Journal of Network and Systems Management
https://doi.org/10.1007/s10922-019-09502-2

Scalable Load Balancing and Flow Management in Dynamic Heterogeneous Wireless Networks

Tom De Schepper[1] · **Steven Latré**[1] · **Jeroen Famaey**[1]

Received: 23 December 2018 / Revised: 25 May 2019 / Accepted: 11 June 2019

Abstract
The number of connected devices has reached 18 billion in 2017 and this will nearly double by 2022, while also new wireless communication technologies become available. Since these modern devices support the use of multiple communication technologies, efforts have been made to enable simultaneous usage and handovers between the different technologies for these devices. However, existing solutions are missing the intelligence to decide on fine-grained (e.g. flow or packet level) optimizations that can drastically enhance the network's performance (e.g., throughput) and user experience. To this extent, we present a multi-technology flow-management load balancing approach for heterogeneous wireless networks that dynamically re-routes traffic through heterogeneous networks, in order to maximize the global throughput. This dynamic approach can be deployed on top of existing solutions and takes into account the specific characteristics of the different technologies, as well as station mobility. We both present a mathematical problem formulation and a heuristic that ensures practical scalability. We demonstrate the heuristic's ability to increase the network-wide throughput by more than 100% across a variety of scenarios and scalability up to 10,000 devices.

Keywords Real-time wireless network management · Network optimization · Multi-technology load balancing · Inter-technology handovers

✉ Tom De Schepper
tom.deschepper@uantwerpen.be

Steven Latré
steven.latre@uantwerpen.be

Jeroen Famaey
jeroen.famaey@uantwerpen.be

[1] IDLab - Department of Mathematics and Computer Science, University of Antwerp - imec, Sint-Pietersvliet 7, 2000 Antwerp, Belgium

Published online: 18 June 2019

1 Introduction

Over the years, we have witnessed a tremendous increase in the utilization and availability of wireless networks and devices. The number of connected devices has reached 18 billion in 2017 and will further grow to 28.5 billion in 2022 [1]. Similarly, the heterogeneity and complexity of (wireless) networks are increasing as new technologies (e.g. IEEE 802.11ay and 802.11ax) are being released [2]. As a consequence, the management burden increases as each of these wireless communication technology has its own unique characteristics (e.g., capacity and range). Furthermore, these networks are typically being managed statically as, for instance, no centralized intelligence is present and connections are being established based on default priorities (e.g., connecting to the closest infrastructure device by default). This makes them unable to automatically react in a timely fashion to temporary disruptions that cause Quality of Service (QoS) degradations. Figure 1 illustrates this envisioned coexistence of different types of heterogeneous wireless networks, each with its own application domain, technologies, and devices. At the left top side of the figure we see a Local Area Network (LAN) environment, consisting of multiple access points (APs) offering a number of different wireless technologies (e.g., directional 60 GHz communication, a sub-1 GHz low power technology, and more standard 2.4 GHz and 5 GHz IEEE 802.11 (Wi-Fi). Similarly, at the top right side of Fig. 1 we show an Industry 4.0 use case where a combination of Internet of Things (IoT) technologies and Wi-Fi is used. Furthermore, we see a number of Vehicular AdHoc Networks (VANETs) (using IEEE 802.11p or LTE-Vehicular (LTE-V)) that connect both different vehicles and vehicles with road-side units, while requiring reliable real-time communication.

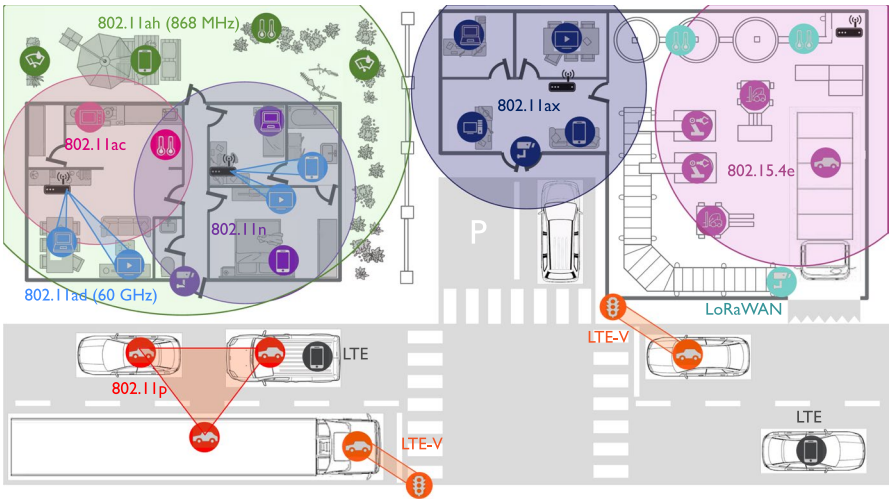

Fig. 1 Example of different possible heterogeneous wireless networks (Home network, Industry 4.0 network, and Vehicular networks) consisting out a variety of technologies (indicated by different colors) and a multitude of devices

As both modern connected devices and wireless networks are equipped with multiple communication technologies, efforts have been made to allow devices to simultaneously use different communication technologies or to switch in real-time between them. This dynamic network and traffic management allows for network optimizations such as multipath routing, load balancing, and dynamic path reconfiguration. This stands in stark contrast to traditional approaches that typically delegate this to the application layer, or even worse, to the user. The most important dynamic multi-technology frameworks and standards that have been proposed are IEEE 1905.1 [3], Multipath Transmission Control Protocol (MPTCP) [4], LTE-Wireless Local Area Network Aggregation (LWA) [5], and ORCHESTRA [6]. IEEE 1905.1 allows to dynamically re-route flows across different interfaces, while MPTCP allows to split a Transmission Control Protocol (TCP) flow across different paths through the network. Furthermore, LWA allows to offload traffic between an LTE base station and Wi-Fi APs, while ORCHESTRA offers a transparent management solution by introducing a Virtual MAC (VMAC), arching different technologies per device, and a centralized controller. While these frameworks and standards introduce the features needed to enable dynamic flow rerouting and load balancing, they are missing the intelligence to decide on the network-wide optimizations.

To this extent, we propose a multi-technology load balancing approach that can balance devices across different APs and divide traffic flows over different paths (and thus technologies) through the network. The approach can be run on top of any of the aforementioned frameworks or standards and makes use of their management functions to dynamically and in real-time handover devices or reroute traffic in order to configure the network to achieve maximum global throughput. In contrast to existing load balancing approaches, we do not assume full knowledge over the network and use real-time monitoring information from the frameworks used under the hood. Furthermore, the approach is completely technology independent and we lay the focus on practical usability in a multitude of scenarios.

In our previous work we have mainly focused on flow management in specific heterogeneous LANs and providing a framework to enable seamless multi-technology management (e.g., inter-technology handovers) [7]. The scenarios considered only stationary devices, a single AP and limited technologies. Furthermore, we introduced a mathematical problem formulation for load balancing in heterogeneous wireless networks but this approach lacked practical applicability [8]. In this paper, we extend these works in several ways. First, we further shift the focus to the more challenging environment of wireless networks and take into account the presence of multiple APs (or base stations) and the mobility of stations. Second, we present a heuristic approach to ensure practical usability and scalability, in large scale environments as well. Third, we focus on optimal parameter selection and extensive evaluations across different scenarios.

The contributions of this paper are threefold: first, we introduce a mathematical model of the load balancing problem in Sect. 3, for both devices and traffic flows, in heterogeneous wireless networks. The problem is formulated as a Mixed Integer Quadratic Program (MIQP), which can be solved using existing linear programming approaches. Second, as solving the mathematical model does not scale well, we present a greedy heuristic algorithm that still takes into account the specifics of the

wireless networks. This is done in Sect. 4. Third, in Sect. 5, we evaluate the resulting model and the heuristic in a variety of scenarios, using different network configurations, based on NS-3 simulations. We show, among others, that the increase in network-wide throughput by the heuristic algorithm is in the same range as provided by the MIQP. Furthermore, these contributions are accompanied by an overview of related work in Sect. 2, while conclusions are provided in Sect. 6.

2 Related Work

In this section, we discuss existing work on both the topic of multi-technology network management and load balancing in heterogeneous network environments.

2.1 Multi-technology Standards and Frameworks

The first attempt towards a multi-technology management framework is made by the IEEE 1905.1 standard [3]. This standard introduces an abstract layer on top of the current data link layer (i.e., OSI layer 2) that hides the underlying diversity in Medium Access Control (MAC) technologies (Ethernet, Wi-Fi, Powerline HomePlug, and Multimedia over Coax (MoCA)). Compliant devices are assigned a unique virtual MAC address, representing the corresponding device on the network. Data link header rules make it possible to transparently switch flows between multiple heterogeneous interfaces. Despite its potential, IEEE 1905 never really took off, without follow-up releases and only a few products that support it.

