ARCHITECTURE

In this section, we define the functionality of our approach, describe the capabilities of the controller and introduce the OVL.

A. Offered functionality

Our solution has two modes of operation, which correspond to the type of client devices that are available in a network. The first mode corresponds to devices that have the proposed OVL implemented. This offers all the available flexibility, control and QoS. The following functionalities are offered:

a) Load balancing: Packets are balanced over different technologies. This can be differentiated by applications to support different QoS requirements or undifferentiated but with the goal to increase throughput and utilize all the technologies to their fullest.

b) Replication: Packets are duplicated over different technologies to increase reliability. This can increase throughput or reduce delay in difficult environments, such as high interference environments.
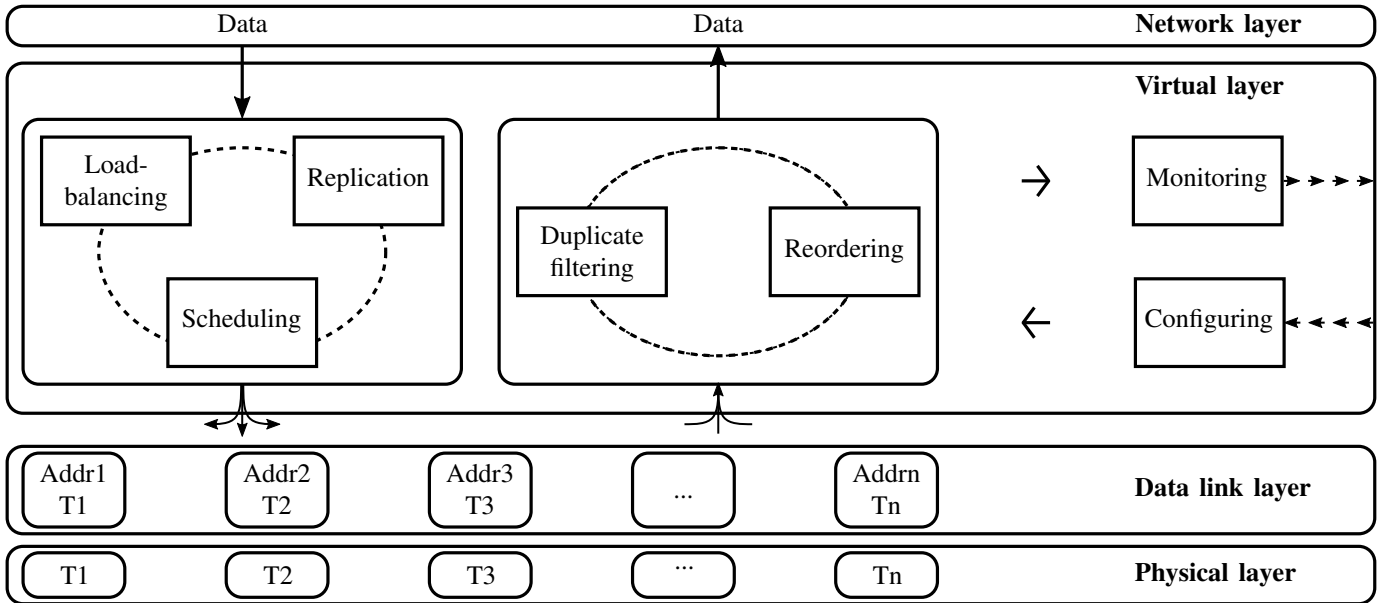
Fig. 1. The ORCHESTRA virtual layer is on top of the data link layer and acts as a data link layer itself to the network layer.
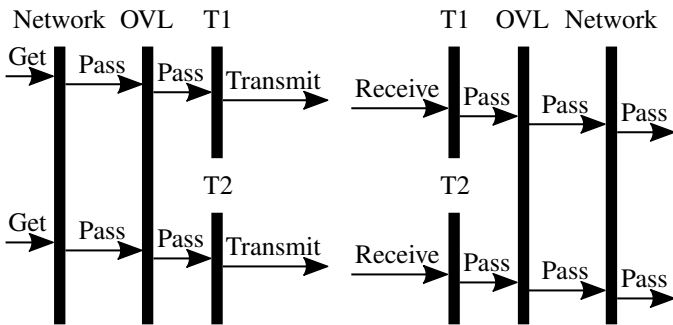

Fig. 2. Load balancing distributes the traffic over multiple technologies.

c) Scheduling: Applying a time-division multiple access (TDMA) based schedule on top of the access control of the technology. This allows for a coarse grained control of transmission, which is especially useful with a high amount of clients. This way it is possible to cope with interference from neighboring stations within the network, from neighboring networks or other sources outside of our control.

d) Handovers: Moving a client from one AP to another while keeping the current technology (horizontal handovers) or moving a client from one technology to another while keeping the current AP (vertical handovers) or changing both. This allows completely seamless roaming, without loss of connection, and guarantees that the best technology and AP combination is used.

