ORCHESTRA: Enabling Inter-Technology Network Management in Heterogeneous Wireless Networks

Tom De Schepper*, Patrick Bosch*, Ensar Zeljković*, Farouk Mahfoudhi*, 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*—Modern connected devices are equipped with the ability to connect to the Internet using a variety of different wireless network technologies. Current network management solutions fail to provide a fine-grained, coordinated, and transparent answer to this heterogeneity, while the lower layers of the OSI stack simply ignore it by providing full separation of layers. To address this, we propose the ORCHESTRA framework to manage the different devices in heterogeneous wireless networks and introduce capabilities such as packet-level dynamic and intelligent handovers (both inter- and intra-technology), load balancing, replication, and scheduling. The framework is the first of its kind in providing a fine-grained packet-level control across different technologies by introducing a fully transparent virtual MAC layer and an SDN-like controller with global intelligence. Furthermore, we present a novel optimization problem formulation that can be solved to optimally configure the network. We provide a thorough evaluation through simulations and a prototype implementation. We show that our framework enables, in a real-life setting, transparent and real-time inter-technology handovers and that coordinated load balancing can double the network-wide throughput across different scenarios.**

***Index Terms*—network orchestration, virtual MAC layer, load balancing, heterogeneous wireless networks.**

I. Introduction

Today's networks consist of a plethora of heterogeneous devices that are equipped with the ability to connect to the Internet using a variety of different network technologies (e.g., ZigBee, 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 Long Term Evolution (LTE)-A, 60 GHz Wi-Fi, visible light communications, and Bluetooth 5.0 [1, 2]. 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 [3, 4].

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. Each of these technologies operates independently, isolated from each other and cooperation between them is neither considered nor easily possible with the current design of the lower layers of the OSI stack [5, 6]. 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 varying traffic demands). 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 are missing [7, 8]. Furthermore, a solution like LTE-Wireless Local Area Network (WLAN) Aggregation (LWA), where LTE packets are tunneled via Wi-Fi, is not completely transparent and can not be easily extended to other technologies [9]. All of this indicates the need for intelligent and dynamic inter- and intra-technology routing and interface selection optimizations to aid in unlocking the network's full potential.

In this paper, we further introduce and evaluate our software-defined framework ORCHESTRA that relies on network virtualization to cope with the above described heterogeneous challenges and is able to support inter-technology management [10]. It consists of two major parts: the virtual MAC layer (VMAC) and the ORCHESTRA controller. The VMAC (denoted as the ORCHESTRA Virtual Layer (OVL) in our previous work [10]) unifies the Medium Access Control (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. A Software-Defined Networking (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 VMAC-enabled and legacy devices across the entire network, and decides, amongst others, on technology and path selection, access point (AP) assignment, and channel access. This framework allows for the implementation of the previously mentioned optimizations like intra- and inter-technology handovers, load balancing, duplication, and dynamic path reconfiguration.

The contributions of this paper are threefold. First, we present the ORCHESTRA framework, both the VMAC and controller, into detail. We also discuss the interactions with different networking components and protocols. Second, we present a Mixed Integer Quadratic Programming (MIQP) formulation that can be run on top of the controller, to

optimize the network configuration and increase the overall throughput. Third, we implemented the full framework in a real-life prototype, allowing us to demonstrate features such as handovers, load balancing and replication in a realistic setting.

We extend our previous work in the following ways [10]: first, a more in-depth description of the framework is given. Second we introduce a novel MIQP formulation that also takes into account the mobility of stations and uses a more detailed estimation of the capacity of wireless links. Third, we significantly extended our evaluation of the framework, both in a real-life prototype and through ns-3 simulations to demonstrate the added value of using the ORCHESTRA controller together with the VMACs on the devices. Finally, we also present a more extensive study of the related work.

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 VMAC architecture and functionality in Section III, while the load balancing problem and algorithm are stated in Section IV. Sections V and VI discuss the results achieved with, respectively, the prototype and through simulation. Finally, conclusions and future research directions are provided in Section VII.

II. Related work

In this section we discuss existing work on the two main areas addressed in this paper: inter technology handovers and load balancing in heterogeneous networks. Afterwards, we provide a summary comparying ORCHESTRA to the state of the art.

A. Inter technology handovers and management

The idea of introducing a layer between the existing MAC layers and the network layer, similar to the presented VMAC, has also been proposed in the IEEE 1905.1 standard [7]. Devices that are compliant to the IEEE 1905.1 standard have an abstract layer on top of the current data link layer (i.e., OSI layer 2), which hides the underlying diversity in MAC technologies. A unique virtual MAC address, assigned to each abstract layer, represents the corresponding device on the network. In combination with data link header rules, this makes is possible to transparently switch flows between multiple heterogeneous interfaces. In contrast to our packet-based solution, the IEEE 1905.1 standard only grants flow-level control over the network. Furthermore, it only supports Ethernet, Wi-Fi, Powerline HomePlug, and Multimedia over Coax (MoCA) as underlying technologies. Despite its potential, IEEE 1905 never really took off. Since its release in 2013, it has been under active development, without follow-up releases and no products exist yet that support it.

In strong contrast, MPTCP is being used in industry (e.g., by telecom providers) and consumer devices, especially smartphones (e.g., by Siri in iPhones) [8, 11]. MPTCP is a Transmission Control Protocol (TCP) extension that enables the transmission and reception of data concurrently on multiple network interfaces. Multiple regular TCP connections, denoted as subflows, are combined into a single MPTCP connection, while each subflow can follow a different path. Application data can be divided across these subflows to attain a higher throughput, or duplicated for reliability. The division of data among the different subflows is decided by a scheduler that introduces a form of intelligence that can react to dynamic network characteristics (e.g., increased RTT) [12]. 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.

Besides MPTCP, other multi-technology advances have been made in the area of smartphones and mobile networks. To meet growing demands, the use of both unlicensed spectrum (LTE-LAA/LTE-U) and Wi-Fi technology (LWA) for traffic offload has been proposed [13, 9]. In the first case, LTE technology is directly used in the unlicensed spectrum (especially the 5 GHz band), likely causing severe performance degradation for existing Wi-Fi systems [14, 15]. On the other hand, LWA proposes to combine an LTE Evolved Node B (eNB) with one or more Wi-Fi APs by either a physical integration or an external network interface. While still the 5 GHz band will be more heavily utilized, no hardware changes are required on the infrastructure [16]. From a user perspective, both LTE and Wi-Fi are used seamlessly as mobile traffic flows are tunneled over the Wi-Fi connection. In contrast, ORCHESTRA offers a more transparent approach, as LTE and Wi-Fi are considered equally and no additional tunnels are required.

Finally, it is also worth mentioning that different SDN alternatives for the management of heterogeneous wireless networks have been proposed. One of the best known solutions is the 5G-EmPOWER framework that focuses on virtualized network functions in wireless networks [17]. In line with the thought of Network Function Virtualization (NFV), it moves intelligence from an AP to a controller. Similarly to other approaches it relies on an OpenFlow controller and is thus limited to flow-level control, while also limited functionalities and technologies are supported.

B. Load balancing in heterogeneous Local Area Networks (LANs)

While the previously discussed multi-technology solutions have features to enable dynamic flow redirection, they lack the necessary algorithms and intelligence. To this extent a number of load balancing algorithms have been defined: Macone et al. propose a per-packet load balancing algorithm [18] for home networks that runs centrally on the gateway. However, it assumes full instantaneous knowledge of network resources and conditions. A decentralized load balancing algorithm specifically for heterogeneous wireless access networks was proposed by Oddi et al. [19]. This algorithm relies on a multi-connection transport layer in order to cope with the drawbacks of per-packet load balancing in the case of TCP. The proposed algorithm is based on the Wardrop equilibrium and does not take into account the fact that users do not have dedicated network resources when using wireless technologies. In general, it was shown that determining the actual available bandwidth on the links has a big impact on the results of distributing the flows [20]. Recent load balancing solutions

focus also on energy optimization. Bouchet et al. proposed an algorithm that aims to reduce energy consumption and use the most energy efficient link while still providing a good QoS [21, 22]. This is done by assuming the energy consumption model is known in advance, and not by real-time measurements on the devices.

In the area of 5G networks there has been research on dividing different connections across different technologies. 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) [23]. Decisions are made centrally by the base station and different strategies have been proposed using, among others, utility functions, multiple attributes decision making, Markov chains, game theory, and user location [23, 24]. Additionally, load balancing policies also look at the number of connected devices to a base station. However, these strategies take only a limited number of parameters into account, with Received Signal Strength Indicator (RSSI) and Signal to noise Ratio Signal To Noise Ratio (SNR) being the most popular ones [25]. 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 [26]. Current solutions are technology specific and do not take actual application or QoS parameters and objectives into account.