In stark contrast, MPTCP on the other hand, is currently being widely used by, among others, telecommunication operators to split traffic across both wired and wireless backbone networks (called hybrid access networks). This is, in particular, the case for Digital Subscriber Line (DSL) and LTE solutions, to circumvent the limited capacity of DSL wires (also known as DSL-LTE bonding) [9]. Furthermore, MPTCP is also actively being used on a large scale in Android and iOS devices (e.g., by Siri) [10]. MPTCP is a TCP extension that enables the transmission and reception of data concurrently over multiple network interfaces in order to maximize resource usage and increase redundancy in multi-technology networks [4, 11]. Multiple regular TCP connections (denoted as subflows), are offered as one to the application layer, while under the hood each subflow can follow different paths through the network [4]. Based on the ever-changing network characteristics (e.g., increased RTT), the MPTCP scheduler can divide or duplicate application data across these sub-flows [12]. However, this scheduling is done per connection between two hosts and not a network-wide scale. Furthermore, MPTCP has proven to be very aggressive towards other (regular) TCP connections in the network, sometimes even without added benefits for the MPTCP users [13].

In order to cope with the ever-growing bandwidth and traffic speed demands, especially towards the highly hyped 5G networks, the 3GPP community began exploring the wireless spectrum outside of the traditional licensed 3G/4G bands. Two approaches have been proposed to offload traffic: the direct usage of LTE in

the unlicensed spectrum (i.e., LTE-LAA/LTE-U) and the combined usage of LTE in the licensed and Wi-Fi technology in the unlicensed spectrum (i.e., LWA) [5, 14]. While the first can cause severe performance degradations in coexisting Wi-Fi systems (especially in the 5 GHz band), the LWA approach clearly introduces fewer coexistence issues and no hardware changes are required on the infrastructure [15, 16]. From a user perspective, both LTE and Wi-Fi are used seamlessly as mobile traffic flows are tunneled over the Wi-Fi connection. Some commercially available products for LAN exist that use a similar tunneling approach between a so-called pro-active router and a cloud instance, while under the hood different technologies (e.g., DSL, fiber, satellite or LTE) are hidden away (e.g., Mushroom Networks [17]).

Recently, the ORCHESTRA framework has been proposed as the first solution that can be used transparently with all technologies and communication protocols [6]. The framework consists of two parts: a VMAC on the devices and a centralized or cloud-based controller. The VMAC unifies the underlying heterogeneous technologies per device, offering a single interface to the upper layers with a single IP address. Based on packet matching rules, the VMAC forwards packets to the designated underlying technologies. This allows for fine-grained features such as packet-level load balancing, vertical handovers, and duplication of traffic flows for reliability. The controller gathers real-time monitoring information from the different VMACs and can, in turn, send commands to update the rules on specific VMACs in order to optimize the network.

Summarized, different solutions have been proposed that allow for multi-technology management and features (e.g., handovers or duplication). In complement, there is a need for algorithms and intelligence (as the approach presented in this work) that use these frameworks and standards to optimize the network. This is offered by the proposed load balancing algorithm that can be used on top of the listed solutions. We discuss in Sect. 3.5 the deployment of our approach on top of the ORCHESTRA framework.

2.2 Load Balancing in Heterogeneous Networks

Multi-technology load balancing has mainly been investigated in two research domains: LANs and Wide Area Networks (WANs) (4G/5G). A per-packet load balancing algorithm for LANs was proposed by Macone et al. [18]. The algorithm runs centralized on the gateway and assumes full instantaneous knowledge of network resources and conditions. Furthermore, Oddi et al. [19] introduced a decentralized load balancing algorithm specifically for heterogeneous wireless access networks based on the Wardrop equilibrium. However, it does not take into account the fact that users do not have dedicated network resources when using wireless technologies. In general, Olevera–Irigoyen have shown that determining the actual available bandwidth on the links has a big impact on the results of load balancing the flows in a (wireless) network, in particular with time-varying capacity Wi-Fi and Power line communication (PLC) links [20]. Recent load balancing solutions for LANs focus also on energy optimization by, for instance, selecting the most energy efficient link while still providing a good QoS [21, 22]. However, this is done by

assuming the energy consumption model is known in advance, and not by real-time measurements.

In WANs, most research proposes technology-specific solutions that are capable of load balancing across only two technologies (particularly, LTE and Wi-Fi or Wi-Fi and WiMAX) [23]. Typically, load balancing policies are based on the number of connected devices to a base station. Furthermore, a number of decision strategies have been proposed, using, among others, utility functions, multiple attributes decision making, Markov chains, and game theory [23, 24]. However, these strategies take only a limited number of parameters into account, with Received Signal Strength Indicator (RSSI) and Signal To Noise Ratio (SNR) being (by far) the most popular ones [25, 26]. A large number of open issues remain, including, but not limited to, 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 [25]. More recently, Harutyunyan et al. introduce an Integer Linear Programming (ILP) formulation for traffic-aware balancing devices across LTE and Wi-Fi infrastructures [27]. Moreover, in light of the proposed New Radio principle for 5G networks interest has grown in handover and load balancing approaches for millimeter-wave communications [28, 29]. For instance, a user association scheme based on mixed integer nonlinear programming has been proposed [30]. However, further research and optimizations are needed within this specific area [28].

Summarized, most existing work on load balancing in heterogeneous networks makes use of theoretical models that assume, unrealistic, full knowledge over the detailed state of the network. Furthermore, the specific nature of wireless networks is ignored and approaches are technology dependent. Opposed to existing work, we propose a technology independent approach that focuses on wireless networks (taking into account the specifics), while using only real-time monitored and carefully estimated information.

3 Multi-technology Load Balancing Problem Formulation

In this section, we first introduce a model for heterogeneous wireless networks and present a MIQP representing the load balancing problem. Afterwards, we discuss in detail the interaction with the network and how certain parameters can be determined.

3.1 Problem Definition and Motivation

In Fig. 1, we illustrated already how different wireless networks are omnipresent, each with its own application domain, technologies, and devices. In order to increase the QoS within these networks, support future technologies and account for rising traffic demands or user expectations, intelligent management approaches are needed. In the previous section, we have listed different solutions that enable network features such as inter-technology handovers or load balancing, needed to perform the

necessary optimizations in the wireless networks. The coordination among all the devices in the network and the use of real-time monitoring information are essential to account for the ever-changing wireless context. Recently, the Software-Defined Networking (SDN) paradigm has found its way to these wireless networks to facilitate, among others, station mobility. However, while solutions like ORCHESTRA or 5G-EmPOWER enable the management features, they do not contain the intelligence to decide on the needed optimizations.

To this extent, we envision an intelligent solution that can be deployed on top of the aforementioned frameworks. The approach will use the real-time monitoring information provided by the underlying framework to decide on a better network-wide configuration. After calculating an improved network configuration, the communication and management features (e.g., seamless handovers) of the underlying framework to roll-out the configuration. Our approach focuses on the multi-technology load balancing of stations across different infrastructure devices and of traffic flows across the different available connections and network paths.

Various use cases can benefit from the presented load balancing approach. A straightforward application domain is LANs. For instance, in our homes, we are nearly continuously connected to the internet and consuming online services like live video streaming, Voice over IP (VoIP) calls, and multiplayer gaming. Similarly, in an office scenario, services like multi-person teleconferencing or Virtual Reality (VR) prototyping are being consumed. In this context, our load balancing approach can be used to divide the traffic optimally and thus increasing the bandwidth that is available per flow, as such allowing the services to offer the highest quality and meet the demands of the users. Furthermore, as more and more devices are being connected to the Internet, also the backhaul networks need to be able to support this increasing number of devices and, consequently, the growing demands in traffic. A scalable load balancing approach can come to the rescue by balancing the connected devices and their traffic across different technologies and infrastructure units (e.g., base stations of APs). Other use cases can, among others, be found in the areas of smart cities of connected vehicles.

3.2 Network Model

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

- S is the set of stations $\{s_1, s_2, \ldots, s_n\}$. These stations represent all kinds of connected devices, depending on the modeled network (e.g., smartphones, sensors, vehicles).
- T is the set of technologies $\{t_1, t_2, \ldots, t_n\}$. This can, for instance, be Wi-Fi (e.g., IEEE 802.11ac, IEEE 802.11ad, …) or LTE.
- B is the set of all Basic Service Sets (BSSs) $\{b_1, b_2, \ldots, b_n\}$. A BSS is defined as a set of stations $\{s_1, s_2, \ldots, s_n\}$ that are connected, over a specific technology, to an AP, a LTE base station, or an equivalent infrastructure device. In other words, a BSS encapsulates all the stations that can contend with each other since they

share the capacity of a technology. We assume no interference between BSS that are in the range of each other (i.e., use of different channels).

