Andrea Ortiz, Arash Asadi, Gek Hong (Allyson) Sim, Daniel Steinmetzer, Matthias Hollick, "SCAROS: A Scalable and Robust Self-backhauling Solution for Highly Dynamic Millimeter-Wave Networks," in *IEEE Journal on Selected Areas in Communications: Special Issue on Millimeter-Wave Networking*. Fourth quarter 2019

SCAROS: A Scalable and Robust Self-backhauling Solution for Highly Dynamic Millimeter-Wave Networks

Andrea Ortiz†, Arash Asadi*, Gek Hong (Allyson) Sim*, Daniel Steinmetzer*, Matthias Hollick*
*Secure Mobile Networking lab (SEEMOO), Technische Universität Darmstadt
{firstname.lastname}@seemoo.tu-darmstadt.de
†Communication Engineering Lab, Technische Universität Darmstadt
{firstname.lastname}@nt.tu-darmstadt.de

Abstract*—Millimeter-wave (mmWave) backhauling is key to ultra-dense deployments in beyond-5G networks because providing every base station with a dedicated fiber-optic backhaul link to the core network is technically too complicated and economically too costly. Self-backhauling allows the operators to provide fiber connectivity only to a small subset of base stations (Fiber-BSs), whereas the rest of the base stations reach the core network via a (multi-hop) wireless link towards the Fiber-BS. Although a very attractive architecture, self-backhauling is proven to be an NP-hard route selection and resource allocation problem. The existing self-backhauling solutions lack practicality because: (i) they require solving a fairly complex combinatorial problem every time there is a change in the network (e.g., channel fluctuations), or (ii) they ignore the impact of network dynamics which are inherent to mobile networks. In this article, we propose SCAROS which is a semi-distributed learning algorithm that aims at minimizing the end-to-end latency as well as enhancing the robustness against network dynamics including load imbalance, channel variations, and link failures. We benchmark SCAROS against state-of-the-art approaches under a real-world deployment scenario in Manhattan and using realistic beam patterns obtained from off-the-shelf mmWave devices. The evaluation demonstrates that SCAROS achieves the lowest latency, at least $1.8\times$ higher throughput, and the highest flexibility against variability or link failures in the system.

I. Introduction

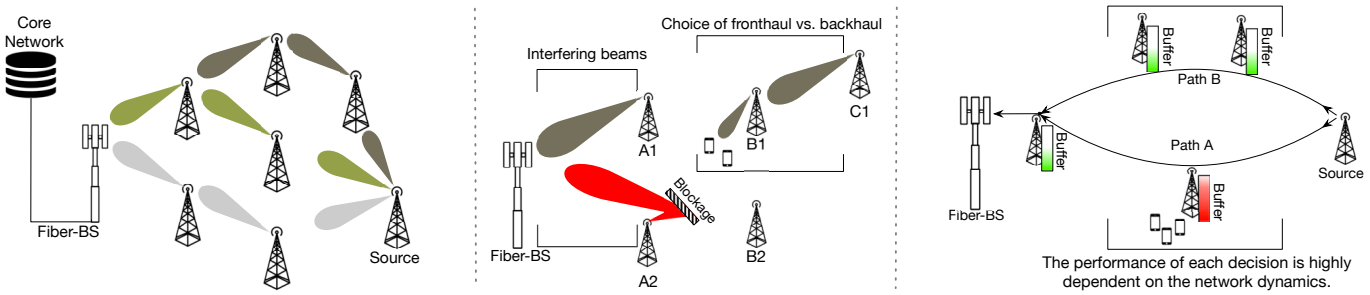
The mmWave bands (30-300 GHz) are mainly characterized by short communication range (due to high pathloss) and very high susceptibility to blockages such as the human body and buildings (due to high penetration loss). Ultra-dense deployment will be essential for millimeter-wave (mmWave) cellular networks in order to cope with the short communication range and excessive blockage possibilities [1], [2]. However, the traditional architecture where each base station has a dedicated fiber-optic link to the core network is unscalable and economically unjustifiable for mobile operators.