C. Comparison to ORCHESTRA

To summarize, in contrast to existing solutions, ORCHESTRA offers a more fine-grained control over heterogeneous networks due to the packet-level operations and its coordinated intelligence. The ORCHESTRA framework can also be used in a broader range of networking scenarios with all types of technologies, while offering a holistic way of managing all of them. Furthermore, current research on load balancing in LANs mostly focuses on the development of theoretical models that assume the detailed knowledge of flow throughput requirements and dynamic network conditions. The specific nature of wireless networks (e.g., interference, link quality variability) is often ignored and approaches are technology specific and not suitable for matching QoS requirements. In contrast, the work presented in this paper uses real-time distributed monitoring information, rather than assuming complete knowledge of the network. Moreover, the specific characteristics of wireless networks and the mobility of stations are also considered explicitly.

III. ORCHESTRA ARCHITECTURE

In this section, we define the functionality of our approach, describe the controller's capabilities, and introduce the novel VMAC layer. The latter was renamed from OVL into VMAC, to be more descriptive and clear.

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 VMAC layer 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.

c) Handovers: Moving a client from one AP to another while keeping the current technology (horizontal or handovers) or moving a client from one technology to another (vertical handovers). 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 three 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 occurs.

B. ORCHESTRA virtual MAC layer (VMAC)

Currently connectivity is handled on an interface basis and there is only one active interface. 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.

The VMAC implementation is located above the existing data link layers 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. The virtual layer only uses the depicted functionality to organize the different MAC layers and let it appear to the networking layer that only one interface, the VMAC, is present. As a consequence, only a single, unified IP address is needed, while a single unique virtual MAC address is not required.

The data link layer has no knowledge about the VMAC and behaves like in a normal network stack, forwarding incoming packets upwards. They then pass through the VMAC, are filtered or ordered, and are then passed to the network layer. Similar, an outgoing packet is passed from the network layer to the VMAC where functionality like load balancing or duplication is applied, and is then passed to the correct interface at

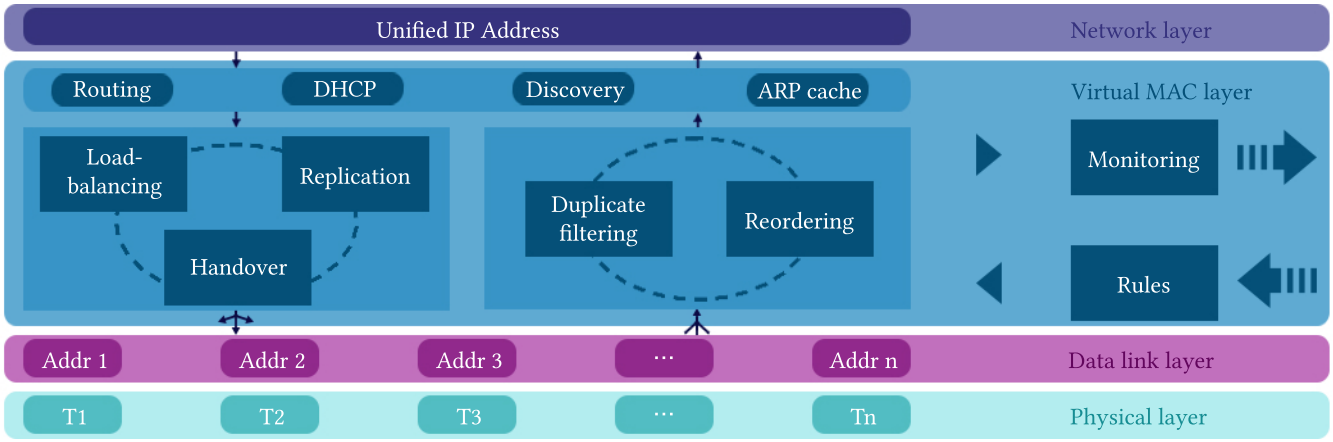


Fig. 1: The ORCHESTRA virtual layer is on top of the data link layer and acts as a data link layer itself to the network layer.

the correct time. Contrary to existing solutions, 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 VMAC 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 IP addresses, port, transport protocol type, sequence numbers and more. They also denote the weights for each interface in the case of load balancing. 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 VMAC 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 an ordered hash map of packets and the last sequence number that was forwarded to the network layer. Additionally, every packet has a timeout after which it is forwarded regardless of other conditions. 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 map if it is the next one. If this is the case, we remove the head and continue checking the head of the map, until a packet is missing. If the arriving packet is not the next one, there are two possibilities. First, the sequence number is lower than the expected one. This means that it is a late packet and it will be forwarded immediately. Second, the sequence number is higher than the expected one. In this case, the packet will be stored in the map at the first position where it has a smaller sequence number than the current packet at that position. Additionally, a timeout for the packet is added.

b) Replication: Replication can be done for specific packets or for all packets of a flow. In this case, a 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) 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 controlled by a SDN/NFV controller or have the VMAC as well. When this is not the case, a client that has a VMAC can still perform a handover and buffer its packets to lower the packet loss. However, the overall duration of the handover and the overall performance depends on the configuration of the APs.

C. ORCHESTRA controller

While the VMAC allows achieving fine-grained MAC control inside a single node, the ORCHESTRA controller enables management and orchestration of the entire network. This is illustrated in Figure 2. 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 2, the ORCHESTRA controller is one step higher in the hierarchy than existing SDN/NFV 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 com-

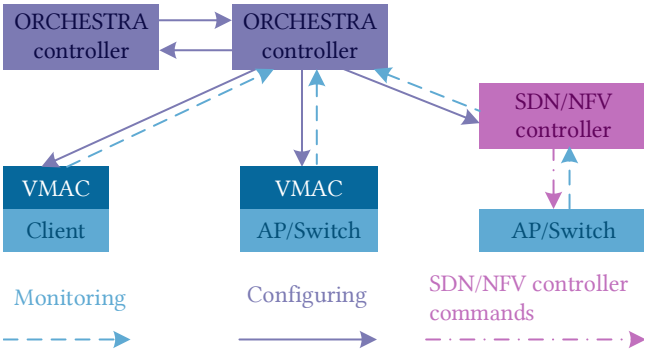


Fig. 2: ORCHESTRA is the center of network management and coordinates existing SDN-based controllers as well as clients through different protocols.

municate 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 2). In detail, we use the underlying API to get 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 VMAC as the end point on the client. Here the controller sends User Datagram Protocol (UDP) packets which the virtual layer intercepts 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, port, or packet type serve as filters whereas the previously defined functionality serves as what is to be applied. Messages from the controller are recognized according to the controller IP and if no controller is known yet 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 message to initiate a handover contains at least one technology (or interface) with the current AP configuration and the new configuration that needs to be installed (i.e., the new AP to connect to). Note that there can also be several configuration combinations as several technologies (i.e., interfaces) can be active at the same time. The ORCHESTRA controller instructs the wireless SDN/NFV controller and the VMAC of the affected AP as well to inform them about the handover and to start synchronizing the handover with the client. The VMAC 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 Time-Division Multiple Access (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

Previously we explained how rules on the VMAC can be used for, amongst others, flow scheduling. While we mentioned how the controller can set these rules, we did not yet discuss the intelligence that is required to decide on a specific configuration. We now present a novel load balancing algorithm that runs on top of the controller and optimizes the overall network throughput. This MIQP formulation differs from our previous work in the fact that we model the network more accurately [10]. For instance, we model individual traffic flows instead of bundling all traffic per station. Furthermore, we also explicitly take into account the shared spectrum of wireless technologies and the mobility of stations.

A. Network model

A network is modeled as a multi-graph defined as a tuple (S,T,B) where:

- S is the set of stations $\{s_1, s_2, ..., s_n\}$. These stations represent the different consumer devices within the LAN such as smartphones, tablets, or laptops.
- T is the set of technologies $\{t_1, t_2, ..., t_n\}$. In this paper we focus on the use of IEEE 802.11n at 2.4 GHz to IEEE 802.11ac on 5 GHz.
- B is the set of all Basic Service Sets (BSSs) $\{b_1, b_2, ..., b_n\}$. A BSS is defined as a set of stations $\{s_1, s_2, ..., s_n\}$ that are connected to an AP using a certain technology. In other words, a BSS encapsulates all the stations that can interfere with each other since they share the capacity of a technology.