Furthermore, we define the following sets and elements:

- $\forall s \in S : T_s$: defines per station the set of all technologies $t \in T$ that are supported by that particular station.
- $\forall b \in B : B_t$: is the set of all BSS that offer a certain technology $t \in T$.
- $\forall s \in S : B_s$: set of all BSS to which a station $s \in S$ can belong. In other words, these are all the BSS of which the AP is in range of the station (for a supported technology).
- Finally, we define $d_{s,b}$ and $b_{s,b}$ to be, respectively, the data rate (depending on the Modulation and Coding Scheme (MCS)) and bit error rate of the station $s \in S$ for a specific BSS $b \in B$. These values depend on the mobility and position of stations, with respect to each BSS, and can change heavily over time. We discuss the estimation of $d_{s,b}$ and $b_{s,b}$ later on (in Sect. 3.4).

In addition to the network topology, we also need to model traffic flows going through the network. 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 $s_f \in S$ 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 $\mathrm{f} \in \mathbb{R}^+$ and r_f^{out} the outgoing desired rate of $\mathrm{f} \in \mathbb{R}^+$. 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 rates. 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 the case of a User Datagram Protocol (UDP) flow origination from the same web server, the outgoing rate will be 0 as there are no ACKs.

3.3 MIQP Formulation

The load balancing problem considered in this paper is modeled as a MIQP, which consists of the necessary inputs, decision variables, an objective function, and a set of constraints. The inputs of the presented MIQP consist of the previously described network and flow model. 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 [7]. We discuss this further in Sect. 3.4.

Next, we define the following decision variables:

- $\tau_f^{in} \in \left[0, r_f^{in}\right]$; the total incoming rate assigned to a flow $f \in F$.
- $\tau_f^{out} \in \left[0, r_f^{out}\right]$; the total outgoing rate assigned to a flow $f \in F$.

- $\lambda^{in}_{f,b} \in \{0,1\}$; 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 $\lambda^{in}_{f,b} = 1$, otherwise it equals 0.
- $\lambda^{out}_{f,b} \in \{0,1\}$; 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 $\lambda^{out}_{f,b} = 1$, otherwise it equals 0.
- $\gamma_{s,t,b} \in \{0,1\}$; 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 or base station.
- $\delta \in [0,1]$: the maximal load over all BSS.

As an objective function, the model maximizes the total rate (bandwidth) of the network-wide traffic, both incoming and outgoing, while minimizing the relative maximal load over all BSS:

$$max\left(\omega \cdot \left(\sum_{f \in F} \tau^{in}_f + \tau^{out}_f\right) + (1-\omega) \cdot (-\delta) \cdot \left(\sum_{b \in B} \chi_b\right)\right)$$

This objective function consists of two weighted subfunctions that need to be optimized (with the relative weight between them denoted by ω). The first subfunction represents the total assigned rate across all flows (which needs to be maximized). The second part represents the division of load across all available BSSs. The idea is to minimize the maximal relative load, denoted by δ, across all BSSs [31]. As many mathematical solvers do not allow the usage of maximization or minimization functions within the objective function, δ is bounded by the final constraint. Note that the multiplication of δ with $\sum_{b \in B} \chi_b$ is only needed for normalization. While the goal is to maximize network-wide throughput, the load balancing objective is necessary to spread all connected devices over APs and technologies. This limits the probability that the BSS becomes overloaded when a new device joins the network and connects to that BSS.

We complete the MIQP formulation by defining several constraints. We first define a constraint that limits the total rate over all traffic flows on a station, going over a certain BSS, by the maximal rate supported by the configuration of that station:

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

Next, we define two constraints that guarantee the conservation of flows in the network (i.e., the right endpoints):

- $\forall f \in F : \sum_{b \in B_{s_f}} \lambda^{in}_{f,b} = 1$
- $\forall f \in F : \sum_{b \in B_{s_f}} \lambda^{out}_{f,b} = 1$

Furthermore, we also need to make sure that a station can be connected to only one BSS per technology (this corresponds to reality where a device is in general only equipped with a single radio per technology):

- $\forall s \in S, \forall t \in T_s : \sum b \in B_s \cap B_t \gamma_{s,t,b} = 1$
- $\forall s \in S, \forall t \in T_s, \forall b \in B_s \cap B_t, \forall f \in F_s : \lambda^{in}_{f,b} \leqslant \gamma_{s,t,b}$
- $\forall s \in S, \forall t \in T_s, \forall b \in B_s \cap B_t, \forall f \in F_s : \lambda^{out}_{f,b} \leqslant \gamma_{s,t,b}$

Finally, we define the constraint that bounds the value of δ for balancing the load across BSSs, while also making sure that the capacity of the BSSs and their underlying technologies is not exceeded:

- $\forall b \in B : \sum_{f \in F} \lambda^{in}_{f,b} \cdot \tau^{in}_{f} + \lambda^{out}_{f,b} \cdot \tau^{out}_{f} \leqslant \delta \cdot \chi_b$

3.4 Parameter Estimation

In the next Section, we explain how monitoring information is acquired from the underlying framework and fed into the MIQP to calculate the optimal configuration. While some of the gathered monitoring information, like station and traffic information, can be used directly without the need for further processing, some other parameters and information are also required. A key element for determining an optimal configuration is to have an accurate overview of the available bandwidth per BSS. The big impact of determining the actual available bandwidth of wireless links on the results of load balancing approaches has been shown in literature [20, 32]. The actual bandwidth of wireless technologies depends on several parameters such as the theoretical physical bandwidth, the configuration of APs, the interference of other devices within or outside the network, and the number of traffic in the network. Estimating each of these parameters is very challenging and is in some cases a separate research problem on its own (e.g., interference modeling). In order to avoid the use of complex and resource intensive theoretical models, we make use of the approximation function χ_b to estimate the capacity of the wireless technologies [7]. For each BSS $b \in B$, we define χ_b as follows:

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

The parameters α and β are technology depending and capture the specifics of the wireless network under consideration. The following dynamic experimental method can be used to determine these parameters [7]: a series of experiments can be conducted per technology where the number of stations and the flow rates are varied within a predefined range of values. For instance, the number of stations can be varied from 1 to 10. Similarly, the flow rates can be varied between the theoretical data rate of that technology to a relatively low value, depending on the number of stations and thus flows present. When the achieved data rates are stored, for example, per number of stations present, the average values can afterwards be interpolated

(using a linear trend line) as a function of the number of stations, leading to the approximation function defined above. This method can be applied for each individual heterogeneous network to capture the specific characteristics of that particular environment. As characteristics of the wireless environment change over time, this method can be rapidly re-executed if needed [7].

Furthermore, the MIQP also requires the data rate (depending on the MCS) and bit error rate of the station $s \in S$ for a specific configuration, respectively denoted by $d_{s,b}$ and $b_{s,b}$. For the two parameters a mapping can be constructed: in case of the first parameter this is a mapping from measured RSSI values to MCS values (and theoretical data rate). For the second parameter $b_{s,b}$, a linear function can map the measured RSSI values to packet loss, in order to correct the theoretical achievable data rate. Both mappings can be experimentally determined by using the well-known fingerprinting approach to record MCS and packet loss values at different distances (and thus different RSSI values) in the network environment [33]. These mappings can be re-created to adjust for dynamic changes to the network environment.

3.5 Deployment and Interaction with Underlying Framework

In Sect. 2.1 we listed a number of existing multi-technology frameworks and standards that offer features such as handovers and dynamic flow-rerouting in order to optimize the network configuration and performance. While our load balancing approach can be deployed on top of all listed frameworks, we choose to deploy this onto ORCHESTRA for several reasons. First, it offers the centralized control and monitoring features needed as inputs of our load balancing approach and to roll-out the calculated optimal configuration. Second, it offers the option to split traffic flows on a packet-level across different paths in real-time and in a fully transparent manner through the VMAC functionalities. This is in strong contrast to, for instance, MPTCP that only works between two endpoints and does not offer centralized control and monitoring information.

Through the framework we interact with the network and its devices in the following ways: in a regular interval, all VMACs send monitoring information to the controller that keeps the most recent information stored. For each flow, the following information is stored: the number of transmitted and received packets, the number of transmitted and received bytes, the source, the destination, and the type. Furthermore, per link information such as the number of packet errors, the number of transmitted and received packets, MAC throughput, link availability and the theoretical physical rate is reported. Finally, also information regarding the wireless technologies is stored, like the RSSI values for all APs that are in range per station, for a specific technology. The necessary information to calculate the optimal configuration (e.g., flow rates, flow destinations, and available BSSs per station) is gathered from the stored monitoring information and passed to the MIQP. In turn, after the MIQP has calculated the optimal configuration, we translate this configuration from the MIQP variables to specific per-device VMAC rules. Finally, the controller of ORCHESTRA will handle the transmission of the updated rules to each device and the configuration is thus rolled-out.