Recently, there has been a growing interest in self-backhauling solutions to cope with the required deployment density of mmWave networks [3]–[5]. In a self-backhauled network, fiber-optic connectivity to the core network is only provided to a portion of base stations (i.e., Fiber-BSs) whereas the rest of the base stations reach the core network through a (multi-hop) wireless connection which ends at the Fiber-BS, see Figure 1a. Although an appealing solution, self-backhauled mmWave networks face major challenges to meet the practical requirements of today's cellular networks in terms of computational complexity, QoS, and robustness.

A. Challenges

In essence, self-backhauling is a joint route selection and scheduling problem which requires careful consideration on the selection of path towards the core network, the allocation of frequency resources among base stations, the interference to/from other base stations, and the scheduling between backhaul traffic (data from other base stations) as opposed to the fronthaul traffic (data from the users associated to each base station). The multitude of the parameters under consideration coupled with highly dynamic nature of mobile networks as well as the susceptibility of mmWave signals to blockages [6]–[8] have made self-backhauling a real challenge for ultra-dense deployment in beyond-5G networks.

We elaborate on some of these challenges using an illustrative example in Figure 1. As shown in Figure 1a, several different paths (denoted by different colors) are available between the Fiber-BS and its associated self-backhauled base stations. Although the multitude of available routes potentially improves robustness, it also increases the complexity of the route selection. Furthermore, many links along these routes interfere with each other, due to the use of shared spectrum, and cannot be activated simultaneously. Thus adding an scheduling dimension to the routing problems. Figure 1b illustrates more practical constraints in terms of interference and resource allocation. Recent works in [2], [9], [10] indicate the inapplicability of the interference-free assumption for mmWave link (a.k.a. pencil beamforming or noise limited regime) in realistic scenarios due to imperfect beamforming and strong reflections in urban scenarios. For instance, base stations A1 and A2 (in Figure 1b) interfere with each other although their beams are theoretically pointed at different directions. Due to the first-order reflection, the interference caused by A2 is not negligible at the Fiber-BS. Furthermore, we also observe that the scheduling problem is not limited to the end-to-end route towards the Fiber-BS since each base station should also decide between carrying fronthaul and backhaul traffic (see base station B1 in Figure 1b). This decision impacts the overall



(a) (b) (c)

Fig. 1: An illustrative example of practical challenges faced by self-backhauled mmWave cellular networks.

latency experienced within the network. Figure 1c emphasizes the short life-time of every decision in the presence of network dynamics. For example, choosing the shortest path (Path A) is not always the best decision. In fact, many factors can turn Path A (initially the best decision) into a bottleneck causing a buffer overflow and packet drops. These factors include (i) load imbalance and bursty user arrivals (e.g., a bus/tram enters the cell), (ii) temporary blockage due to moving obstacles (e.g., large vehicles), and (iii) interference from a neighboring cell. Although the impact of these factors may be temporary, their occurrence is frequent. Thus, *mmWave self-backhauled networks are exposed to frequent and significant changes, which can turn a good decision in the current time slot to a poor choice for the next.* The multitude of parameters in play and their inter-dependencies do not allow for tractable modeling and often results in NP-Hard formulations.

B. Related work

The state of the art has approached the mmWave self-backhauling from an optimization point of view [1]–[4], [11]–[13], which has been proven to be NP-hard in its simplest form (e.g., in absence of interference and network dynamics) [1], [4]. Consequently, these works either rely on simplifying assumptions to obtain non-NP-hard formulations [2]–[4], [11], [13]; or leverage heuristics [12] and relaxation of problem constraints to find an approximate solution to the NP-hard problem [1]. Specifically, a noise-limited system (i.e., interference-free) is assumed in [1], [4], [11] to derive a mathematically tractable objective function. However, *mmWave networks are exposed to interference* due to imperfect beamforming in practice [10], [14] and first-/second-order reflections in urban scenarios [2], [9]. The fully-backlogged-queue assumption in [1]–[3], [11]–[13] does not account for the traffic variation and load imbalance across the network, which can lead to the creation of bottlenecks or sub-optimal performance due to resource overbooking as shown in Figure 1c. Furthermore, the majority of the proposed solutions are tailored for throughput maximization [1], [2], [4], [11], [12], despite the importance of latency in backhaul links. Due to the importance of adaptability in mmWave systems, many works have been dedicated to mitigate link outages due to beam misalignment problem caused by user mobility, blocked links in mobile scenarios, or both [7], [15]–[19]. Specifically, the authors in [7], [15] and [16], [17] design beam alignment algorithms to select the best beams with low overhead and delay. These algorithms operate on the basis of predicting the blockages [7], [15] and profiling the users' mobility [16], [17], respectively. Similarly, highlighting the need for fast beamforming in directional mmWave communications, the authors in [19] design a new beam alignment protocol (namely Agile-Link) which reduces the beam alignment delay in mmWave devices operating under 802.11ad to 2.5 ms. Aiming to improve steering accuracy for mobile users, the author in [18] experimentally show that light sensors aid in achieving the steering accuracy of more than 97%. The work of Yuan et al. in [1] is the closest to this article, in which the authors formulate the route selection and scheduling problem (under interference-free assumption) as a linear program which is then solved for throughput optimally using matching theory.