Furthermore, we define the following sets and elements to complete the network model:

- $\forall s \in S : T_s$: defines for each station the set of all technologies $t \in T$ that are supported by the station.
- $\forall b \in B : B_t$: is the set of BSS for a certain technology $t \in T$.
- $\forall s \in S : B_s$ set of basic service sets to which $s \in S$ can belong. In other words these are all the BSS of which the AP are in range of the station.
- Finally, we define $d_{s,b}$ and $b_{s,b}$ to be, respectively, the data rate (depending on the MCS) and bit error rate of the station $s \in S$ for a specific BSS $b \in B$.

In addition to the network topology, traffic flows going through the network also need to be modeled. Therefore, we define F as the set of all flows. A flow $f \in F$ is a triple $< s_f, r_f^{in}, r_f^{out} >$ with $s_f \in N$ the station within the network that is the source or destination of the flow within the network, r_f^{in} the incoming desired rate of f $\in \mathbb{R}^+$ and r_f^{out} the outgoing desired rate of f $\in \mathbb{R}^+$. Note that we do assume that the gateway is always one of the two endpoints of the flow, while the other is denoted by s_f. Furthermore, we separate the desired rate of the flow between the incoming and outgoing rate. This allows us to more precisely schedule all flows across the different paths, as incoming and outgoing packets of a flow can be assigned a different route. To clarify, for a TCP flow originating from some web server, the incoming rate is the rate of the data traffic, while the outgoing rate is the one of the ACKs.

In contrast to our previous work [10, 27], this model takes, amongst others, the mobility of stations into account (through $d_{s,b}$ and $b_{s,b}$). The required data is provided by the monitoring functionalities of the ORCHESTRA framework. Currently the measured rssi values per station and the MCS value (assigned by the Minstrel rate adaptation algorithm) are considered.

B. MIQP formulation

The load balancing problem considered in this paper is modeled as an MIQP formulation, which consists of the necessary inputs, decision variables, an objective function, and a set of constraints. This model can be solved (i.e., to find the value of the decision variables, given the constraints and objective function) using a variety of optimization algorithms and heuristics. In this paper, the Gurobi solver is used, which finds the optimal solution to the problem.

The inputs of the presented MIQP consist of the previously described network and flow model. Concretely, there are the following inputs: S, T, B, T_s, B_t, B_s, $d_{s,b}$, $b_{s,b}$, and F.

Additionally, we need one more input: we define χ_b to be a linear function that approximates the capacity of the different BSSs, taking into account the number of stations and the particular technology. In Section IV-C, we discuss this function in detail.

Furthermore, we define the following decision variables:

- $\tau_f^{in} \in \left[0, r_f^{in}\right]$; this variable defines the total incoming rate assigned to a flow $f \in F$.
- $\tau_f^{out} \in \left[0, r_f^{out}\right]$; this variable defines the total outgoing rate assigned to a flow $f \in F$.
- $\lambda_{f,b}^{in} \in \{0, 1\}$; this variable represents the path for the incoming traffic of a flow. If the incoming traffic of flow $f \in F$ is scheduled over BSS $b \in B_{s_f}$ then $\lambda_{f,b}^{in}$ = 1, otherwise it equals 0.
- $\lambda_{f,b}^{out} \in \{0, 1\}$; this variable represents the path for the outgoing traffic of a flow. If the outgoing traffic of flow $f \in F$ is scheduled over BSS $b \in B_{s_f}$ then $\lambda_{f,b}^{out}$ = 1, otherwise it equals 0.
- $\gamma_{s,t,b} \in \{0, 1\}$; this variable represents the connection between a station and an AP. It is equal to 1 if a station $s \in S$ on technology $t \in S_t$ is part of the BSS $b \in B_s \cap B_t$, otherwise it equals 0. In other words, we assume that per technology a station can only be connected to one AP.

As an objective function, the model maximizes the total rate (bandwidth) of the traffic, both incoming and outgoing, across the entire network:

- max $\left(\sum_{f \in F} \tau_f^{in} + \tau_f^{out}\right)$

Finally, we define the following constraints:

- We first define a constraint that the capacity of BSSs and their underlying technologies is not exceeded:
  - $\forall b \in B : \sum_{f \in F} \lambda_{f,b}^{in} \cdot \tau_f^{in} + \lambda_{f,b}^{out} \cdot \tau_f^{out} \leqslant \chi_b$
- Next, we define a constraint that limits the total estimated rate over all traffic flows on a station, going over a certain BSS, by the rate that is supported by the configuration of that station:

- $\forall s \in S, \forall b \in B_s : \sum_{f \in F_s} \lambda^{in}_{f,b} \cdot \tau^{in}_f + \lambda^{out}_{f,b} \cdot \tau^{out}_f \leqslant d_{s_f,b} \cdot b_{s_f,b}$

- Furthermore, we define two constraints that guarantee the conservation of flows in the network (i.e., the right endpoints):
  - $\forall f \in F : \sum_{b \in B_{s_f}} \lambda^{in}_{f,b} = 1$
  - $\forall f \in F : \sum_{b \in B_{s_f}} \lambda^{out}_{f,b} = 1$
- The last group of constraints make sure that a device can be connected to only one AP (or BSS) per technology:
  - $\forall s \in S, \forall t \in T_s : \sum b \in B_s \cap B_t \gamma_{s,t,b} = 1$
  - $\forall s \in S, \forall t \in T_s, \forall b \in B_s \cap B_t, \forall f \in F_s : \lambda^{in}_{f,b} \leqslant \gamma_{s,t,b}$
  - $\forall s \in S, \forall t \in T_s, \forall b \in B_s \cap B_t, \forall f \in F_s : \lambda^{out}_{f,b} \leqslant \gamma_{s,t,b}$

Note that while this problem is formulated as a quadratic problem, because of the two capacity constraints, it can be easily linearized by making use of the well-known Big M method. This also done automatically by modern solvers.

C. Parameter estimation

In this section, we discuss the most important parameters that have an influence on the optimal solution. First of all, it is important to have a correct estimation of the capacity of a wireless link or BSS. Since the actually capacity of wireless connection is depending on a lot of factors, such as the configuration of APs, interference of other devices within or outside the network, and the amount of traffic in the network, it differs significantly from the theoretical capacity [28]. To avoid the dependency on large complex models, we propose the use of an approximation function that estimates the capacity of a BSS depending on the number of stations connected [27]:

$$\chi_b(\alpha, \beta) = \alpha \cdot (\sum_{f \in F} \lambda^{in}_{f,b} + \lambda^{out}_{f,b}) + \beta$$

The parameters α and β are technology specific and can be experimentally determined. Note that this function forms the right hand side of the first constraint.

Furthermore, the rate supported by the configuration of the station and the position of that station can also heavily influence the achievable throughput (cfr. the second constraint). Therefore, we introduced $d_{s,b}$ and $b_{s,b}$, respectively, the data rate (depending on the MCS) and bit error rate of the station $s \in S$ for a specific configuration. The maximum data rate of a station is, amongst others, determined by the MCS value. To take into account the mobility of stations, we propose to make a mapping from measured RSSI values to MCS values, as the latter then leads to a theoretical rate. This mapping will be a stepwise function as multiple rssi values will corresponds to the same MCS. Furthermore, we make use of second linear function to map measured rssi values to packet loss, in order to correct the theoretical achievable data rate. Both functions can be experimentally determined, per unique topology and technology, by taking fingerprints of the recorded MCS and packet loss at different distances (and thus different rssi values) from the AP. This approach is comparable to the well know fingerprinting technique in wireless localization [29].

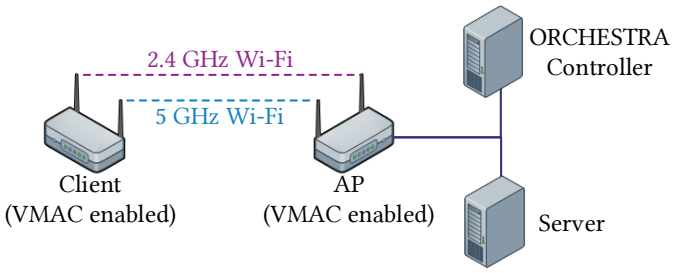


Fig. 3: Architecture of the prototype setup

V. Prototype results and discussion

This section evaluates the main features of the ORCHESTRA framework in a real-life setting. We start by explaining the experimental setup and afterwards we evaluate the performance of vertical handovers, load balancing, and duplication.

A. Prototype setup

The prototype setup consists of four devices: one ORCHESTRA controller, one server, and two devices with the VMAC layer implemented. These two VMAC-enabled devices represent a client and AP, each of them equipped with both a 2.4 GHz and 5 GHz Wi-Fi interface. To support comparison with legacy scenarios, the VMACs on both client and AP can be disabled. The ORCHESTRA controller issues commands and rules to the client and AP, as explained in Section III-C, while the server acts as an endpoint for the traffic flows of the client. Both the ORCHESTRA controller and server are connected by wire to the AP. For all the devices we make use of Intel NUCs with a Ubunutu 16.04 operating system. The experimental setup is depicted in Figure 3.