The question that remains is when exactly we have to run the load balancing algorithm. This clearly depends on the dynamic characteristics of the network and its environment as in a very static scenario it would only be a waste of resources when the algorithm is running almost continuously. But in contrast, this could be the right thing to do in a very dynamic scenario. An example of such a highly dynamic environment could be the VANETs depicted in Fig. 1. The topology and devices in such networks are highly volatile depending on the number of cars passing by while requiring reliable real-time communication. In order to have an approach that can be utilized across a multitude of scenarios and networks, we propose to trigger the execution of the algorithm when dynamic changes to the network are detected in the monitoring information. This could, for instance, be a variation in one of the flow rates of at least x%. Furthermore, a timeout can be added to ensure the execution of the algorithm on a regular base, when no such dynamic events would occur. Note that while this repetitive execution allows reacting to dynamic behavior such as station mobility or changed traffic demands, this also requires a limited execution time of the algorithm.

4 Heuristic Approach

Optimally solving the MIQP model scales exponentially in terms of the number of devices and flows in the network. As such, heuristic solutions are needed for larger scenarios. To this extent, we propose a heuristic approach in this section.

When solving the multi-technology load balancing problem addressed in this paper, it is necessary to balance both stations across available APs (or base stations) and flows across different paths (i.e., technologies). Both are clearly linked together as flows can only be scheduled across established paths or connections. However, it could be that the capacity of the technologies of the current connections is not sufficient to schedule all the flows from a single station. This would mean that new connections need to be established to less occupied APs or base stations, if possible. The previously introduced MIQP performs the station and flow load balancing jointly while finding the network configuration with the highest possible overall throughput. In order to reduce the complexity and computation time, we propose a heuristic approach that consists of two steps. First, we load balance stations across the available infrastructure devices and resources. Second, we route the flows across the different available paths, established in the first step. We make use of the same inputs as the MIQP formulation (defined in Sect. 3.2).

Algorithm 1 First step: Station Association

1: **for** $s \in S$ **do**
2: **for** $t \in T_s$ **do**
3: Let $W[1 \ldots |b|]$ be a new array ▷ $b \in B_s \cap B_t$
4: **for** $b \in B_s \cap B_t$ **do**
5: **if** $\max_{b' \in B} \sum_{s' \in S} \gamma_{s,t,b} > 0$ **then**
6: $W[b] \leftarrow \frac{rssi_{s,b,t}}{\max_{b' \in B_s} rssi_{s,b',t}} + \frac{\sum_{s' \in S} \gamma_{s,t,b}}{\max_{b' \in B} \sum_{s' \in S} \gamma_{s,t,b}}$
7: **else**
8: $W[b] \leftarrow \frac{rssi_{s,b,t}}{\max_{b' \in B_s} rssi_{s,b',t}}$
9: **end if**
10: **end for**
11: $\gamma_{s,t,b} \leftarrow 1,$ with $b \in B_s$ and $W[b] = \min_{b' \in B_s} W[b']$
12: **end for**
13: **end for**

In the first step, depicted in Algorithm 1 we iterate over all stations in S. This list of stations can be sorted based on a number of criteria. For instance, according to the arrival of the stations in the network or on the decreasing sum of rates (across all flows) per station. We have opted for the latter more greedy approach. For each supported technology per station, we create a map with an assigned score per available BSS (line 3). This score combines the relative distance from the station to each infrastructure device with the load on that AP or base station (lines 4–10). This score allows us to take into account the mobility of stations and the shared spectrum per infrastructure device. We distinguish two cases. First, the most common case where already at least one station has been assigned to a BSS, meaning the max load across all BSSs is larger than zero (line 6). Second, the initial case where no load was assigned yet (line 8). Here, we only take into account the relative distance to avoid a division by zero. Next, the station is assigned to the BSS with the lowest score (line 11).

Algorithm 2 Second step: Flow Path

1: Let $C[1 \dots |B|]$ be a new array
2: $C[b] \leftarrow \chi_b$ ▷ $\forall b \in B$
3: Let $T[1 \dots |B|]$ be a new array
4: **for** $f \in F$ **do**
5: $\lambda^{in}_{f,b} \leftarrow 1$ with $\gamma_{s_f,t,b} = 1$ and $C[b] = \max_{\gamma_{s_f,t,b'}=1} C[b']$ ▷ $\forall b' \in B_{s_f}, \forall t \in T$
6: $T[b] \leftarrow T[b] + \min\left(\left(d_{s_f,b} \cdot b_{s_f,b}\right), r^{in}_f\right)$
7: $C[b] \leftarrow \max\left(0, (\chi_b - T[b])\right)$
8: $\lambda^{out}_{f,b} \leftarrow 1$ with $\gamma_{s_f,t,b} = 1$ and $C[b] = \max_{\gamma_{s_f,t,b'}=1} C[b']$ ▷ $\forall b' \in B_{s_f}, \forall t \in T$
9: $T[b] \leftarrow T[b] + \min\left(\left(d_{s_f,b} \cdot b_{s_f,b}\right), r^{out}_f\right)$
10: $C[b] \leftarrow \max\left(0, (\chi_b - T[b])\right)$
11: **end for**

The second step of the heuristic is shown in Algorithm 2. We first create an array where we store the remaining capacity (initially the max capacity) per BSS (line 1 and 2). A second array represents the total assigned rates per BSS (line 3). We then iterate over all flows in F. Once again these flows can be sorted by decreasing rates. For each flow, we first assign a path for the incoming traffic by selecting the BSS with the most capacity remaining (line 5). Next, we update the traffic assigned to the selected BSS by adding the minimum from the allowed rate on the station (depending on the MCS) and the incoming rate of the flow (line 6). On line 7, we also update the remaining capacity of the selected BSS by subtracting the assigned rates from the approximation function χ_b. By doing so, we account for the loss in the maximal capacity of a wireless technology when more and more devices are added. After the selection of the path for the incoming traffic, we repeat the same for the outgoing traffic on lines eight to ten.

5 Results and Discussion

In this section, we evaluate the presented load balancing approach across a variety of scenarios. We focus on comparing the performance of the heuristic against the MIQP and demonstrating the scalability and versatility of the approach. For this, we mainly make use of simulation results obtained from the ns-3 event-based network simulator [34], complemented with a direct algorithmic evaluation in python. The structure of this section is as follows. First, we discuss the evaluation setups and the topology of the different scenarios. Next, we discuss in detail how we selected the values for the different required parameters. Afterwards, the performance of the approach, in terms of achieved throughput and execution time, is evaluated in a variety of static and dynamic scenarios.

5.1 Evaluation Setup

Most simulations are conducted using the NS-3.27 network simulator, where we implemented the entire ORCHESTRA framework [6], the MIQP problem formulation, and our heuristic approach. To optimally solve the MIQP we make use of the Gurobi Optimizer (7.5.2). All experiments are conducted on a single core of an Intel® Xeon® E5-2680 Processor running at 2.8 GHz and with 8 GB RAM. Furthermore, we also extended the basic NS-3.27 implementation to allow for multi-channel Wi-Fi networks. During all of our experiments, we assume two technologies present: IEEE 802.11n and IEEE 802.11ac (respectively, 2.4 GHz and 5 GHz Wi-Fi). Note that as our load balancing approach is fully technology independent, it is of less importance which technologies were selected for the evaluation. Every scenario has at least two APs that support both technologies. Dynamic rate adaptation for all devices is made possible through the Minstrel rate adaptation algorithm. Besides the generated traffic flows themselves, also the management traffic is considered in the simulations. In other words, the packets that contain monitoring information and configuration instructions, sent between the devices and the controller or vice versa, are also generated and transmitted. As such, our results consider the overhead of the management interactions.

As NS-3 emulates all packets within a network, simulation time grows exponentially when increasing network size and traffic number. In order to allow us to investigate the scalability of the approach to larger networks and to perform a rapid evaluation of algorithm parameters, we created a second experimental setup, outside of the NS-3 simulator. In Python, we implemented, on a 2016 Intel NUC, both the MIQP and heuristic approach. As before, the Gurobi Optimizer (7.5.2) is used to solve the MIQP. Furthermore, we created a framework that could artificially generate the required inputs for the algorithms. This allows us to easily test the impact of varying configurations of stations, APs, and flows, without the need for any network interaction or full network simulation. This was mainly used to investigate the execution time and scalability of the approaches, and not for optimality or network performance. For each experiment, we will clearly specify the ranges of values that were evaluated. Finally, we assume the presence of 4 technologies (e.g., IEEE 802.11n, IEEE 802.11ac, IEEE 802.11ad, and LTE).