The second mode of operation is applied to devices that, compared to our solution, offer legacy functionality. These devices do not have our virtual layer and are therefore limited in functionality. Out of the four areas of offered functionality, only handovers can be performed: it is possible to perform intra-technology handovers (i.e., roaming between APs) if

this is supported by the wireless framework. Inter-technology handovers are, however, limited to band steering, where the infrastructure tries to force a client to a certain frequency band (e.g. from IEEE 802.11n at 2.4 GHz to IEEE 802.11ac on 5 GHz). This is highly depending on the client itself and does not work for all devices. Furthermore, no QoS guarantees can be provided and it is likely that connection loss of several seconds is incurred.

B. ORCHESTRA virtual layer

Currently connectivity is handled on an interface basis. This means changing the interface results in a connection loss. We solve this issue by introducing a virtual layer and abstracting connectivity from the user and applications. It also enables us to implement functionality that works across multiple technologies.

Our OVL implementation is located above the existing MAC layer and below the network layer in the OSI model, as shown in Figure 1. Existing layers are thus not modified and packets are still passing through them. Our virtual layer only uses the depicted functionality to organize the different MAC layers and let it appear as one to the networking layer. As a consequence only a single IP address is needed, while a single unique virtual MAC is not required.

The data link layer has no knowledge about the OVL and behaves like in a normal network stack, forwarding packets upwards. They then pass through our OVL, are filtered or ordered, and are then passed to the network layer. Similar, a packet is passed from the network layer to the OVL and passes our functionality, where load balancing, replication or scheduling is applied, and is then passed to the correct interface at the correct time. Our approach is packet-based and not flow-based. This means that, for instance, load balancing can be done on a packet level instead of a flow level, which allows

for more flexibility and performance, especially if a flow is consuming all of the available bandwidth of a technology. The OVL is configured through messages from the controller which include rules that define how each type of packet is processed and through which technologies it is sent. These rules can, for instance, be based on source and destination addresses, packet types or on weights denoting how many packets should be forwarded to each interface accordingly. All of the passing packets are monitored and aggregated into statistics for each technology and then forwarded to the controller through the monitoring module.

a) Load balancing: Load balancing is realized by assigning a certain percentage of the packets to a specific technology. The OVL implements this by using the weighted round robin principle that allows to assign a defined amount of packets to a technology before it switches to the next one. Reordering of packets is done by keeping a sorted list of packets and the last sequence number that was forwarded to the network layer. When a packet arrives, we check if the sequence number is the next expected one and if that is the case, we immediately forward it, increase the last sequence number, and check the head of the list if it is the next one. If this is the case, we remove the head and continue checking the head of the list, until a packet is missing. If the arriving packet is not the next one, we check the head of the list, forward it, and remove it, if it is the next in line. Again, we do this until a packet is missing, if the arriving packet is the missing one, we forward it and continue, otherwise, we insert it at the first position, where it has a smaller sequence number than the current packet at that position and shift the whole list. This sequence of described actions is also shown in Figure 2.

b) Replication: Replication can be done for specific packets or for all packets of a flow. In this case, every packet is duplicated and forwarded over all specified technologies. On the receiving side, we maintain a hash table of received packets and through the configuration we know how many duplicates should arrive. Additionally, after a timeout, packets in the hash table are removed. If an arriving packet is not in the hash table yet, it is forwarded to the upper layer immediately and a new entry in the table is made. If there already is an entry, then the packet is dropped and, if the limit of possible duplicates is received, the entry is removed.

c) Scheduling: The scheduling functionality is on top of the scheduling of the corresponding technology and uses multi-frequency time-division multiple access (MF-TDMA) as it is not only assigning a time slot, but also a frequency. E.g. for Wi-Fi this means that it is on top of carrier sense multiple access with collision avoidance (CSMA/CA). This also means that it is not as fine-grained as implementing it directly in the corresponding technology's MAC layer, but especially for Wi-Fi it still offers advantages e.g. to avoid external interference. Scheduling is realized by maintaining a queue for each technology and a schedule that is received from the ORCHESTRA controller. When a free slot is available for a technology, the first packet or first packets, depending on the slot time, is forwarded to the corresponding MAC layer. The MAC layer of the technology then transmits it like it would any other packet. The amount of packets that are sent in each time slot is determined by the controller and are communicated as well. The controller tries to predict the amount of interference according to historic data and computes out of that the amount of packets.