The VMAC is implemented by using the Click Modular Router [30]. We opted for Click as it allows for fast and high-level prototyping, which is handy for ongoing research. A more finalized solution will be implemented on kernel-level in the future. We make use of existing Click elements for basic packet handling and communication with the different interfaces and layers. The VMAC logic for handling rule matching, reordering, and (de)duplication is implemented in new Click elements.

For each of the evaluated functionalities we investigate the impact on both UDP and TCP traffic (using iperf) and compare our solution to a baseline. Each experiment is repeated ten times and results are averaged. This evaluation is presented in the following subsections.

B. Handover evaluation

The key feature, enabled by the ORCHESTRA framework, is the enabling of instantaneous and transparent multi-technology handovers. To demonstrate this, the following experiment was conducted: the controller issues handover commands every 30 s, for a total of three times. We compare the ORCHESTRA solution to two legacy scenarios: a scenario where the client has a VMAC layer and the AP does not, and a scenario where neither client nor AP are equipped with our solution. As Ubuntu machines are used, we had to

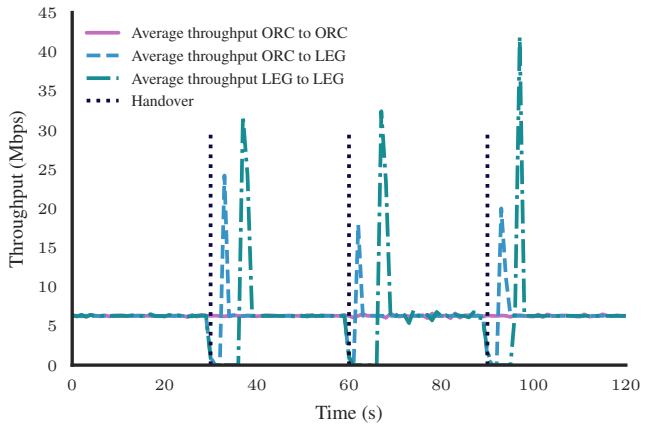


Fig. 4: 6 Mbps TCP stream with handover every 30 seconds. Two ORCHESTRA enabled device have no drop, one has a 2 second drop, and none has a 6 second drop.

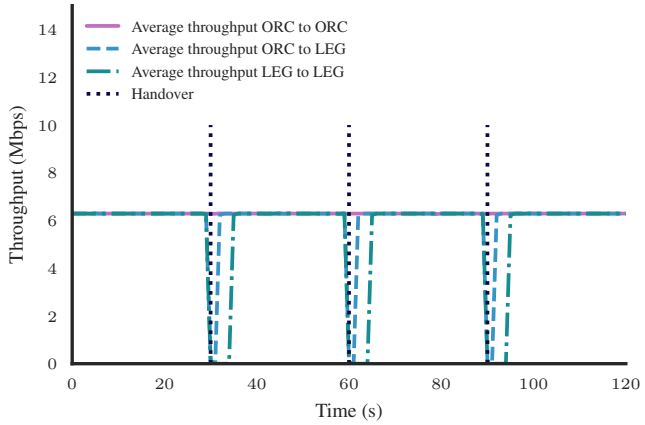


Fig. 5: 6 Mbps UDP stream with handover every 30 seconds. Two ORCHESTRA enabled device have no drop, one has a 2 second drop, and none has a 5 second drop.

simulate the handover of devices without a VMAC layer. By default, Ubuntu reacts very slow, if at all, when a connection is breaking down and multiple technologies are available. For a wireless connection, this can easily take 15 seconds (s) or more, in which case the connection completely drops. Therefore, to have *iperf* not break down, a script monitored the link continuously and if no traffic was detected for four seconds, it switched the route to the correct interface.

Figures 4 and 5 show the resulting graphs, respectively, for TCP and UDP traffic of 6 Mbps. The handovers, every 30 s are depicted by the dashed vertical lines. Note that ORC refers to an ORCHESTRA enabled device, while LEG refers to a legacy device.

For the TCP case, three different throughput patterns can be detected for the three different configurations. First of all, the problems experienced by legacy devices can be clearly seen as traffic drops completely upon a handover, because the underlying connection was lost. As a result of this connection loss for 6 s, iperf with TCP overcompensates after reconnecting by increasing the amount of traffic until the 6 Mbps on average is reached. This can clearly be seen as the throughput heavily increases and is going up to 30 Mbps and higher for a brief spike. In other words, the application or operating system need to handle the connection loss and upon detection, it needs to reestablish the connection.

The second case, consisting of one VMAC and one legacy device, already behaves differently as the VMAC on the client can detect if a connection is going down much faster than the operating system. If it is dropping, it switches easily to another technology and send packets over the new connection. This can be seen in Figure 4 as the drop in throughput is only for 2 s, in contrast to the 6 s with two legacy devices. While improving the downtime, the drop itself is not completely avoidable as the handover is done without informing the AP.

This is in stark contrast to the case with two ORCHESTRA enabled devices, where the loss of connection is mitigated completely and a smooth handover is shown as the throughput is constant during the handover. This heavily improves performance and the stream does not need to overcompensate for the time the connection is down.

For the UDP case, similar behavior is experienced. However, as UDP is more resilient to sudden link failure as it does not require acknowledgements and packets are sent regardless, if a connection exists or not. Therefore, no sudden increases in throughput can be seen. For the configuration with two legacy devices, a connection loss of up to 5 s is shown, while for the second case with only one legacy device there is a drop of 2 s. When both devices are equipped with a VMAC, the throughput drop is completely gone and a smooth handover with continuous performance can be seen.

C. Load balancing evaluation

The packet-level control introduced by the VMAC allows for packet-level load balancing across multiple interfaces. We demonstrate this by an experiment where the controller issues load balancing rules to both the client and AP device. At the start a traffic stream of 6 Mbps is balanced fifty-fifty across both available interfaces. After 30 s this is altered into 30 % of traffic over 2.4 GHz and 70 % over 5 GHz. At the 60 s mark we return to the initial fifty-fifty configuration, before ending up with a 70-30 % for, respectively, 2.4 GHz and 5 GHz.

Figures 6 and 7 show the resulting graphs, respectively, for TCP and UDP traffic. In both figures the average throughput for both interfaces individually is shown, as well as the total achieved throughput at the server for both interfaces combined.

From the results for both TCP and UDP it is clear that the packet-based load balancing works as intended, as the traffic is in both cased distributed across both interfaces according to the set weights. While the UDP stream has a stable throughput across both interfaces, as shown in Figure 7, this is not the case for the TCP stream, shown in Figure 6. The TCP stream does on average achieve the desired throughput of 6 Mbps, but the throughput fluctuates around this average value. This is caused by the reordering of packets that is necessary because TCP requires a correctly ordered stream. Due to the varying characteristics (e.g., latency or interference)

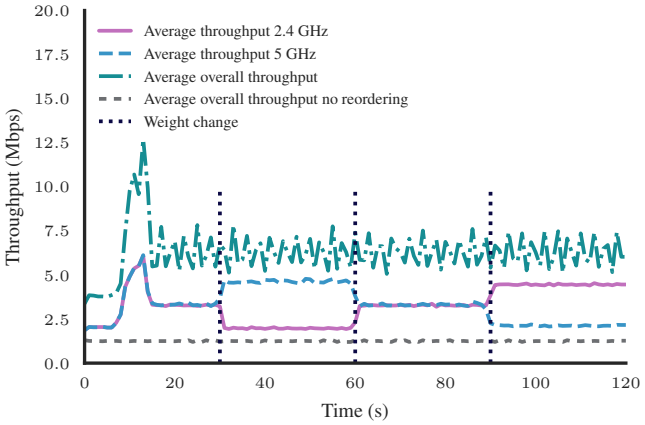


Fig. 6: 6 Mbps TCP stream load-balanced over two technologies with a weight change from 50/50 (2.4 GHz/5 GHz) to 30/70 to 50/50 to 70/30.