For every scenario throughout the evaluation, we provide a comparison to a fully distributed baseline, where each device decides for itself to which AP to connect, based on the best RSSI values. For this baseline, we assume that when the RSSI of the current connection drops below a certain threshold, a better connection is selected (if present) for that device on that specific technology. The selection of the threshold value will be discussed in the next section. In other words, the baseline corresponds to the current state of the art, where one of the discussed multi-technology management solutions (Sect. 2.1) is in place, without the centralized intelligence, but with seamless handovers. Furthermore, we also compare against the performance of a fully randomized algorithm that selects uniformly at random for each station the corresponding infrastructure device (i.e., BSS) to connect to, and for each incoming and outgoing flow its path. Note that a comparison to our previous work is

not possible as that did not support the option for multiple APs in a single network [7].

In order to generate representative network topologies and conditions, several types of devices are defined, each with different mobility and traffic rates. This information is depicted in Table 1. 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 (all except for the televisions) moves around according to the Random Waypoint Model within a certain area, with a random start position and a uniformly random chosen speed between 0.3 and 0.7 $\frac{m}{s}$. The size of the area and the wait times at the waypoints depend on the scenario. Moreover, in 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 download 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 [35]. We decided to use only TCP traffic flows, as current Internet traffic is dominated by TCP [36]. Finally, for every described scenario, results are averaged over 20 different randomly generated flow and topology configurations.

5.2 Selection of Parameters

Both in the algorithms themselves as in the interaction with the network there are a number of parameters that can potentially have a large impact on the evaluation results. Below we discuss all parameters one by one and clearly highlight how the values are selected.

– Weight w for MIQP objective function: as the objective function of the MIQP is built out of two subfuctions, respectively, the throughput maximization function and the load balancing function, a weight is needed to combine both goals. Using

Table 1 Overview of the devices, and the supported flow rates, used in the scenarios

Device type (and mobility)	Rate boundaries per flow type		
	Download (Mbps)	Video stream (Mbps)	Conference call (Mbps)
Laptop (mobile)	10–30	8–20	4–10
HD Television	5–25	10–20	5–10
4K Television	5–25	15–25	7.5–12.5
Tablet (mobile)	1–8	2.4–9	1.2–4.5
Smartphone (mobile)	1–8	2.4–9	1.2–4.5

our Python experimental setup, we optimally solved the MIQP for a large number of scenarios, testing out a range of weights per scenario. We used Gurobi to provide an optimal configuration for a specific scenario and weight. Afterwards, we tried adding a random number of flows to the calculated configuration. The number of capacity requested by the flows that were available (normalized over the total capacity of the network), was added to the objective score calculated by Gurobi for that specific scenario and weight. In case no capacity was available the objective score did not change. Per unique combination of scenario and weight, we repeated this 20 times. In total we varied the number of stations between 5 and 15, the number of APs between 2 and 5, and the number of additional flow between 1 and half of the number of stations selected. While considering weights between 0 and 1 with a step-size of 0.01 between the different considered values. Finally, we averaged results per weight across all scenarios and normalized the scores between 0 and 1. Results are depicted in Fig. 2. From the figure, it is clear that the best performance was achieved when using a weight of 0.91. Furthermore, we see that for the value of 0.91, the network-wide throughput (as calculated by the MIQP) is at its highest, while this is not the case for the difference in load across the APs. Intuitively, this clearly shows that the major objective is to maximize network-wide throughput. However, to account for the dynamic behavior of traffic, the weight needs to be selected where the difference in load is minimized as much as possible, without strongly impacting network-wide throughput. During all the following experiments, the weight of 0.91 will be used. Note that for visualization reasons, we do not show the weights below 0.5, as they scored the lowest of all, while also the MIQP throughput is normalized between 0 and 1.

– Parameters α and β for the χ_b capacity approximation function: here we applied the method as described in detail in Sect. 3.4. Per technology, we considered a number of stations between 1 and 15, while varying the flow rates between the

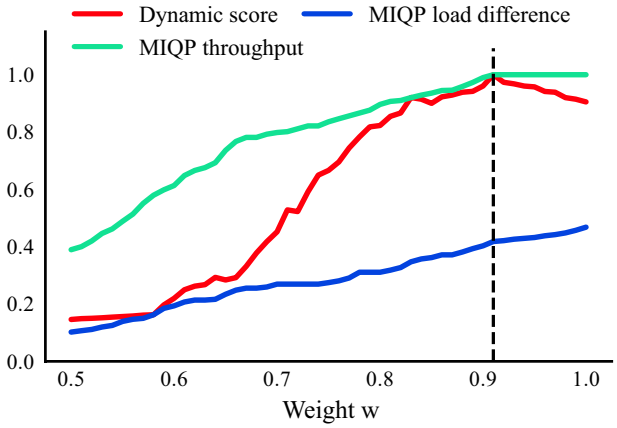

Fig. 2 Normalized score, optimal throughput (normalized), and optimal load difference over different scenarios for different values of the weight in the MIQP objective function

theoretical data rate of the particular technology and 1 Mbps. We determined the following parameters: for the function χ_b, α and β are respectively, for 2.4 GHz Wi-Fi − 1.74 and 57.58, and for 5 GHz Wi-Fi − 3.21 and 112.99.

- Algorithm execution parameters: the execution of the algorithm (either MIQP or heuristic) is triggered 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 the arrival of a new flow) or when it has been 10 s since the last execution. The latter ensures that the network configuration is optimized on a regular base, even in very static environments. The value of the parameter can be chosen based on the applicable environment. The first value (of 25%) was chosen based on a similar experiment as conducted for the weight of the objective function. In our NS-3 implementation, we tried out different values and selected the one with the highest impact. Furthermore, to avoid oscillations in the decision making and allow changes to occur in the network, there should be at least 2 s between two consecutive executions. The other two values were selected based on expert knowledge.
- MIQP time limit: in order to ensure the continuation of experiments and thus ending simulations in a feasible amount of time, a time limit is set for solving the MIQP. Here, a value of 900 s was chosen. Note that this value was chosen a magnitude larger than the number of time maximally available between two executions and required for reactive real-time optimizations. This allows us to sufficiently investigate the scalability of the MIQP in terms of execution time and show that it is not feasible to solve the MIQP in real-time.
- Baseline RSSI threshold: as mentioned in the previous section, a threshold is used to determine when a device needs to handover to a better connection (if existing). We chose a threshold of −75 for this, as this value is considered to still correspond to an average connection quality. Note that during some of the following experiments, we also tried out other threshold values (e.g., −65, −70, and −80) with limited differences.

5.3 Static Scenarios

In order to get a first impression of the performance of the different approaches, we created three basic scenarios with varying topologies. As depicted in Table 2, these scenarios grow in size and density. The results for all three scenarios are shown in, respectively, Figs. 3, 4, and 5. The graphs compare the baseline, MIQP formulation, random algorithm, and the heuristic to the sum of the desired flow rates (known as we use fixed flow rates here). Across all graphs, we clearly see a significant improvement by our approach in comparison to the distributed baseline and to the random algorithm. Moreover, we see nearly no difference between the heuristic and the optimal MIQP approach.