To the best of our knowledge, the prior works, in contrast to our paper, do not account for latency, interference, and traffic load. The other key missing feature is the lack of adaptability and robustness to network dynamics including link failures, traffic variation and channel fluctuations, that is, *any changes in the system requires recomputing the solution*. This feature is crucial in mobile networks where performance (e.g., throughput, latency) is highly affected by network dynamics such as noise and interference, mobility, blockage, user arrival rate, which are mostly uncontrollable and unavoidable, making adaptation the best remedy.

C. Our approach

The key issue in self-backhauling is the size of the problem (a.k.a. the curse of dimensionality). We tackle this problem first by focusing on *solution-oriented modeling* [20]. This approach is better understood when compared with causal modeling. In causal modeling, we often construct a complex system in which all major variables are included. For example, in a cellular network, we can build a model based on the most impacting variables, e.g., transmission power, pathloss, blockages, number of users, and interference. Solution-oriented modeling is, however, focused on reducing these variables to abstractions derived from our observations. For instance, a link with low throughput may suffer from high interference, high pathloss, or temporary blockage. In a solution-oriented

modeling, we focus on our observation of throughput rather than the underlying cause.

The causal modeling, which is often used for system behavior analysis, results in a much more complicated formulation since all independent variables should be individually included and studied. We follow a so-called *symptom-based* modeling methodology in which we formulate the problem based on the symptoms (e.g., throughput/latency variation) as opposed to the what caused them (e.g., interference, load imbalance, blockage). We choose this approach because a causal design, as elaborated before, results again in NP-Hard formulations, while the majority of the underlying causes remain outside our control, e.g., traffic load, blockage. The next step towards tackling the curse of dimensionality consists in devising a *semi-distributed design which breaks the problem into sub-problems* that can be solved in parallel. The feasibility of the solutions is then evaluated by a central entity (i.e., the Fiber-BS) which breaks ties in case of conflicting decisions, e.g., two base stations choose to transmit to the same next-hop node.

In this article, we propose the SCAROS algorithm, a latency-aware machine-learning-based approach which is not only highly scalable (due to the above-mentioned modeling strategy) but also robust against network dynamics by means of its quick adaption to new scenarios. The latter is facilitated through our proposed semi-distributed learning approach. The following summarizes our contributions and results:

- We model the route selection and scheduling in self-backhauled mmWave networks as a Markov Decision Process (MDP) that takes into account the sequential decision making nature of the problem. Moreover, it considers the stochasticity inherent to the calculation of the end-to-end latency.
- We provide the first reinforcement learning approach for mmWave self-backhauling. Our approach gives a solution to the MDP model and aims at minimizing the end-to-end latency as well as maintaining robustness against network dynamics. In contrast to state-of-the-art solutions, our proposed learning approach does not require prior information regarding the topology nor the network dynamics.
- We conduct an extensive measurement campaign in an anechoic chamber to identify the 3D beam patterns radiated from off-the-shelf mmWave antenna arrays. We integrate these patterns into our simulator to emulate realistic beamforming under real-world conditions.
- To provide intuition on the performance gains compared to other approaches, we benchmark our algorithm against both, a fully centralized reinforcement learning (RL) [21] and a recent self-backhauling solution based on linear optimization [1]. The results show that SCAROS achieves the lowest latency, at least $1.8\times$ higher throughput. The semi-distributed nature of SCAROS makes it more robust to network dynamics compared to the benchmark schemes.