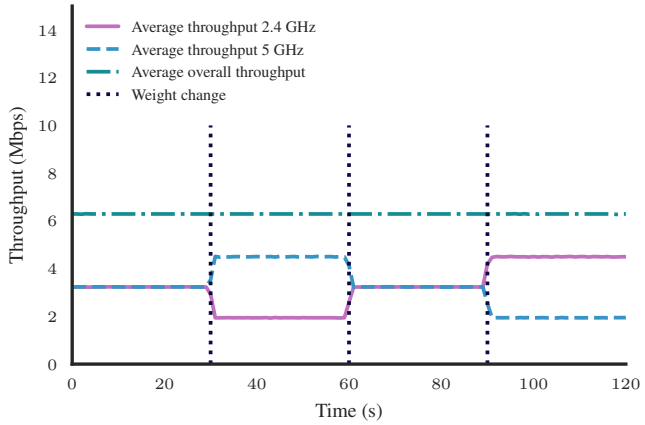


Fig. 7: 6 Mbps UDP stream load-balanced over two technologies with a weight change from 50/50 (2.4 GHz/5 GHz) to 30/70 to 50/50 to 70/30.

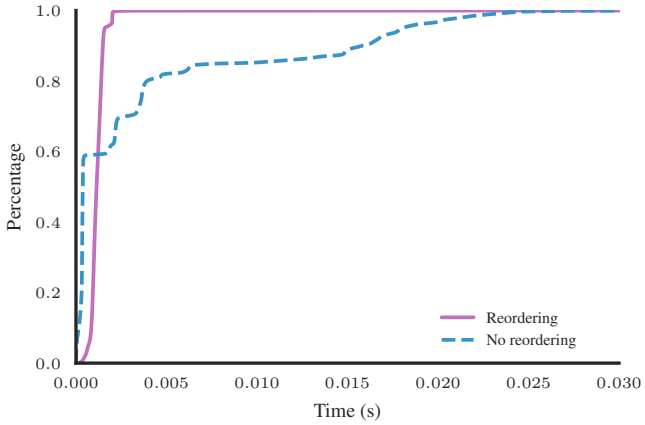


Fig. 8: Cumulative plot of packet jitter of 6 Mbps TCP stream load-balanced over two technologies.

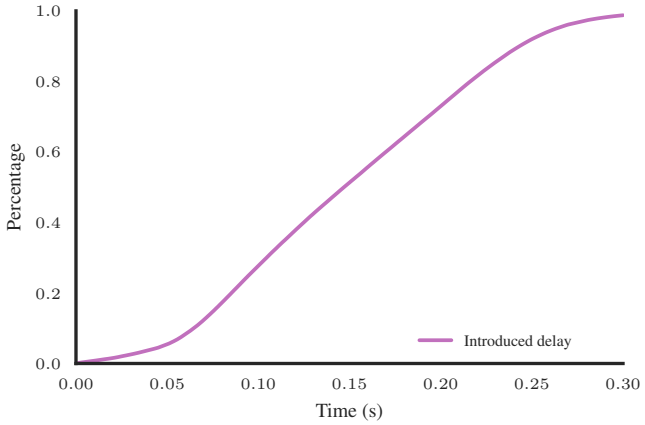


Fig. 9: Cumulative plot of introduced delay by reordering packets of a 6 Mbps TCP stream load-balanced over two technologies.

in the two different frequency bands, load-balanced packets are not received in order. The need for packet reordering is clearly shown in Figure 6 where the TCP stream only achieves a throughput of 1.3 Mbps when reordering is disabled. Furthermore, the small throughput fluctuations are explained by the fact that out-of-order packets are placed in a hash map, until the missing packets have been received, as such the rate in which packets are forwarded to the upper layers is varying. This can lead to reaction of the TCP congestion control algorithms.

To evaluate the impact of the addition of packet reordering for a TCP stream we looked at the delay that is added to received packets. Figure 8 shows the cumulative packet jitter for the TCP stream, while Figure 9 shows the cumulative delay that is being added because of the packet reordering. The difference in throughput is enabled by a lower jitter in the case of reordering. It is clear though that the increased throughput comes at the cost of increased delay. The impact of and possible improvements to the TCP behavior under packet-reordering is an interesting and challenging topic that we will further investigate in future work.

D. Duplication evaluation

Similar to the load-balancing, the control offered by the VMAC on a packet level, allows for the duplication of packets. As packets can be transmitted across multiple interfaces, the reliability is improved, as changes of receiving a certain packet increases. This is shown in the following experiment, where packets of a traffic flow with a rate of 6 Mbps are transmitted across both the 2.4 GHz and 5 GHz Wi-Fi interfaces. At the receiver side, duplicate packets need to be filtered out before pushing them to the upper layers, as explained in Section III-B. In order to emulate an unreliable environment we implemented a drop probability of 20 % per packet for each of the interfaces. As both connections are unreliable, there are a relatively high amount of lost packets and thus timeouts in the de-duplication step (when a packet it dropped at both interfaces). Figures 10 and 11 show the results for, respectively, TCP and UDP Similarly, to the resulting graphs of the load-balancing

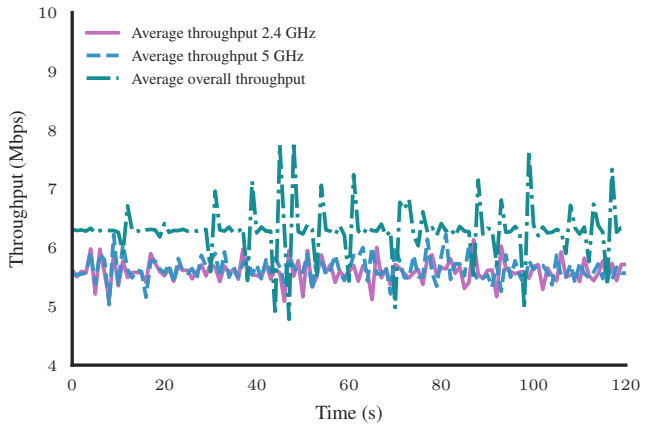


Fig. 10: 6 Mbps TCP stream with duplication over two technologies and a configured drop of 20 % per link. Full throughput is still achieved.

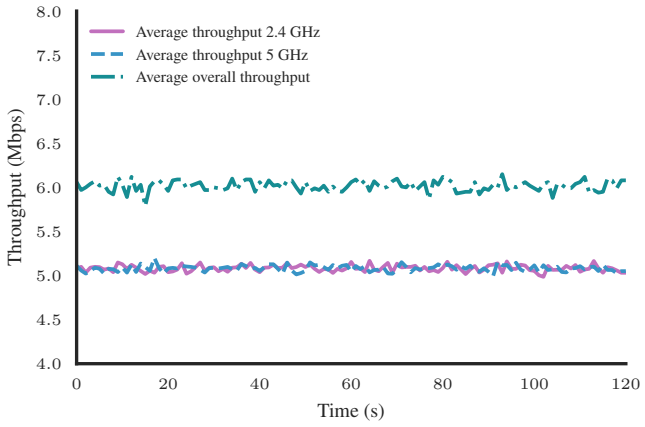


Fig. 11: 6 Mbps UDP stream with duplication over two technologies and a configured drop of 20 % per link. Full throughput is still achieved.

experiments, both the overall average throughput across both interfaces, as well as the average throughput per individual connection are shown.

For both the TCP and UDP scenario the desired throughput of 6 Mbps is still achieved, despite the inflicted packet loss. This is the clearest visible for the UDP case, depicted in Figure 11, as there the throughput is less fluctuating (compared to the TCP graph in Figure 10) as it is less affected from loss due to the absence of a retransmission scheme. As there is a packet-drop probability of 20 % on each interface, the average throughput on both the 2.4 GHz and 5 GHz Wi-Fi interfaces is, as expected, roughly 20 % lower then the desired 6 Mbps. In contrast to the very stable throughput of the UDP stream in the load balancing experiment (cfr. Figure 7), here some small fluctuations (both up and downwards) can be seen. The reason for this lays in the fact that packets are not always received at the exact same rate at the upper layers, because of drops.

For the TCP stream, higher fluctuations can be seen in Figure 10 as it is affected differently by the packet loss. As TCP has a retransmission scheme, when a packet is not received, it does need to request it again to the sender. This behavior briefly hampers throughput (indicated by the slight drops), before being corrected afterwards (indicated by the high upwards spikes). Despite this fluctuations, for both the TCP and UDP traffic stream, the desired rate is met. This indicates that ORCHESTRA indeed offers increased reliability when desired.

VI. Simulation results and discussion

This section evaluates the proposed load balancing algorithm using simulation results obtained from the ns-3 event-based network simulator [31]. In this section we illustrate the advantage of combining the intelligence on the centralized controller, having a global view over the network, with the VMAC on the devices in the network. First, the evaluation setup and scenario are discussed. Second, the framework's performance, in terms of achieved throughput and execution time, is evaluated in a variety of static and dynamic scenarios. For every scenario, we provide a comparison to a fully distributed baseline where each device is equipped with a VMAC, but decides for itself to which AP to connect, based on the best RSSI values. Furthermore, we also compare the novel formulation to our previous work [10].