For the Home scenario, we can report the following rates (± the standard error), respectively for the baseline, random algorithm, MIQP, and heuristic: 81.61 Mbps (±2.62), 87.34 Mbps (±2.24), 90.15 Mbps (±2.36), and 89.84 Mbps (±2.37). There is thus an improvement of, respectively, 10.46 and 10.08% compared to the baseline

Table 2 Setup for static scenarios

Device	Home (20 × 10 m)	Small office (25 × 10 m)	Large office (30 × 15 m)	Flows
APs	2	3	4	N/A
Laptop	2	9	12	Download/Conf. call
HD TV	0	1	1	Video stream
4k TV	1	0	1	Video stream
Tablet	2	1	2	All types of flows
Smartphone	3	5	8	All types of flows
Total	10	19	28	

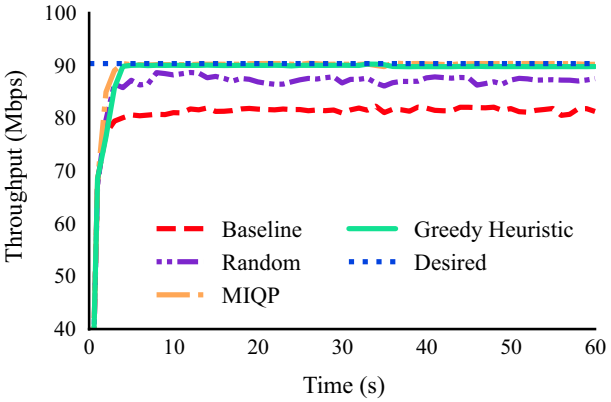

Fig. 3 Throughput as a function of time for the home scenario, comparing the heuristic, MIQP formulation, random algorithm, and the baseline

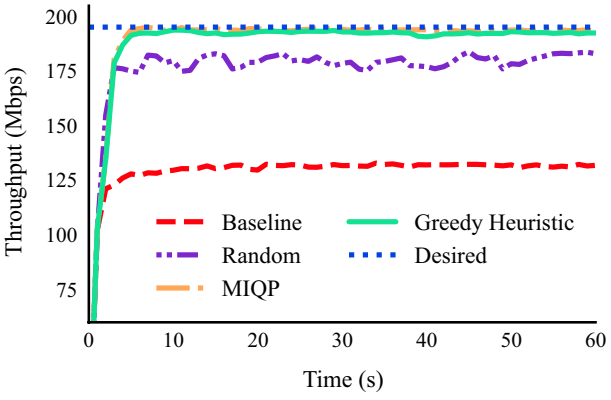

Fig. 4 Throughput as a function of time for the small office scenario, comparing the heuristic, MIQP formulation, random algorithm, and the baseline

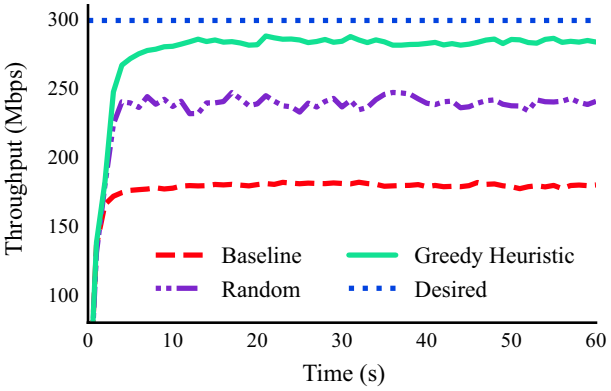

Fig. 5 Throughput as a function of time for the large office scenario, comparing the heuristic, random algorithm, and the baseline

for the optimal and heuristic solutions, while the difference of 0.31 Mbps between the MIQP and the heuristic solution is negligible. Furthermore, there is an increase of, respectively, 2.81 and 2.5 Mbps towards the random algorithm. Note that the random algorithm performs better than the baseline due to the fact that by selecting connections and flow routes at random, a simple form of load balancing is performed (on average). As the total desired rate is 90.40 Mbps (±2.35), it is clear that our approach succeeds in providing the optimal network configuration. Similarly for the Small office scenario, the following average rates are achieved: 131.46 Mbps (±3.73), 179.85 Mbps (±3.52), 193.90 Mbps (±3.76), 192.63 Mbps (±3.14), for respectively, the baseline, random algorithm, MIQP, and heuristic. The increases towards the baseline are larger than for the Home scenario: 47.50 and 46.53%, while also the increase towards the random algorithm is larger (respectively, 10.69 and 9.72%). The difference between the heuristic and the optimal MIQP solution is 1.27 Mbps (or 0.65%), which is once again negligible. The same can be said for meeting the requirements of the flows as the total desired rate is 195.21 Mbps (±3.46). For the Large office scenario, the largest network considered, it was impossible to calculate solutions for the MIQP within the time limit of 900 s. For, respectively, the baseline, random algorithm, and heuristic the following rates are achieved: 179.71 Mbps (±3.61), 239.41 Mbps (±3.41), and 283.60 Mbps (±3.61). This means that there is an increase of 103.89 Mbps or 57.81% towards the baseline. Compared to the random algorithm, there is an increase of 44.19 Mbps pr 24.59% If we compare the throughput of the heuristic to the overall desired rate, we see that the heuristic is 15.41 Mbps off. The reason for this is that the limits of the wireless technologies are being reached. This is also the reason for the fluctuations that can be seen in the throughput of the heuristic in Fig. 5.

As already mentioned, it proved to be infeasible to optimally solve the MIQP for the larger Office scenario. While it was possible to find a solution for the first two scenarios, the execution time was high: respectively, 16.38 s (±4.28) and 736.58 s (±39.71). Note that these execution times are significantly above the

minimal interval (of 2 s) between two consecutive runs of the algorithm. Luckily, the heuristic performs significantly better in terms of execution time: 8.23×10^{-5}s ($\pm 3.92 \times 10^{-5}$), 1.93×10^{-4}s ($\pm 1.24 \times 10^{-5}$), and 5.23×10^{-4}s ($\pm 2.58 \times 10^{-5}$) for, respectively, all three scenarios. We will discuss the scalability of both the MIQP and heuristic approach in more detail in the next section but the significant difference between the two is already clearly illustrated.

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 3, show that the algorithm always significantly outperforms the baseline. However, for the case with the highest mobility (and lowest wait times), the baseline performs significantly better, than in the other cases. We believe this to be due to the higher number of handovers, triggered by the mobility and its more reactive nature. Furthermore, the heuristic also outperforms the random algorithm across all scenarios.

5.4 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 was 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, 50, and 60% of the theoretical network capacity. Moreover, the presence of 3 APs was assumed in a space of 20 by 15 m with a waypoint wait time of 5–15 s.

From Fig. 6 it is clear that our heuristic approach offers a significant improvement towards both the baseline and the random algorithm. This improvement grows when the percentage of network traffic increases. For instance, for a load of 60% there is an increase from 135.57 Mbps ($\pm$ 2.50) for the baseline and 226.45 Mbps ($\pm$ 1.79) for the random approach, to 267.60 Mbps ($\pm$ 3.04) for the heuristic. This is an increase of, respectively, 97.39% and 30.36%. More importantly, we see that the heuristic allows, in general, to satisfy the traffic demands of all flows. Only at 60%, there is a difference of 35.58 Mbps (or 11.74% of the total demand) between

Table 3 Impact of mobility on throughput

	Wait times (s)	Baseline (Mbps)	Random (Mbps)	Heuristic (Mbps)
Home	0–10	83.16 ($\pm$ 3.31)	86.94 ($\pm$ 2.20)	89.74 ($\pm$ 2.31)
	5–15	81.61 ($\pm$ 2.62)	87.34 ($\pm$ 2.24)	89.84 ($\pm$ 2.37)
	10–20	80.32 ($\pm$ 2.88)	87.85 ($\pm$ 2.30)	89.35 ($\pm$ 2.25)
SME	0–10	157.19 ($\pm$ 4.70)	178.06 ($\pm$ 3.65)	191.03 ($\pm$ 3.80)
	5–15	131.46 ($\pm$ 3.73)	179.85 ($\pm$ 3.52)	192.63 ($\pm$ 3.14)
	10–20	135.46 ($\pm$ 3.98)	179.89 ($\pm$ 3.95)	193.16 ($\pm$ 3.19)
Office	0–10	229.47 ($\pm$ 6.22)	238.78 ($\pm$ 3.67)	282.91 ($\pm$ 3.62)
	5–15	179.71 ($\pm$ 3.61)	239.41 ($\pm$ 3.41)	283.60 ($\pm$ 3.61)
	10–20	178.73 ($\pm$ 5.06)	239.27 ($\pm$ 3.93)	286.56 ($\pm$ 3.30)

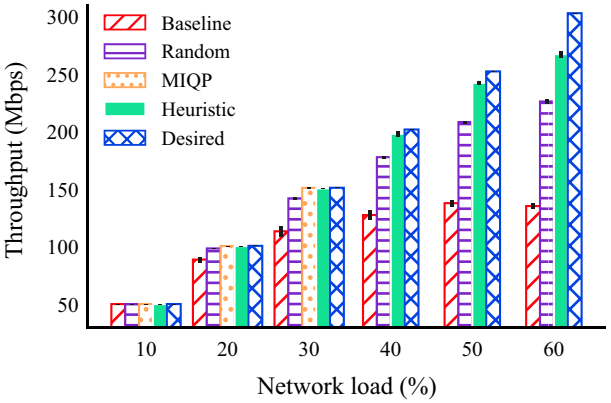

Fig. 6 Throughput as a function of network load, error bars depict the standard error

the desired rates and the achieved throughput. However, this is largely due to reaching the limits of the wireless technologies as our network loads are based on the theoretical capacities, which can not be met in reality due to capacity loss at higher layers (e.g., back-off timers, retransmissions, etc). Furthermore, we can also point out that there is nearly no difference between the optimal (MIQP) solution and the heuristic one, in terms of achieved network-wide throughput. For instance, at a load of 30% there is only a difference of 0.37 Mbps (respectively, 151.42 Mbps ($\pm$0.51) and 151.05 Mbps ($\pm$0.21)). Note that there are no throughput results depicted for the MIQP formulation for the network loads of 40, 50, and 60% due to the high computation time. This is similar to the Office scenario in the previous section.