d) Handovers: For a handover all involved parties (the client, the APs with the corresponding technologies, and, if necessary, switches) are informed and configured to accommodate for the moving of the client. For a client itself, this means that it starts negotiating with the old AP and technology when exactly to move and for the new AP when exactly to start using it. This is important to make sure that downstream traffic arrives correctly. For this purpose the client synchronizes the time with both APs and then offers the earliest time the change can happen. The APs either agree or propose another time until all parties agree. During the time of the switch, the virtual layers on the client and the new AP buffer the packets and send them as soon as the switch is acknowledged. To support this, the AP needs to be controllers by a SDN/NFV controller or have the OVL as well.

C. ORCHESTRA controller

While the OVL allows achieving fine-grained MAC control inside a single node, the ORCHESTRA controller enables management - orchestration of the entire network. This is illustrated in Figure 3. The controller combines all network logic and is the single point of decision making. This includes the assignment of clients to end points or the route which packets should take.

As can be seen in Figure 3, the ORCHESTRA controller is one step higher in the hierarchy than existing SDN/glsnfv controllers it interfaces with. It provides interfaces to these existing SDN/NFV controllers, both for wired and wireless networks. Additionally, it provides interfaces to directly communicate with switches, APs, and client devices. The communication for client devices and APs is realized through our virtual layer. Through the interfaces, the ORCHESTRA controller collects monitoring information and issues commands and reconfigurations.

a) Interaction with infrastructure components: The ORCHESTRA controller takes advantage of current SDN/NFV controllers with their capabilities to communicate with either switches or APs. This is usually realized with OpenFlow in the case of switches or a proprietary protocol in the case of the APs. Current controllers can instruct the hardware how to handle packets and clients. This functionality is utilized by the ORCHESTRA controller through a northbound interface on the respective SDN/NFV controller as well as our own. Most SDN/NFV controller implementations allow running applications on top of them, which makes the application independent from the core implementation. This allows for the creation of an application that acts as an interface towards the ORCHESTRA controller in order to relay commands and monitoring information (Figure 3). In detail, we use the underlying API to get
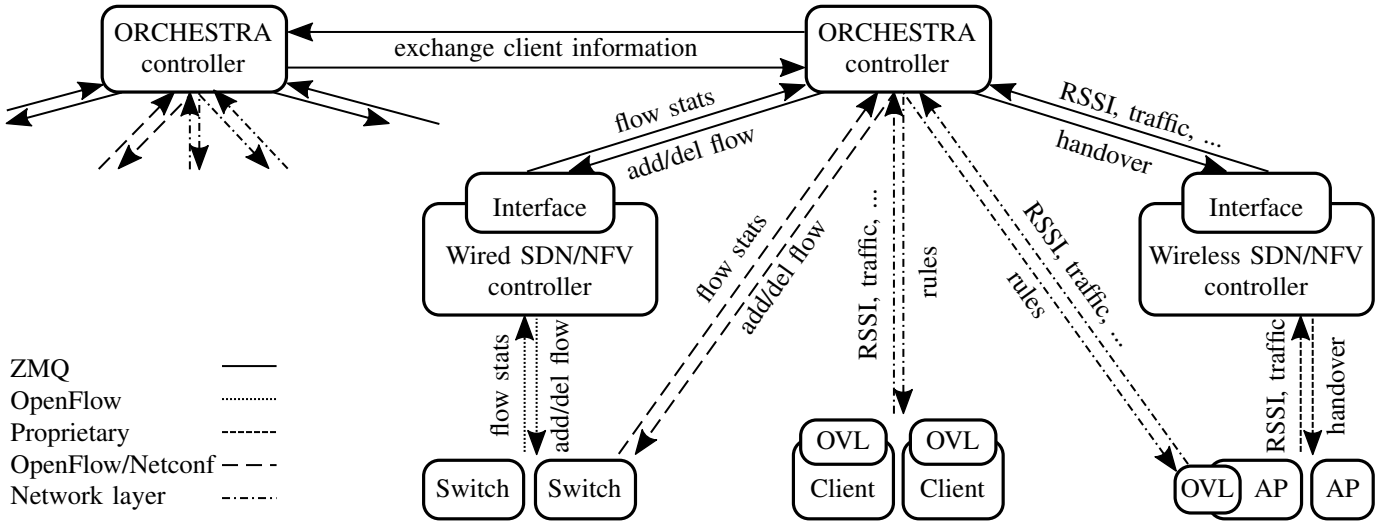
Fig. 3. ORCHESTRA is the center of network management and coordinates existing SDN-based controllers as well as clients through different protocols.