II. System model

Our scenario consists of a multi-tier cellular network with two types of base stations, namely, Fiber-BS and self-backhauled base stations (S-BSs). The Fiber-BS is connected to the core network with a dedicated fiber backhaul link, whereas the S-BSs reach the core network via a multi-hop wireless link (except the S-BSs from the tier-1) towards a Fiber-BS (as shown in Figure 1a). Each base station is equipped with a single RF chain. We follow the integrated backhaul architecture (currently a 3GPP study item [22]) in which an S-BS can either communicate over backhaul or fronthaul link at each time instance. As specified by [22], all the communication is done over the same frequency bands (i.e., inband).

We consider a time-slotted system in which all the time slots have equal duration, and we denote each time slot with its index i, $i = 1, ..., I$, where $I \in \mathbb{N}$ is a finite time horizon. We denote $N_{\text{BS}} \in \mathbb{N}$ as the number of S-BSs associated to the Fiber-BS. Each S-BS $n \in \{1, ..., N_{\text{BS}}\}$ keeps track of the average queueing time of the packets in its buffer. This average queueing time is denoted by $t^{\text{queue}}_{n,i} \in \mathbb{R}^+$. Additionally, the transmission time from S-BS n to S-BS m in time slot i is denoted by $t^{\text{tx}}_{n-m,i} \in \mathbb{R}^+$.

Let us define a route selection and scheduling solution for time slot i as $\mathbf{X_i} \in \mathcal{X}$, where $\mathcal{X}$ is the set of all feasible route selection and scheduling solutions. Specifically, $\mathbf{X}_i$ is a $N_{\text{BS}} \times N_{\text{BS}}$ matrix calculated as:

$$\mathbf{X}_i = \begin{pmatrix} x^{(i)}_{1,1} & x^{(i)}_{1,2} & \cdots & x^{(i)}_{1,N_{\text{BS}}} \\ x^{(i)}_{2,1} & x^{(i)}_{2,2} & \cdots & x^{(i)}_{2,N_{\text{BS}}} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(i)}_{N_{\text{BS}},1} & x^{(i)}_{N_{\text{BS}},2} & \cdots & x^{(i)}_{N_{\text{BS}},N_{\text{BS}}} \end{pmatrix}. \quad (1)$$

$x^{(i)}_{n,m} \in \{-1, 0, 1\}$ firstly determines whether the link (n, m) between S-BS n and m, $n \neq m$, is activated in time slot i. Secondly, it indicates the direction of the communication, i.e, $x^{(i)}_{n,m} = 1$ when n transmits backhaul data to m, $x^{(i)}_{n,m} = -1$ when n receives backhaul data from m and $x^{(i)}_{n,m} = 0$ when the link is not activated. Furthermore, $x^{(i)}_{n,n} = 1$ means S-BS n is receiving data from the fronthaul. Our goal is to find a route selection and scheduling solution that minimizes the average end-to-end latency T computed as:

$$\bar{T} = \frac{1}{N_{\text{BS}}} \sum_{n=1}^{N_{\text{BS}}} \mathrm{T}_n(\mathbf{X}), \quad (2)$$

where the function $\mathrm{T}_n(\mathbf{X})$ is defined as the resulting end-to-end latency observed in S-BS n when the composite route selection and scheduling solution $\mathbf{X}$, with $\mathbf{X} = [\mathbf{X}_1, ..., \mathbf{X}_I]$, is used. The function $\mathrm{T}_n(\mathbf{X})$ is a stochastic function that depends on the location of the S-BS, user mobility, interference and queue dynamics.

III. Problem Formulation

Since the achievable performance is non-deterministic due to the stochastic nature of $\mathrm{T}_n(\mathbf{X})$, we formulate the average

end-to-end latency minimization problem as a Markov decision process (MDP). An MDP is a suitable tool for decision-making situations in which the outcome is partly random [21]. Moreover, as the route selection and scheduling does not depend on the previous loads but the current one, the system under consideration fulfills the Markov property.

An MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{X}, \mathbb{P}, \mathcal{T} \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{X}$ is a set of actions, $\mathbb{P}$ is a transition model and $\mathcal{T}$ is a set of rewards [21]. In our model, the state $\mathbf{s}_i \in \mathcal{S}$ is an N_{BS}-dimensional vector containing the queuing time of each base station in time slot i, i.e., $\mathbf{s}_i = t_{1,i}^{\text{queue}}, ..., t_{N_{\text{BS}},i}^{\text{queue}}$. The size $|\mathcal{S}|$ of the state space can be calculated as

$$|\mathcal{S}| = \prod_{n=1}^{N_{\text{BS}}} T_n^{\text{queue}}, \tag{3}$$

where T_n^{queue} is the maximum queueing time units in S-BS n. However, in the general case, the queuing time $t_{n,i}^{\text{queue}}$ of each S-BS scan take any value in a continuous range. As a result, the set $\mathcal{S}$ has infinite size. The action set $\mathcal{X}$ is composed by all the possible route selection and scheduling solutions $\mathbf{X}_i$ defined in Section II. Note that the size $|\mathcal{X}|$ of the action set, although finite, increases exponentially with the number of S-BS and can be calculated as

$$|\mathcal{X}| = \prod_{n=1}^{N_{\text{BS}}} |\mathcal{X}_n|, \tag{4}$$

where $\mathcal{X}_n \subseteq \mathcal{X}$ is the set of possible actions S-BS n can take. Furthermore, note that due to technical limitations, not all of these actions are feasible. For example, an S-BS cannot simultaneously transmit to a neighboring S-BS and receive data from the fronthaul using the available RF chain. Moreover, the feasibility of the possible route selection and scheduling solutions depends on the dynamics of the network, e.g., a neighboring S-BS can be temporarily not available. The transition model $\mathbb{P}$ defines the transition probabilities $\mathrm{P}(\mathbf{s}_{i+1}|\mathbf{s}_i, \mathbf{X}_i)$ of reaching state $\mathbf{s}_{i+1}$ after taking action $\mathbf{X}_i$ when in state $\mathbf{s}_i$, i.e., the probability of obtaining a given set of queueing times t_n^{queue} in the S-BS as the result of selecting a particular route selection and scheduling solution. Finally, the reward $T_i \in \mathcal{T} \subset \mathbb{R}^+$ indicates how beneficial it is to take action $\mathbf{X}_i$ in state $\mathbf{s}_i$, and it corresponds to the end-to-end latency in time slot i.

The solution of an MDP is a policy π which is a mapping from a given state $\mathbf{s}_i$ to the action $\mathbf{X}_i$ that should be selected, i.e., $\mathbf{X}_i = \pi(\mathbf{s}_i)$ [23]. In our case, the policy determines the route selection and scheduling that should be adopted given the queuing time in each S-BS.

IV. SCAROS ALGORITHM

Traditionally, dynamic programming techniques are used to solve problems formulated as MDP [21]. However, in the self-backhauling scenario the use of such techniques is not possible due to the lack of an accurate model for the transition probabilities among states that considers all the aspects of a real implementation, e.g., mobility, interference, and queue dynamics. Moreover, the dimensionality of the problem, i.e., the number of S-BSs to be considered, renders such techniques computationally intractable. To overcome these challenges, we propose SCAROS, a reinforcement learning algorithm that aims at minimizing the average end-to-end latency. By analyzing the characteristics of the problem and leveraging a semi-distributed approach, the proposed algorithm is able to overcome the curse of dimensionality while maintaining robustness against network dynamics. In the following, we describe different components of SCAROS in detail.