We measured for both approaches the time it takes to calculate a solution. Table 4 shows the averages of the measured values across the different network loads. It is clear that the computation time for the MIQP scales exponentially. For instance, for only 14 flows (i.e., load of 30%) it takes already 478.36 s ($\pm$36.39) to compute the configuration. For higher loads, it was infeasible to calculate a solution within the time limit of 900 s. This clearly indicates that the MIQP solution cannot be used in very dynamic real-life wireless networks. The computation times reported when using the heuristic is drastically lower. For instance, for 14 flows it takes only 1.85×10^{-4} s ($\pm 8.81 \times 10^{-5}$) to find a solution.

Table 4 Comparison of the execution time for the MIQP and heuristic solutions, under increasing network load

Load	Flows	Exec. time MIQP	Exec. time heuristic
10	6	8.17 s ($\pm$1.08)	4.17×10^{-5}s ($\pm 1.62 \times 10^{-5}$)
20	10	29.75 s ($\pm$6.84)	1.25×10^{-4}s ($\pm 7.21 \times 10^{-5}$)
30	14	478.36 s ($\pm$36.39)	1.85×10^{-4}s ($\pm 8.81 \times 10^{-5}$)
40	19	N/A	2.16×10^{-4}s ($\pm 1.81 \times 10^{-4}$)
50	24	N/A	4.29×10^{-4}s ($\pm 2.06 \times 10^{-4}$)
60	29	N/A	5.90×10^{-4}s ($\pm 2.86 \times 10^{-4}$)

To further investigate the scalability of the heuristic approach, we performed a separate experiment, with our Python setup, emulating larger networks. We artificially provide the necessary inputs to the heuristic, thereby varying the number of stations between 100 and 10,000 and the number of APs between 10 and 1000. Furthermore, we assume the presence of 4 technologies (e.g., IEEE 802.11n, IEEE 802.11ac, IEEE 802.11ad, and LTE). For each pair of the number of stations and APs, we take the average of 20 executions, each with a randomly generated topology. Figure 7 shows the resulting heatmap, where every colored cell indicates the average time to solve the heuristic for a specific pair. We can see that the execution time mainly depends on the number of stations (and thus flows), and less on the number of APs. While the time increases when the number of stations grows, it stays under 3 s for all configurations. In particular, it takes on average 2.95 s ($\pm$0.07) to calculate a solution for the largest configuration. This means that the heuristic algorithm allows reacting to dynamic network changes and allows us to perform optimizations in real-time. In detail, at a second granularity for very large networks.

5.5 Dynamic Scenarios

Up to now, we have only considered scenarios with static flow rates and arrival times, as this helped in determining the impact of mobility and the use of real-time monitoring. We will now consider a more dynamic scenario, as this is more realistic and the adaptability to dynamic conditions is also key for our approach. All download flows thus act as in reality and consume as much bandwidth as possible until the desired number 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 randomly chosen between 5 and 15 s and 10 and 30 s,

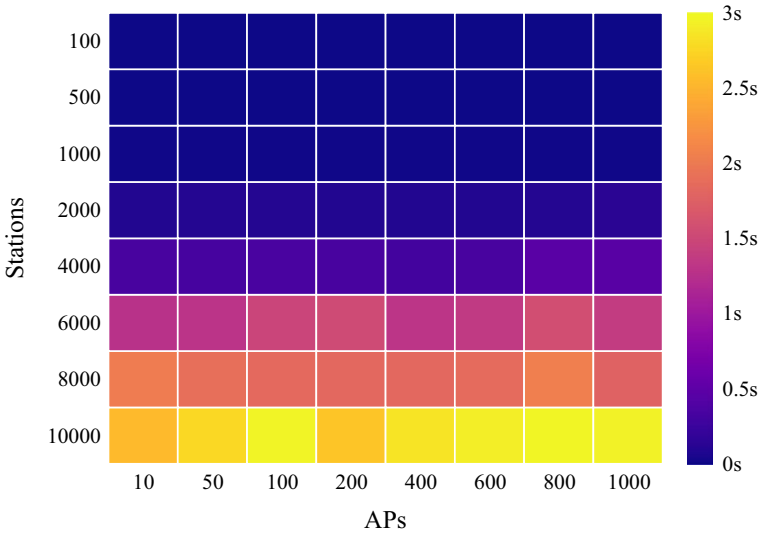

Fig. 7 Scalability of heuristic in terms of stations and APs

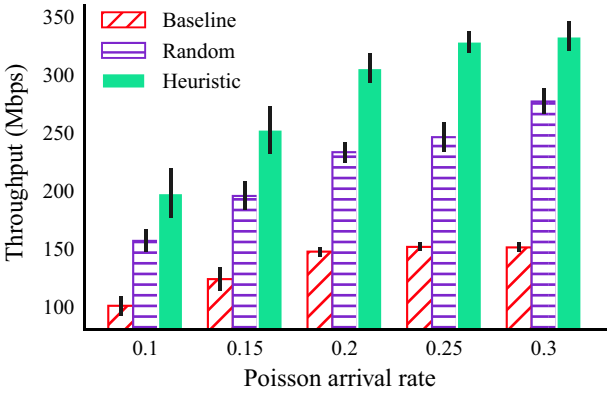

Fig. 8 Throughput as a function of poisson parameters, error bars depict the standard error, for a short flow length

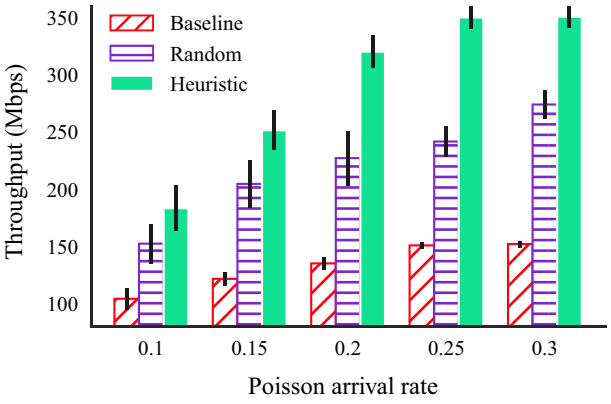

Fig. 9 Throughput as a function of poisson parameters, error bars depict the standard error, for a long flow length

respectively for the first and second scenario. Furthermore, the impact of different values for the Poisson arrival rate (λ) is evaluated.

Figures 8 and 9 shows that for all different arrival rates the heuristic approach significantly outperforms the distributed baseline and the random approach, across both scenarios. For the first scenario we see that for 0.1 as Poisson interval, the baseline achieves a throughput of 100.60 Mbps ($\pm$9.23) and 156.65 Mbps ($\pm$9.91), while the heuristic allows for a network-wide throughput of 197.87 Mbps ($\pm$21.48). When using a parameter value of 0.25, throughputs of 151.07 Mbps ($\pm$2.94), 246.18 Mbps ($\pm$12.73), and 328.77 Mbps ($\pm$9.35) are achieved, respectively for the baseline, random algorithm, and heuristic approach.

Equivalently, for the second scenario we see that for 0.1 as parameter value, the baseline achieves a throughput of 104.43 Mbps ($\pm$9.23), the random algorithm

attains 152.61 Mbps (±17.13), while with the heuristic 183.58 Mbps (±19.94) is achieved. For 0.25 as parameter value, the baseline and random approach achieve, respectively, a throughput of 152.30 Mbps (±2.86) and 241.87 Mbps (±13.51). In contrast, the algorithm allows for a throughput of 351.02 Mbps (±9.58). This is a gain of, respectively, 130.48% and 71.67%. Similarly to the experiments with varying network loads, the throughput of the heuristics does not further grow for the last parameter value (of 0.3), as the total network capacity has been reached. Finally, note that we have also repeated this experiment for other ranges of flow lengths, but the results were nearly identical and are therefore omitted.