throughput per flow and topology information from the wired SDN/NFV controller and device to AP assignment, device throughput, and device to AP RSSI values from the wireless SDN/NFV controller. We also use it to issue flow rules to the switches and to assign devices to APs. Communication with our ORCHESTRA controller can be realized through different kinds of communication frameworks or protocols.

b) Interaction with clients: For communication with the client, we use the OVL as the end point on the client. Here the controller uses network layer traffic without a specific transport protocol as the virtual layer filters out those messages and applies the configuration. The controller can instruct the client which technologies to use and if load balancing, replication, or scheduling should be applied. For this, the controller sends a rule set with what to filter on and what to apply. IP addresses, transport protocol, or packet type serve as filters whereas the previously defined functionality serves as what is to be applied. Messages from the controller are recognized by being only network layer traffic and by an identifier in the first 64 bytes of the payload of the packet. Handovers to a different technology or AP can be issued as well. The minimum message for a handover includes the current AP - technology configuration and the new AP - technology configuration the client should switch to. This can also be several combinations as several technologies can be active at the same time. The ORCHESTRA controller instructs the wireless SDN/NFV controller and the OVL of the affected AP as well to inform them about the handover and to start synchronizing the handover with the client. The OVL sends back monitoring information that includes the available technologies, its required throughput, its QoS requirements, and the signal strength for each technology and AP it can see.

c) Storage and decision-making logic: The ORCHESTRA controller itself has two parts besides the communication interfaces. The first part consists of a data store where all information is aggregated and combined

into one state model. This model consists of the topology information from the SDN/NFV controller responsible for the backbone network, including the APs as transition points to the wireless network, and the throughput of each flow on each switch. Additionally, it includes the current assignment of clients to APs and technologies, as well as the RSSI and throughput of each client and technology. The second part includes the decision logic which contains one or multiple algorithms that perform certain functionality. An example of this is the load balancing algorithm, presented in Section IV, that distributes devices among APs and technologies to optimize throughput for all devices. Other algorithms can, for instance, also focus on TDMA based scheduling or even on energy efficiency. The decision logic uses the aggregated information of the storage as input and provides a certain configuration for the network. In the case of the load balancing algorithm this is a client to AP to technology assignment as well as the correct flows on the switches. Based on this configuration, the necessary commands are issued to the corresponding devices across the network to actually roll-out the configuration.

d) Scalability and controller distribution: To tackle the challenge of scalability, our controller is distributable as well. State information about common devices is shared between controllers. This includes the traffic requirements and more importantly, the RSSI values to estimate the distance. Only information of devices that both controllers have information on will be shared among the controllers to reduce overhead. For example, a device sees two APs and is connected to one of them. One of the APs is in the region of the first controller and the other AP is in the region of the second controller. As both controllers have information on a device, it is shared among them to consider them when a handover is needed. If a handover is needed because a newly computed assignment would place it in the region of another controller, the controller currently responsible for the device informs the other

controller to take over the device. The new controller then updates its flow rules and AP configuration and acknowledges the handover. After that, the old controller deletes the flow rules and the AP configuration and only monitors the device.

IV. MULTI-TECHNOLOGY AND-AP LOAD BALANCING

In this section we discuss in more detail the problem of scheduling traffic across the different available technologies and assigning devices to APs. This problem allows us to illustrate an important subset of the capabilities of ORCHESTRA. We define a precise mathematical description of the problem via an LP formulation as follows:

The topology of the LAN is defined as a triple (S,A,T) where:

- S is the set of stations $\{s_1, s_2, \ldots, s_n\}$. These stations represent the different wireless consumer devices within the LAN (e.g., smartphone, tablet, or laptop). Furthermore, we define r_s to be the total desired traffic for station $s \in S$. Due to the fact that wireless technologies have a shared medium, we do not have to separate the traffic rates between incoming and outgoing flows.
- A is the set of APs $\{a_1, a_2, \ldots, a_n\}$.
- T is the set of technologies $\{t_1, t_2, \ldots, t_t\}$ and $c_{a,t} \in \mathbb{R}_0$ represents the total capacity for a technology $t \in T$, given an AP $a \in A$. By assuming the capacity to be dependent on a combination of an AP and technology we acknowledge the fact that there can be configurational differences between two APs providing the same technology within the LAN, which can have an impact on the capacity of that technology. For instance, an AP can provide 2.4 GHz Wi-Fi with a channel width of 20 MHz, while a second AP can provide the same technology but with a channel width of 40 MHz