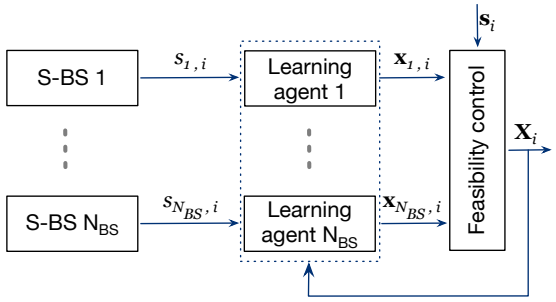


Fig. 2: Structure of the proposed semi-distributed learning approach

A. Semi-distributed approach

SCAROS benefits from a semi-distributed design that tackles the main issues of the MDP problem described in Sec. III, i.e., the infinite number of states and the exponential increase of the action space defined in (4). The structure of our proposed approach is depicted in Figure 2. The diagram illustrates the semi-distributed nature of SCAROS in which a learning agent is associated to every base station (S-BS). Each agent is responsible for learning the estimated time required to reach the Fiber-BS via the available neighboring base stations and consequently identifying the path which leads to minimum end-to-end latency. The intuition behind this approach is that by minimizing the average end-to-end latency of each base station, the overall average end-to-end latency in the network is also minimized. Furthermore, as the decisions of the different learning agents can lead to conflicting and non-feasible route selection and scheduling solutions, a central entity (i.e., Fiber-BS) is in charge of controlling the feasibility of the solutions and of breaking the possible ties. Note that the Fiber-BS operates on minimal information, i.e., the selected action of each S-BS. Looking at these actions, the Fiber-BS identifies whether the local decisions are infeasible. A detailed description of the feasibility control function is presented in Section IV-B.

In order to learn the average end-to-end latency of each S-BS, a multi-armed bandit (MAB) problem is formulated in every learning agent. The action set $\mathcal{X}_n$ of learning agent n is composed of the communication modes of the corresponding S-BS, i.e., transmission to the neighboring base stations and reception from fronthaul and backhaul. As a result, the *action space increases only linearly with the number of S-BS*. Formally, we define the action $\mathbf{x}_n^{(i)} \in \mathcal{X}_n$ selected by learning agent n in time slot i as the N_{BS}-dimensional vector

$\mathbf{x}_n^{(i)} = x_{n,1}^i, ..., x_{n,N_{\mathrm{BS}}}^i$, with $x_{n,m}^{(i)} \in \{-1, 0, 1\}$. Furthermore, considering that S-BS n can only transmit or receive backhaul data from its neighboring S-BS, the equalities:

$$\sum_{\substack{m=1 \\ m \in \mathcal{B}_n}}^{N_{\mathrm{BS}}} |x_{n,m}^{(i)}| = 1, \tag{5}$$

and

$$\sum_{\substack{m=1 \\ m \notin \mathcal{B}_n}}^{N_{\mathrm{BS}}} |x_{n,m}^{(i)}| = 0, \tag{6}$$

must hold for every $\mathbf{x}_n^{(i)} \in \mathcal{X}_n$ to ensure the feasibility of the actions. Note that $\mathcal{B}_n \subset \mathbb{N}$ represents the set containing the indices of the neighboring base stations of S-BS n, including its own index n.

For the selection of the actions, the state $s_{n,i} = t_{n,i}^{\mathrm{queue}}$ of S-BS n is used as the context information that determines which actions are available, e.g., when there is no packet in the queues, i.e., $t_{n,i}^{\mathrm{queue}} = 0$, only the reception modes are available. *By considering the state as context information, the infinite number of states of the original formulation is no longer a limiting factor.* Furthermore, to judge the suitability of the different possible actions, each learning agent learns the end-to-end latency associated with them.

We denote this end-to-end latency as $\bar{t}(\mathbf{x}_n^{(i)})$ and it corresponds to the expected time in which the packets from S-BS n will take to reach the Fiber-BS after selecting action $\mathbf{x}_n^{(i)}$ in time slot i. $\bar{t}(\mathbf{x}_n^{(i)})$ is updated in each time slot (when action $\mathbf{x}_n$ has been selected) as

$$\bar{t}_{i+1}(\mathbf{x}_n^{(i)}) = \bar{t}_i(\mathbf{x}_n^{(i)}) + \alpha_n \left(T_n(\mathbf{x}_n^{(i)}) - \bar{t}_i(\mathbf{x}_n^{(i)}) \right), \tag{7}$$

where $0 \leq \alpha_n \leq 1$ is a fraction that affects the learning rate and $T_n(\mathbf{x}_n^{(i)})$ is the reward obtained in the S-BS when action $\mathbf{x}_n^{(i)}$ is selected. Note that we have added the index i to $\bar{t}(\mathbf{x}_n^{(i)})$ to emphasize the fact that it is updated over time.