6 Conclusions

This article addresses the need for intelligent management of heterogeneous wireless networks. We introduce a multi-technology load balancing approach that can balance devices across different APs and steer traffic across different paths through the network, on top of existing management frameworks and standards (like MPTCP). Our approach focuses on the dynamic and challenging environment of wireless networks and takes into account specific parameters such as the mobility of users and coexistence of multiple APs. This allows us to optimize the performance of the network in terms of network-wide throughput. We present both a mathematical problem formulation, through a MIQP, and a heuristic algorithm to ensure practical usability. A thorough evaluation, through a combination of extensive simulations, shows that the presented approach can indeed offer a significant improvement in terms of throughput. This throughput increase is more than twice as high for multiple scenarios compared to a state of the art baseline. Furthermore, we also show that the heuristic approach scales well (up to at least 10,000 devices in a single network) and that the heuristic can be used to adapt the network configuration to dynamic behavior at a second granularity. Future work includes, among others, the evaluation of the load balancing approach in real-life testbeds and investigating the applicability towards IoT networks (e.g., in terms of energy consumption) or VANETs (i.e., coping with a highly dynamic environment).

Compliance with Ethical Standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Cisco: Cisco Visual Networking Index: Forecast and Trends, 2017–2022. Cisco pp. 1–71 (2017)
2. Afaqui, M.S., Garcia-Villegas, E., Lopez-Aguilera, E.: IEEE 802.11ax: challenges and requirements for future high efficiency WiFi. IEEE Wirel. Commun. **24**(3), 130–137 (2016)
3. IEEE 1905.1: Standard for a Convergent Digital Home Network for Heterogeneous Technologies (2013)

4. Ford, A., Raiciu, C., Handley, M., Bonaventure, O.: TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, RFC Editor (2013). http://www.rfc-editor.org/rfc/rfc6824.txt
5. Hoymann, C., Astely, D., Stattin, M., Wikström, G., Cheng, J.F.T., Höglund, A., Frenne, M., Blasco, R., Huschke, J., Gunnarsson, F.: LTE release 14 outlook. IEEE Commun. Mag. **54**(6), 44–49 (2016)
6. De Schepper, T., Bosch, P., Zeljkovi, E., Haxhibeqiri, J., Hoebeke, J., Famaey, J., Latre, S.: ORCHESTRA: enabling inter-technology network management in heterogeneous wireless networks. IEEE Trans. Netw. Serv. Manag. **15**(4), 1733–1746 (2018)
7. De Schepper, T., Latré, S., Famaey, J.: Flow management and load balancing in dynamic heterogeneous LANs. IEEE Trans. Netw. Serv. Manag. **15**(2), 693–706 (2018)
8. De Schepper, T., Latre, S., Famaey, J.: Load balancing and flow management under user mobility in heterogeneous wireless networks. In: 2018 14th International Conference on Network and Service Management (CNSM), pp. 1–9 (2018)
9. Tessares: Hybrid Access Networks with MPTCP. https://www.tessares.net/
10. Rebecchi, F., De Amorim, M.D., Conan, V., Passarella, A., Bruno, R., Conti, M.: Data offloading techniques in cellular networks: A survey. IEEE Commun. Surv. Tutor. **17**(2), 580–603 (2015)
11. De Coninck, Q., Baerts, M., Hesmans, B., Bonaventure, O.: 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) **9631**(September 2015), 57–69 (2016)
12. Paasch, C., Ferlin, S., Alay, O., Bonaventure, O.: Experimental evaluation of multipath TCP schedulers. In: Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop—CSWS '14, pp. 27–32 (2014)
13. Khalili, R., Gast, N., Popovic, M., Le Boudec, J.Y.: Mptcp is not pareto-optimal: performance issues and a possible solution. IEEE/ACM Trans. Netw. (ToN) **21**(5), 1651–1665 (2013)
14. Mukherjee, A., Cheng, J.F., Falahati, S., Koorapaty, H., Kang, D.H., Karaki, R., Falconetti, L., Larsson, D.: Licensed-assisted access LTE: coexistence with IEEE 802.11 and the evolution toward 5G. IEEE Commun. Mag. **54**(6), 50–57 (2016)
15. Abinader, F.M., Almeida, E.P., Chaves, F.S., Cavalcante, A.M., Vieira, R.D., Paiva, R.C., Sobrinho, A.M., Choudhury, S., Tuomaala, E., Doppler, K., Sousa, V.A.: Enabling the coexistence of LTE and Wi-Fi in unlicensed bands. IEEE Commun. Mag. **52**(11), 54–61 (2014)
16. Zhang, N., Zhang, S., Wu, S., Ren, J., Mark, J.W., Shen, X.: Beyond coexistence: traffic steering in LTE networks with unlicensed bands. IEEE Wirel. Commun. **23**(6), 40–46 (2016)
17. Networks, M.: Truffle—Broadband Bonding Appliance. https://www.mushroomnetworks.com/truffle/
18. Macone, D., Oddi, G., Palo, A., Suraci, V.: A dynamic load balancing algorithm for quality of service and mobility management in next generation home networks. Telecommun. Syst. **53**(3), 265–283 (2013)
19. Oddi, G., Pietrabissa, A., Priscoli, F.D., Suraci, V.: A decentralized load balancing algorithm for heterogeneous wireless access networks. In: World Telecommunications Congress, pp. 1–6 (2014)
20. Olvera-Irigoyen, O., Kortebi, A., Toutain, L.: Available bandwidth probing for path selection in heterogeneous home networks. In: IEEE Globecom Workshops (GC Wkshps), pp. 492–497 (2012)
21. Bouchet, O., Kortebi, A., Boucher, M.: Inter-MAC green path selection for heterogeneous networks. In: IEEE Globecom Workshops (GC Wkshps), pp. 487–491 (2012)
22. Kortebi, A., Bouchet, O.: Performance evaluation of inter-mac green path selection protocol. In: 12th Annual IEEE Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), pp. 42–48 (2013)
23. Zekri, M., Jouaber, B., Zeghlache, D.: A review on mobility management and vertical handover solutions over heterogeneous wireless networks. Comput. Commun. **35**(17), 2055–2068 (2012)
24. Gódor, G., Jakó, Z., Knapp, Á., Imre, S.: A survey of handover management in lte-based multi-tier femtocell networks: requirements, challenges and solutions. Comput. Netw. **76**, 17–41 (2015)
25. Andrews, J.G., Singh, S., Ye, Q., Lin, X., Dhillon, H.S.: An overview of load balancing in hetnets: old myths and open problems. IEEE Wirel. Commun. **21**(2), 18–25 (2014)
26. Ng, B., Deng, A., Qu, Y., Seah, W.K.: Changeover prediction model for improving handover support in campus area wlan. In: Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP, pp. 265–272. IEEE (2016)
27. Harutyunyan, D., Herle, S., Maradin, D., Agapiu, G., Riggio, R.: Traffic-aware user association in heterogeneous lte/wifi radio access networks. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–8. IEEE (2018)

28. Lien, S.Y., Shieh, S.L., Huang, Y., Su, B., Hsu, Y.L., Wei, H.Y.: 5g new radio: waveform, frame structure, multiple access, and initial access. IEEE Commun. Mag. **55**(6), 64–71 (2017)
29. Parkvall, S., Dahlman, E., Furuskar, A., Frenne, M.: Nr: the new 5g radio access technology. IEEE Commun. Stand. Mag. **1**(4), 24–30 (2017)
30. Alizadeh, A., Vu, M.: Load balancing user association in millimeter wave mimo networks. IEEE Trans. Wirel. Commun. **18**(6), 2932–2945 (2019)
31. Donoso, Y., Fabregat, R.: Multi-objective Optimization in Computer Networks Using Metaheuristics. CRC Press, Boca Raton (2016)
32. De Schepper, T., Latre, S., Famaey, J.: A transparent load balancing algorithm for heterogeneous Local Area Networks. In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 160–168 (2017)
33. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **37**(6), 1067–1080 (2007)
34. Riley, G.F., Henderson, T.R.: The ns-3 network simulator. In: Modeling and Tools for Network Simulation, pp. 15–34. Springer (2010)
35. NoteTN2224, A.T.: Best Practices for Creating and Deploying HTTP Live Streaming Media for Apple Devices. Tech. rep., Apple (2012). https://developer.apple.com/library/ios/technotes/tn2224/_index.html
36. Lee, D.J., Carpenter, B.E., Brownlee, N.: Media streaming observations: trends in UDP to TCP ratio. Int. J. Adv. Syst. Meas. **3**(3), 147–162 (2010)

Tom De Schepper is a Ph.D. researcher associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from the University of Antwerp in 2015. His research focuses on providing optimizations and management solutions in multi-technology wireless networks. His research resulted in 11 articles published in international peer-reviewed journals and conference proceedings, as well as in a submitted patent application.

Steven Latré is an associate professor at the University of Antwerp and director at the research centre imec, Belgium. He is leading the IDLab Antwerp research group (85+ members), which is performing applied and fundamental research in the area of communication networks and distributed intelligence. His personal research interests are in the domain of machine learning and its application to wireless network optimization. He is author or co-author of more than 100 papers published in international journals or in the proceedings of international conferences.

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.