A. Evaluation setup

The Gurobi Optimizer (7.5.2) is used to solve the MIQP formulation, while simulations were done in ns-3.26 while using a single core of an Intel® Xeon® E5-2680 Processor running at 2.8 GHz and with 8 GB RAM. In the simulator, we implemented the entire ORCHESTRA framework, as described in Section IV. We assume two technologies present: 5 GHz Wi-Fi and 2.4 GHz Wi-Fi. Every scenario has at least two APs that support both technologies. For both types of Wi-Fi, the IEEE 802.11n standard with mode GI $= 400$ ns (short guard interval) is used. For the 5 GHz Wi-Fi a 40 MHz channel is assumed, in contrast to the 20 MHz channel for the 2.4 GHz Wi-Fi technology. This allows for a theoretical data rate of respectively 150 Mbps and 72.2 Mbps. Note that because Wi-Fi roaming and channel scanning are not implemented in the ns-3.26 version, only one single channel is used per technology, across all APs. While this does limit the maximum achievable data rate, it does not influence the evaluation of the presented algorithm, as the environment is identical for both the baseline and the algorithm. Furthermore, dynamic rate adaption is made possible through the Minstrel rate adaption algorithm, which is in real-life used by Linux devices.

In order to generate representative network topologies and conditions, several types of consumer devices are defined, each with different types of interfaces, mobility and flows. All of this information is depicted in Table I. The exact number of devices, the assigned flow type, and the rate of the flow are chosen uniformly at random between an upper and lower bound, based on the involved device and depending on the scenario. Each mobile device moves according to the Random Waypoint Model within a certain area, with a random start position and a uniformly random chosen speed between

TABLE I: Overview of the devices used in the scenarios, including their supported network technologies and flow rates

Device type	Mobile	Technologies	Rate boundaries per flow type		
			Download	Video stream	Video conference
Laptop	Yes	2.4 & 5 GHz	10–30 Mbps	8–20 Mbps	4–10 Mbps
HD Television	No	2.4 & 5 GHz	5–25 Mbps	10–20 Mbps	5–10 Mbps
4K Television	No	2.4 & 5 GHz	5–25 Mbps	15–25 Mbps	7.5–12.5 Mbps
Tablet	Yes	2.4 & 5 GHz	1–8 Mbps	2.4–9 Mbps	1.2–4.5 Mbps
Smartphone modern	Yes	2.4 & 5 GHz	1–8 Mbps	2.4–9 Mbps	1.2–4.5 Mbps
Smartphone old	Yes	2.4 GHz	1–8 Mbps	2.4–9 Mbps	1.2–4.5 Mbps

0.3-0.7 $\frac{m}{s}$. The size of the area and the wait times at the waypoints are depending on the scenario. Moreover, within the static scenarios the flow rates do not change over time, while in the other scenarios the download flows will consume as much bandwidth as possible (reflecting their actual behavior). Assuming a static flow rate for the first part of the evaluations, allows us to better estimate the impact of only the mobility aspect. The size of the downloaded file is uniformly at random chosen between 10 MB and 10 GB. We assign one flow per device and as such do not assume the concurrent usage of both Wi-Fi interfaces, as this is generally not supported by current hardware. Note, that the flow rates were selected based on representative figures from literature of existing applications in these three categories [32]. We decided to use only TCP traffic flows, as current Internet traffic is dominated by TCP [33].

For every described scenario, results are averaged over different randomly generated flow and topology configurations. To this extent, each experiment was repeated 20 times. We also report the standard error for each experiment over these 20 repetitions. As a baseline for comparison, a fully distributed approach is used where all devices have a VMAC but decide for themselves to which AP to connect based on RSSI values. When the RSSI of the current connection drops below a threshold of -75, the devices check if a better connection is possible to another AP. If this is the case, a handover is performed and traffic is rerouted over the best active connection. The execution of the algorithm is trigged by the real-time monitoring component when dynamic changes to the network have been detected (e.g., a variation in one of the flow rates of at least 25 %) or when it has been 10 s since the last execution. To avoid oscillations in the decision making, there should be at least 2 s between two consecutive executions. Finally, the following parameters were experimentally determined for the function χ_b in the algorithm: for α and β respectively, for 2.4 GHz Wi-Fi -1.74 and 57.58, and for 5 GHz Wi-Fi -3.21 and 112.99.

B. Home and office scenarios

To demonstrate the impact of the algorithm in LANs, two basic scenarios were created to reflect two typical representative environments: a home and an office network. The home scenario consists of ten devices: 2 APs, 2 laptops, 3 smartphones (modern), 2 tablets, and 1 4K Television. For the office scenario, we double the number of devices to twenty: 4 APs, 9 laptops, 5 smartphones (modern), 1 tablet, and 1 HD Television. The area in which the mobile devices move is, respectively, 20 on 10[m] and 30 on 15[m], while the wait times at the waypoints are uniformly randomly chosen between 5-15 s in both scenarios. Both scenarios show a strong improvement in the total throughput compared to both the distributed baseline and our previous work, as depicted in Figures 12 and 13.

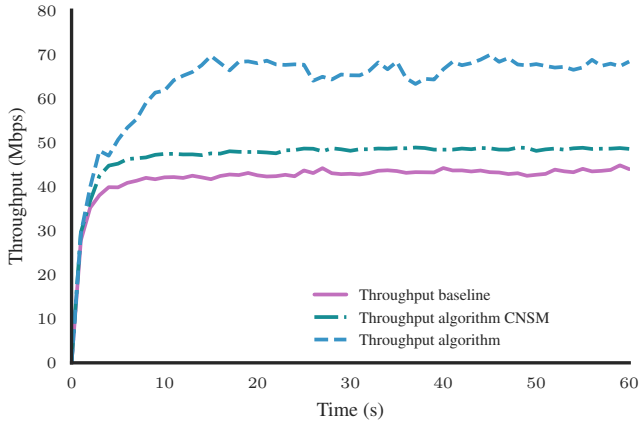


Fig. 12: Throughput as a function of time for a home scenario

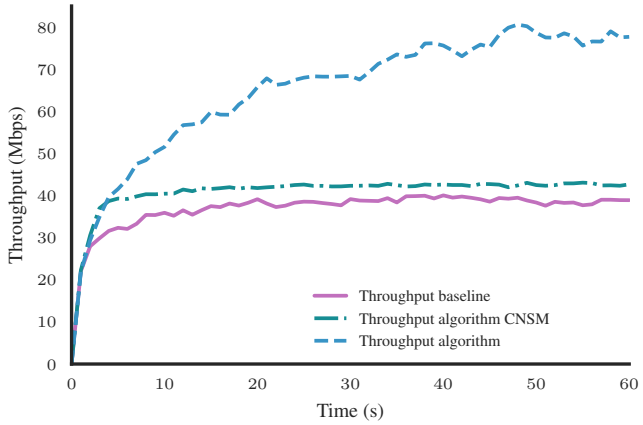


Fig. 13: Throughput as a function of time for an office scenario.

For the home scenario, the average throughput is 43.07 Mbps (±0.15) for the baseline and 48.28 Mbps (±0.08) for the CNSM algorithm. In contrast, the average throughput with the novel MIQP-based algorithm increases to 66.89 Mbps (±1.22). This means that there is an increase of 55.31 % and 38.55 % towards, respectively, the baseline and the previous algorithm. Similarly, for the office scenario there is an increase of 83.69 % in average throughput from 38.39 Mbps (±0.37) to 70.52 Mbps (±2.23) for the baseline. Compared

TABLE II: Impact of mobility on throughput

	Wait times	Baseline	Algorithm
Home	0-10 s	56.46 Mbps (±2.57)	65.68 Mbps (±1.26)
	5-15 s	43.07 Mbps (±0.15)	66.89 Mbps (±1.22)
	10-20 s	43.57 Mbps (±0.27)	65.32 Mbps (±1.27)
Office	0-10 s	64.98 Mbps (±3.16)	69.39 Mbps (±2.17)
	5-15 s	38.39 Mbps (±0.37)	70.52 Mbps (±2.23)
	10-20 s	38.23 Mbps (±0.40)	67.75 Mbps (±2.02)

to the CNSM algorithm, there is an improvement of 66.95 %. Taking into account the wireless network characteristics and the mobility of stations in the formulation of the algorithm, clearly allows for a more tailored network configuration and thus an increased throughput in comparison to our previous work. In both figures, especially in the office scenario, the curve at the left for the algorithm's throughput, indicates that different iterations of the algorithm each results in a better configuration. This can be explained by the fact that the algorithm only knows the measured rate of each flow and not the actual desired rate. So every new configuration that is calculated by the algorithm, allows certain flows to reach a higher rate, which is then picked-up up by the next execution of the algorithm. Other fluctuations in the throughput can be explained by the mobility of some of the devices and by the reaching the maximum capacity of the Wi-Fi. Furthermore, the solve times for the MIQP formulation by the Gurobi solver are on average, for the home and office scenario respectively, 0.10 s (±0.03) and 44.70 s (±861.22). We will elaborate on these solve times and the scalability of the algorithm in the next section.