The inputs for the Integer Linear Programming (ILP) formulation consist of the previously defined network model, as well as the following sets:

- T_s: $\forall s \in S$: the list of technologies $t \in T$ that are supported by a station $s \in S$.
- T_a: $\forall a \in A$: the list of technologies $t \in T$ that are provided by an AP $a \in A$

We define the following decision variables:

- $\lambda_{s,a,t} \in [0,1]$; this variable represents the percentage of the total rate r_s of a station $s \in S$ that is sent over the technology $t \in T_s \cap T_a$ provided by AP $a \in A$. In other words, a station can be connected to different APs at the same time (if a technology is shared among station and AP) and its traffic can be divided across those multiple paths. Zero means that no traffic is sent over that specific technology.
- $\tau_{a,t} \in \mathbb{R}_{\geq 0}$; this variable defines the assigned rate to a certain technology $t \in T$ provided by AP $a \in A$.

As an objective function, the model maximizes the total assigned rate (bandwidth) over all provided technologies:

$$max \sum_{a \in A, t \in T} \tau_{a,t} \qquad (1)$$

Finally, we define the following constraints:

- The capacity constraint makes sure that for each combination of technology and AP, the total possible capacity is not exceeded:

$$\forall a \in A, \forall t \in T : \tau_{a,t} \leq c_{a,t} \qquad (2)$$

- The multipath percentage constraint makes sure that in total all traffic is divided across the multiple paths for a station:

$$\forall s \in S : \sum_{a \in A, t \in T} \lambda_{s,a,t} = 1 \qquad (3)$$

- The final constraint makes sure that the total rate across all technologies and APs is at most the rate desired by all the stations:

$$\forall a \in A, \forall t \in T : \tau_{a,t} \leq \sum_{s \in S} \lambda_{s,a,t} \cdot r_s \qquad (4)$$

V. RESULTS AND DISCUSSION

This section evaluates the ORCHESTRA framework in several scenarios. First, we will describe the experimental setup. Second, we discuss the impact of using the OVL on inter-technology handovers in a real-life setup. Third, we evaluate the algorithm's performance, in terms of achieved throughput, and scalability. Fourth, we evaluate the impact of using distributed ORCHESTRA controllers to further increase overall scalability.

A. Experimental setup

The prototype consists of multiple parts: first, there is the ORCHESTRA controller running on an Intel NUC. Another Intel NUC is used for the controller for the 5G-empower network that contained two dual-band APs. A third Intel NUC was used as a switch to connect all the different components, to this extend Open vSwitch was installed. This switch was controlled by the Ryu OpenFlow controller. Furthermore, we had two Linux devices with an OVL, which was developed using the Click Modular Router. Some regular smartphones and Macbooks where used as legacy devices. All emulations, (e.g., for the evaluation of the algorithm), where conducted on an Intel NUC aswell.

B. Handover evaluation

For the handover comparison we used the experimental setup, described above. We compare the state of the art implementation, which consists of losing the connection and then trying to reestablish it on another technology, with our OVL implementation. We used a 6 Mbps stream, a bitrate needed for an HD video, over 120 seconds. We initiated a handover every 30 seconds and measured throughput for TCP and UDP and loss for UDP.

Figure 4 shows the results, where we can see clear differences between the state of the art and our solution. It is clearly visible that the state of the art is losing its connection and has to reestablish it to continue sending. After reconnecting the application tries to compensate for the time where it is
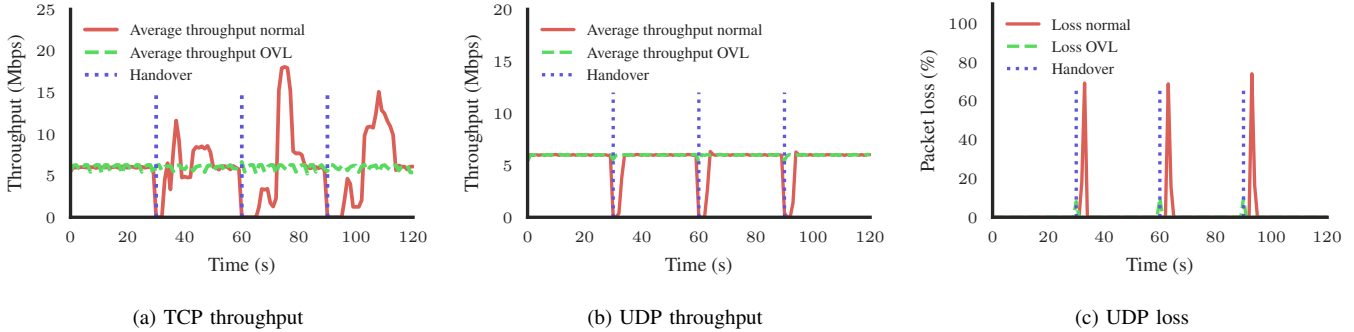
(a) TCP throughput   (b) UDP throughput   (c) UDP loss

Fig. 4. Throughput and loss for 6 Mbps TCP and UDP stream.

(a) Two technologies   (b) Three technologies   (c) Four technologies
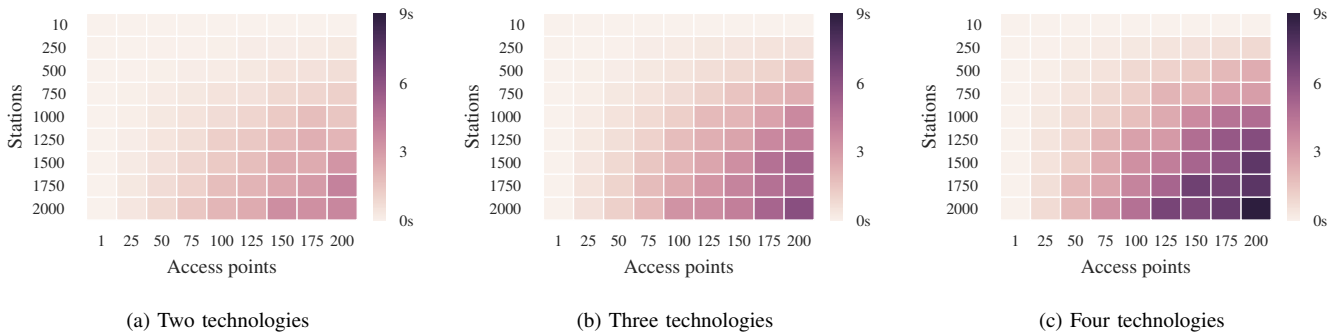
Fig. 5. Heatmap for execution times of LP formulation under increasing number of devices and technologies

not able to send and reaches high data rates. In case of a more congested link, this would not be possible and a video might freeze. The loss behaves in a similar way and increases drastically when a handover happens. ORCHESTRA performs significantly better with no throughput loss during a handover and only minimal packet loss. This means our solution does not need to compensate after a handover and continuously uses the specified data rate without spikes. This also means that for real time traffic, such as a live video, our solution introduces no quality loss and the user experience is consistently high.

C. Load balancing algorithm scalability

The small scale topology and network conditions of our prototype, make it hard to estimate the impact on overall throughput and the feasibility of solving the LP problem in real-time. To this extent, we set up a number of emulations to determine both performance (throughput) and scalability. These emulations take place on a Intel NUC and we artificially provide the necessary inputs to the LP. We varied the number of stations between 10 and 2000 and the number of APs between 1 and 200. Furthermore, we consider four technologies: IEEE 802.11n, IEEE 802.11ac, LTE and, Bluetooth with the following respective capacities (maximum theoretical values): 600 Mbps, 1733,3 Mbps, 1000 Mbps and 3 Mbps. We assume that all APs support all available technologies per scenario, while each client supports a uniform random number of technologies. Moreover, we assign uniformly random amounts

TABLE I
THROUGHPUT COMPARISON BETWEEN LP SOLUTION AND BASELINE

Configuration (Stations/APs)	Baseline throughput (Gbps)	LP throughput (Gbps)
10/1	2.2191 (±0.0653)	2.9098 (±0.1348)
250/25	52.7365 (±0.1809)	70.8027 (±0.5997)
1000/100	212.8500 (±0.7134)	282.3014 (±1.2698)

of traffic to each station, within the interval of 60 % to 140 % times the total network capacity divided by the total number of stations. Hence, each station has a unique amount of traffic, while the overall network is not largely oversaturated.