Finally, we considered the impact of mobility on the overall throughput. Therefore, we varied the waypoint wait times for both scenarios by additional experiments for times between 0-10 s and 10-20 s. The results, listed in Table II, show that the algorithm always outperforms largely the baseline. However, for the case with the highest mobility (and lowest wait times) the baseline performs significantly better, then in the other cases. We believe this to be due the higher number of handovers, triggered by the mobility.

C. Impact of network load and scalability

To investigate the scalability of the algorithm in terms of traffic and execution time, the following scenario was created: a set of devices is randomly generated, each with a uniform randomly assigned flow with a randomly chosen type and rate. The total desired rate of all flows equals a certain percentage of total theoretical network capacity. Experiments were performed for loads of 10, 20, 30, 40, and 50 % of the theoretical network capacity. Moreover, the presence of 3 APs was assumed in a space of 20 on 15 m with a waypoint wait time of 5-15 s. Figure 14 illustrates that the algorithm offers a significant increase in throughput under all loads. In particular, as the network load increases, so does the relative performance of our proposed framework compared to the baseline: at a load of 10 % the baseline achieves a throughput of 41.96 Mbps (±0.66), the CNSM algorithm achieves 48.09 Mbps (±0.41) while with the algorithm this increases to 50.30 Mbps (±0.07). For a network load of 50 % the throughput is, respectively for

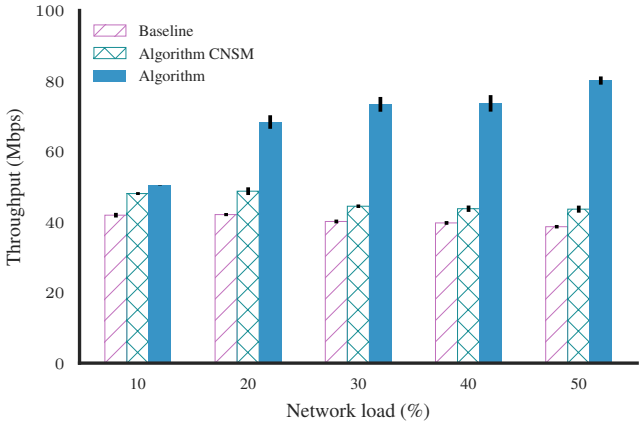


Fig. 14: Throughput as a function of network load, error bars depict the standard error

TABLE III: Comparison of the execution time of the novel algorithm to our previous work, under increasing network load

Load	Flows	Execution time CNSM	Execution time
10	6	1.58×10^{-3} s ($\pm 2.56 \times 10^{-4}$)	0.12 s (±0.05)
20	9	2.48×10^{-3} s ($\pm 4.58 \times 10^{-4}$)	0.39 s (± 0.56)
30	14	3.26×10^{-3} s ($\pm 5.78 \times 10^{-4}$)	9.98 s (±137.56)
40	18	3.89×10^{-3} s ($\pm 6.27 \times 10^{-4}$)	29.49 s (± 179.88)
50	23	4.5×10^{-3} s ($\pm 7.58 \times 10^{-4}$)	37.27 s (± 376.39)

baseline, previous work, and algorithm, 38.69 Mbps (±0.49), 43.67 Mbps (±0.99), and 80.14 Mbps (±1.16). This is an increase of, respectively, 107.13 % and 83.51 %. Note that the throughput of the baseline and the CNSM algorithm slightly decreases over the different experiments, because devices and traffic are being added, while the capacity has been reached.

Furthermore, the time it takes to solve the MIQP and find the optimal configuration, was also measured. Due to the exponential time complexity for solving MIQP formulations, there is a strong increase in execution time with increasing network loads, as depicted in Table III. While it takes for a network load of 10 % only 0.12 s to calculate the optimal configuration, this increases to 37.27 s for a load of 50 %. This large calculation time is in stark contrast with the scalability of the CNSM algorithm that scaled up to 2000 flows, while the solve time did not exceed 10 s [10]. This large difference is explained by the heavily increased complexity of the problem and formulation (increased number of decision variables and constraints) by taking into account, amongst others, the mobility of stations. Therefore, in future research we will look into a heuristic that finds a near-optimal solution for the centralized MIQP in linear times.

D. Dynamic scenarios

So far we have only considered scenarios with static flow rates and arrival times, as we assumed the flows to be present throughout the entire simulation. While this helped in determining the impact of the mobility aspect, this is not always realistic. Furthermore, an important performance metric of the framework is its adaptability to dynamic conditions. To

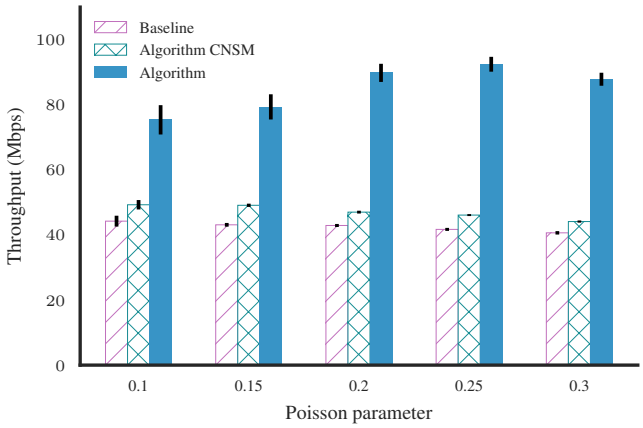


Fig. 15: Throughput as a function of poisson parameters, error bars depict the standard error

evaluate this aspect, we consider such a dynamic scenario in this section. All download flows act as in reality and consume as much bandwidth as possible (or assigned to them by TCP flow control), until the desired amount of data has been downloaded (or the maximum flow length has been reached). Moreover, flows arrive according to a Poisson distribution and the flow length is uniformly at random chosen between the following bounds: 10 and 30 s. Furthermore, the impact of different values for the arrival rate (Poisson parameter λ) is evaluated.

Figure 15 shows the results for this dynamic scenario. For all different arrival rates the framework outperforms the baseline significantly. The improvement is for all parameter values significant and for the first 4 parameter values, the improvement increases with the increase of the Poisson parameter. For 0.1 as parameter value, the baseline achieves a throughput of 44.24 Mbps ($\pm$1.67), the CNSM formulation achieves 49.27 Mbps ($\pm$1.45), while with the algorithm 75.37 Mbps ($\pm$4.51) is achieved. For 0.25 as parameter value, the baseline achieves a throughput of 41.70 Mbps ($\pm$0.46) and the CNSM formulation allows for a throughput of 46.10 Mbps $\pm$0.30). In contrast, the algorithm allows for a throughput of 92.44 Mbps ($\pm$2.28). This is a gain of, respectively, 121.68 % and 100.52 %. Similarly, to the experiments with varying network loads, the throughput of the baseline and algorithms slightly decreases over the different experiments because the total capacity has been reached. For the novel algorithm this is only the case for the last parameter value of 0.3 Finally, we have also repeated this experiment for different flow lengths, but the results were nearly identical and are therefore omitted.

VII. Conclusions

To cope with the heterogeneity in the WLANs of today and tomorrow, we proposed the ORCHESTRA framework in this article. The framework introduces dynamic management and control and 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 (VMAC) that provides a single socket connection to the higher layers and enables features like load balancing, duplication, and intra- and inter-technology handovers. In contrast to our previous work we provide a more in-depth description of the framework and a novel MIQP formulation is introduced that also takes into account the mobility of stations. Furthermore, we developed a more complete implementation of the ORCHESTRA framework, allowing us to evaluate all the proposed features in detail. In a real-life setting we demonstrate that the proposed solution allows for real-time inter-technology handovers without any packet loss and that packet-based load balancing across multiple interfaces where different weights can be enforced. Additionally, we show that the duplication functionality on a packet-level allows to achieve the desired flow rates for both TCP and UDP streams, within an unreliable environment with packet loss. Finally, we also evaluated the impact of running a network-wide load balancing algorithm on the centralized controller. We show that the presented multi-technology load balancing algorithm yields an increase in network-wide throughput of up to 100 %, across a variety of scenarios.

Acknowledgement

Patrick Bosch is funded by FWO, a fund for fundamental scientific research, and the Flemish Government under grant 1S56616N.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021," *Cisco*, pp. 2016–2021, 2016.