We first look at the impact on the performance, in terms of throughput, of the LP algorithm, compared to a static baseline that assumes devices connect to a single technology according to the following priorities: IEEE 802.11ac before IEEE 802.11n before LTE. Moreover, for each scenario the results are averaged across 10 different runs and the standard error is reported. The results, shown in Table I, for three different scenarios show, respectively, an increase of 31.1272 %, 34.26 %, 32.63 % in throughput for the LP.

Figure 5 shows the heatmaps for scenarios with respectively two, three and four technologies, for the scalability evaluation. For each individual configuration, a colored cell indicates the average time, needed to solve the LP, across 10 different runs. In the first scenario only IEEE 802.11n and IEEE 802.11ac are present, while in the second and third scenario LTE and
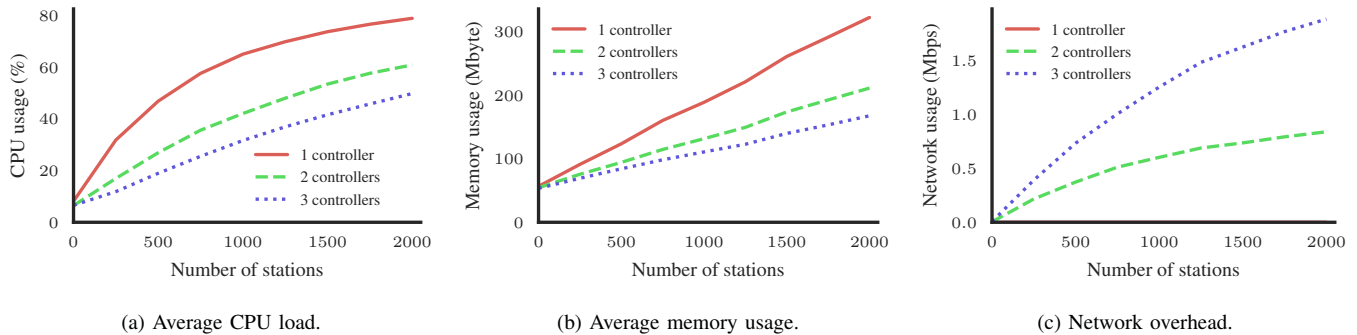
Fig. 6. Increased CPU can be compensated by multiple controllers. The memory footprint is very small and neglectable, which is also the case for the network overhead for the communication between controllers.

Bluetooth are added respectively. The figures clearly indicate that the execution time of the LP increases, as expected, while adding more technologies, stations and, APs, denoted by the evolution towards more darker colors. If we look more closely we see that the optimal solution for most configurations can be calculated under one second, meaning that it is possible to rapidly respond to dynamic network events. For instance, for a configuration with 500 stations and 50 APs the solvetimes ($\pm$ the standard error) are the following: 0.0963 s ($\pm$ 0.0036), 0.1995 s ($\pm$ 0.0105) and, 0.3294 s ($\pm$ 0.0038) for two, three and four technologies respectively. If the number of technologies is fixed at three and we double the number of stations, the average solve time increases to 0.5035 s ($\pm$ 0.0089). If we do the same for the number of APs (number of stations remains 500), we see an increase up to 0.4670 s ($\pm$ 0.0134). It is only when the numbers further increase, that execution times tend to become longer. In the largest case of 4 technologies, 2000 stations and 200 APs, it takes 9.0844 s ($\pm$ 0.4883) to calculate the optimal solution. To keep reaction times fast, control can be distributed, which is evaluated in the next section.

D. Load distribution

In order to improve our solution in large networks, with thousands of devices and a multitude of technologies, we investigate the impact of distributing the computational and management load across different controllers. To this extent, we emulated up to 2000 clients and used up to three controllers to distribute the load. Additionally, we emulated 200 APs that served the clients. Note that the values for the number of stations and access points are identical to the ones used in the scalability evaluation of the LP in the previous section. Furthermore, we average the load for each number of stations and the same hardware as with the previous experiments was used. The results can be seen in Figure 6a where one controller reaches up to 80 % average CPU usage for 2000 clients. This means that there is not much leeway for the assignment computation and either the computation takes longer or the clients are served with a delay which results in a worse user experience, If we have multiple controllers the load is better distributed and the likelihood of a bottleneck when a new client assignment is computed is eliminated. On the other hand,

the distribution overhead regarding computational load is not neglectable. While one controller reaches 80 %, two controller reach 60 % each and three controller reach 40 % each. This means a 50 % overhead which seems to be stable among an increasing number of controllers.