[2] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 130–137, 2016.

[3] O. Galinina, A. Pyattaev, S. Andreev, M. Dohler, and Y. Koucheryavy, "5G multi-RAT LTE-WiFi ultra-dense small cells: Performance dynamics, architecture, and trends," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1224–1240, 2015.

[4] M. Chen, A. Li, W. Liu, and J. Hong, "A Multi-Wireless Bandwidth Aggregation Mechanism in SDN Networks," *Open Electrical & Electronic Engineering Journal*, vol. 9, no. 1, pp. 321–327, 2015.

[5] R. Wang, H. Hu, and X. Yang, "Potentials and challenges of C-RAN supporting multi-RATs toward 5G mobile networks," *IEEE Access*, vol. 2, pp. 1200–1208, 2014.

[6] H. Zimmermann, "OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.

[7] P. Line, C. Standards, and C. Society, *IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies IEEE Communications Society*, 2013, no. April.

[8] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple

Addresses," Tech. Rep., 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6824

[9] C. Hoymann, D. Astely, M. Stattin, G. Wikström, J. F. T. Cheng, A. Höglund, M. Frenne, R. Blasco, J. Huschke, and F. Gunnarsson, "LTE release 14 outlook," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 44–49, 2016.

[10] E. Zeljkovi, T. De Schepper, P. Bosch, I. Vermeulen, J. Haxhibeqiri, J. Hoebeke, J. Famaey, and S. Latre, "ORCHESTRA : Virtualized and Programmable Orchestration of Heterogeneous WLANs," in *2017 13th International Conference on Network and Service Management (CNSM)*, 2017, pp. 1–9.

[11] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "A first analysis of multipath TCP on smartphones," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9631, no. September 2015, pp. 57–69, 2016.

[12] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop - CSWS '14*, 2014, pp. 27–32.

[13] A. Mukherjee, J. F. Cheng, S. Falahati, H. Koorapaty, D. H. Kang, R. Karaki, L. Falconetti, and D. Larsson, "Licensed-Assisted Access LTE: Coexistence with IEEE 802.11 and the evolution toward 5G," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 50–57, 2016.

[14] A. M. Cavalcante, E. Almeida, R. D. Vieira, F. Chaves, R. C. Paiva, F. Abinader, S. Choudhury, E. Tuomaala, and K. Doppler, "Performance evaluation of LTE and Wi-Fi coexistence in unlicensed bands," *IEEE Vehicular Technology Conference*, no. April 2015, 2013.

[15] F. M. Abinader, E. P. Almeida, F. S. Chaves, A. M. Cavalcante, R. D. Vieira, R. C. Paiva, A. M. Sobrinho, S. Choudhury, E. Tuomaala, K. Doppler, and V. A. Sousa, "Enabling the coexistence of LTE and Wi-Fi in unlicensed bands," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 54–61, 2014.

[16] N. Zhang, S. Zhang, S. Wu, J. Ren, J. W. Mark, and X. Shen, "Beyond coexistence: Traffic steering in LTE networks with unlicensed bands," *IEEE Wireless Communications*, vol. 23, no. 6, pp. 40–46, 2016.

[17] 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.

[18] 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.

[19] G. Oddi, A. Pietrabissa, F. Delli Priscoli, and V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks," *World Telecommunications Congress 2014 (WTC2014)*, no. February 2015, pp. 1–6, 2014.

[20] O. Olvera-Irigoyen, A. Kortebi, and L. Toutain, "Available Bandwidth Probing for Path Selection in Heterogeneous Home Networks," in *Globecom Workshops (GC Wkshps)*. IEEE, 2012, pp. 492—-497.

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

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

[23] 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.

[24] 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.

[25] B. Ng, A. Deng, Y. Qu, and W. K. Seah, "Changeover prediction model for improving handover support in campus area WLAN," *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, no. Noms, pp. 265–272, 2016.

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

[27] T. De Schepper, S. Latre, and J. Famaey, "Flow Management and Load Balancing in Dynamic Heterogeneous LANs," *IEEE Transactions on Network and Service Management*, 2018.

[28] ——, "A transparent load balancing algorithm for heterogeneous Local Area Networks," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 160–168.

[29] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[30] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.

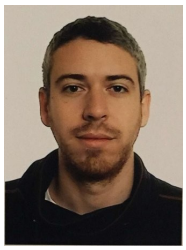
[31] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[32] A. T. NoteTN2224, "Best Practices for Creating and Deploying HTTP Live Streaming Media for the iPhone and iPad," Tech. Rep., 2012.

[33] D. J. Lee, B. E. Carpenter, and N. Brownlee, "Media Streaming Observations: Trends in UDP to TCP Ratio," *International Journal on Advances in Systems and Measurements*, vol. 3, no. 3, pp. 147–162, 2010.

Tom De Schepper is a PhD researcher associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from the University of Antwerp, Belgium in 2015. His Master thesis centered around combining wireless localization techniques with state-of-the-art virtual reality technologies. His current research mainly focuses on providing optimizations and management solutions in multi-technology networks to increase the Quality of Service and the combined usage of multiple (future) wireless technologies. His research resulted in 10 articles published in international peer-reviewed journals and conference proceedings, as well as in a submitted patent application.

Patrick Bosch is a PhD researcher associated with imec and the University of Antwerp, Belgium. He received his Diploma degree in Computer Science from the University of Stuttgart, Germany in 2014. His Diploma thesis centered around optimizing content routing in an SDN based publish/subscribe system. His current research focuses on network orchestration and management of different wireless technologies to improve Quality of Service as well as interference modeling for wireless networks. His research resulted in 6 articles published in international peer-reviewed journals and conference proceedings, as well as in two submitted patent applications.



Ensar Zeljković is currently pursuing a PhD in the Computer sciences. He is a PhD student at the University of Antwerp and the imec research center in Belgium. He holds a Master's degree in Electrical engineering, department of telecommunications from the Faculty of Electrical engineering, University of Sarajevo, Bosnia & Herzegovina. He is currently doing research on Virtual slicing based management of heterogeneous wireless networks.

Farouk Mahfoudhi is an Embedded Software Engineer graduated from the Higher School of Communications of Tunisia in 2012 in Telecommunication Systems Architecture. For more than six years, he has been working in different multinational industrial companies and research centers in Tunisia and Belgium. He has proven skills in embedded software development, network programming, IoT, Set Top Box and drivers development. In 2017, he joined the University of Antwerp and the imec research center as a software developer.

Jetmir Haxhibeqiri received the bachelor's degree in telecommunication from the University of Prishtina, Prishtina, Kosovo, in 2010, and the masters degree in engineering (information technology and computer engineering) from RWTH Aachen University, Aachen, Germany, in 2013. He is currently pursuing the Ph.D. degree at the Internet and Data Laboratory research group (IDLab), Ghent University, Ghent, Belgium. In 2013, he did an internship at Deutche Telekom (T-Laboratories), Darmstadt, Germany, where he was involved in the field of system level simulators for LTE-A. His research interests include wireless communications technologies (IEEE 802.11, IEEE 802.15.4e, LoRa) and their application.

Jeroen Hoebeke is a professor in the Internet Technology and Data Science Lab of Ghent University and imec. He is conducting research on wireless communication solutions for the Internet of Things. On top, he investigates how open standards can be used to roll out connected devices and easily integrate them in IoT applications. He has been active in several IoT domains such as smart building, logistics, health care, industry 4.0, etc. and is author or co-author of more than 100 publications in international journals or conference proceedings.

Jeroen Famaey is an assistant professor associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from Ghent University, Belgium in 2007 and a Ph.D. in Computer Science Engineering from the same university in 2012. He is co-author of over 90 articles published in international peer-reviewed journals and conference proceedings, and 10 submitted patent applications. His research focuses on performance modeling and optimization of wireless networks, with a specific interest in low-power, dense and heterogeneous networks. His research resulted in 6 best paper awards and he was the recipient of the 2015 IEEE Communications Society Outstanding Reviewer award. He is a senior member of the IEEE.

Steven Latré is a professor at the University of Antwerp and director at the imec research institute. He is leading the Internet & Data Lab in Antwerp, a research group which focuses on wireless communications and distributed intelligence. He received a Master of Science degree in computer science from Ghent University, Belgium and a Ph.D. in Computer Science Engineering from the same university. His personal research interests are on wireless network management and multi-agent deep learning. He is author or co-author of more than 100 papers published in international journals or in the proceedings of international conferences. He is the recipient of the IEEE COMSOC award for best Ph.D. in network and service management 2012, the IEEE NOMS Young Professional award 2014, the COMSOC Young Professional award 2015 and is a member of the Young Academy Belgium.