Both memory usage (Figure 6b) and network usage (Figure 6c) are low. While memory usage starts around 100 MB for all configurations, we can see a similar increase in usage for the different amount of controllers. There is again a 50 % overhead for two or more controllers. The network usage for information sharing on the other hand does increase linearly for an increased number of controllers. It tops to around 2.5 Mbps with two controllers and 2000 clients. Compared to the traffic the clients produce, this is very low. All in all, our solution scales very well regarding CPU, memory, and network usage and it is possible to scale it to multiple thousands of clients by distributing control.

VI. Conclusions

In this paper,we presented the ORCHESTRA framework that introduces dynamic management and control in heterogeneous WLANs. The framework consists of two major components: (1) the controller that communicates with all kinds of devices in the network, including legacy clients, and contains the decision-making logic, (2) a virtual MAC layer (OVL) that provides a single socket connection to the higher levels and contains features like load balancing, replication, TDMA scheduling and intra- and inter-technology handovers. In the evaluation part, we show that the implementation of the OVL allows seamless and transparent handovers using a real-life prototype. Furthermore, we have presented an LP algorithm for load balancing clients across multiple technologies and APs. Emulations show that this algorithm scales up to thousands of devices and can provide a significant improvement of approximately 33 % higher throughput in different scenarios. Finally, we show that distributing the ORCHESTRA controller helps to improve scalability even further.

REFERENCES

[1] IEEE Std. 1905.1-2013, "IEEE standard for convergent digital home network for heterogeneous technologies," 2013.

[2] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "Tcp extensions for multipath operation with multiple addresses," Tech. Rep., 2013.

[3] D. Macone, G. Oddi, A. Palo, and V. Suraci, "A dynamic load balancing algorithm for quality of service and mobility management in next generation home networks," *Telecommunication Systems*, vol. 53, no. 3, pp. 265–283, 2013.

[4] G. Oddi, A. Pietrabissa, F. D. Priscoli, and V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks," in *World Telecommunications Congress*, 2014, pp. 1–6.

[5] O. Bouchet, A. Kortebi, and M. Boucher, "Inter-MAC green path selection for heterogeneous networks," in *IEEE Globecom Workshops (GC Wkshps)*, 2012, pp. 487–491.

[6] A. Kortebi and O. Bouchet, "Performance evaluation of inter-mac green path selection protocol," in *12th Annual IEEE Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2013, pp. 42–48.

[7] T. De Schepper, S. Latré, and J. Famaey, "A transparent load balancing algorithm for heterogeneous local area networks," in *International Symposium on Integrated Network Management (IM)*, 2017.

[8] N. Soetens, J. Famaey, M. Verstappen, and S. Latré, "SDN-based management of heterogeneous home networks," in *11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 402–405.

[9] K. Xu, X. Wang, W. Wei, H. Song, and B. Mao, "Toward software defined smart home," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 116–122, 2016.

[10] P. Gallo, K. Kosek-Szott, S. Szott, and I. Tinnirello, "SDN@home: A method for controlling future wireless home networks," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 123–131, 2016.

[11] T. De Schepper, P. Bosch, E. Zeljkovic, K. De Schepper, C. Hawinkel, S. Latré, and J. Famaey, "Sdn-based transparent flow scheduling for heterogeneous wireless lans," 2017.

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[13] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of sdn/openflow controllers," in *Proceedings of the 9th central & eastern european software engineering conference in russia*. ACM, 2013, p. 1.

[14] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined wireless networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[15] M. Zekri, B. Jouaber, and D. Zeghlache, "A review on mobility management and vertical handover solutions over heterogeneous wireless networks," *Computer Communications*, vol. 35, no. 17, pp. 2055–2068, 2012.

[16] G. Gódor, Z. Jakó, Á. Knapp, and S. Imre, "A survey of handover management in lte-based multi-tier femtocell networks: Requirements, challenges and solutions," *Computer Networks*, vol. 76, pp. 17–41, 2015.

[17] J. G. Andrews, S. Singh, Q. Ye, X. Lin, and H. S. Dhillon, "An overview of load balancing in hetnets: Old myths and open problems," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 18–25, 2014.

[18] B. Ng, A. Deng, Y. Qu, and W. K. Seah, "Changeover prediction model for improving handover support in campus area wlan," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 265–272.

[19] S. Fernandes and A. Karmouch, "Vertical mobility management architectures in wireless networks: A comprehensive survey and future directions," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 45–63, 2012.

[20] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "A first analysis of multipath tcp on smartphones," in *International Conference on Passive and Active Network Measurement*. Springer, 2016, pp. 57–69.