From (7), it is clear that immediate feedback of the effect of the selected action is needed to estimate the end-to-end latency. However, as the link between S-BS n and the Fiber-BS can be composed of multiple hops, this immediate feedback cannot be obtained because, in addition to the selected action $\mathbf{x}_n^{(i)}$, the end-to-end latency depends on the actions selected by the other S-BS. Therefore, to overcome this challenge, we define the reward $T_n(\mathbf{x}_n^{(i)})$ based on the state of the target neighboring S-BS $m \in \mathcal{B}_n$, the transmission time $t_{n-m,i}^{\mathrm{tx}}$ and its own estimates of the end-to-end latency. Formally, the reward $T_n(\mathbf{x}_n^{(i)})$ is defined as

$$T_n(\mathbf{x}_n^{(i)}) = \begin{cases} t_{n,i}^{\mathrm{queue}} + \min\limits_{m \in \mathcal{B}_n} (\bar{t}(\mathbf{x}_m)), & \text{if RX} \\ t_{n-m,i}^{\mathrm{tx}} + t_{m,i}^{\mathrm{queue}} + \min\limits_{l \in \mathcal{B}_m} (\bar{t}(\mathbf{x}_m)), & \text{if TX.} \end{cases} \tag{8}$$

At the beginning of the transmission, i.e., at the beginning of time slot $i = 1$, the queues of all S-BS are assumed to be empty. As a result, $t_{n,1}^{\mathrm{queue}} = 0, \forall n$. Furthermore, as no transmission has yet occurred, $t_{n-m,1}^{\mathrm{tx}} = 0, \forall n, m$ and the initial values of $\bar{t}(\mathbf{x}_n)$ depend only on the time-slot duration and the minimum number of hops, which S-BS n requires to reach the Fiber-BS.

B. Feasibility control

The decisions taken by every learning agent can lead to conflicting and non-feasible solutions for the network, e.g., two or more S-BSs might want to transmit to the same S-BS simultaneously. In order to prevent such cases, we incorporated a feasibility control function in SCAROS that, as the name suggests, controls the feasibility of the overall route selection and scheduling solutions. Additionally, this function is tasked with identifying and preventing the use of interfering links, i.e., links that when simultaneously activated lead to high interference, low throughput, and consequently larger latency. The identification of such links is performed locally when an S-BS detects interference from other S-BSs. The interference is then reported to the Fiber-BS to update the table for the interfering links. As a result, the Fiber-BS considers the selection of two interfering link as an infeasible action. The S-BSs can measure the interference from the neighboring S-BSs using the reference signals (see 3GPP 38.211, Section 7.4.1.5). Given the uniqueness of the reference signals, each S-BS is able to identify the interfering S-BSs as well as the severity of the interference caused by them. The result of these measurements is then reported to the Fiber-BS to update the table of interference. Based on the severity of the interference, the Fiber-BS decides whether the interfering links can be activated simultaneously. In this paper, we assume two interfering links can transmit simultaneously as long as the transmissions can be successfully decoded with the lowest MCS (i.e., the SINR is above the required level for the lowest MCS). If the aforementioned condition is not met, the links are added to the table of interference, which renders their simultaneous selection as an infeasible action.

To ensure the feasibility of the solutions, ties have to be broken when the different learning agents select conflicting actions. For this purpose, the feasibility control function takes into account the state $\mathbf{s}_i$ of the network. This means, in case two or more S-BSs are aiming for the same link, the resources are given to the S-BS that is experiencing higher latency at the moment. Algorithm 1 shows how ties are broken in every time slot. First, the current state $\mathbf{s}_i$, as well as the actions $\mathbf{x}_n$ selected by every learning agent, are observed (line 1). Then, the corresponding route selection and scheduling solution $\mathbf{X}_i$ is calculated by concatenating the actions selected by the learning agents (line 2). This concatenation is written as

$$\mathbf{X}_i = \left[{\mathbf{x}_1^{(i)}}^{\mathsf{T}}, ..., {\mathbf{x}_{N_{\mathrm{BS}}}^{(i)}}^{\mathsf{T}} \right]^{\mathsf{T}}. \tag{9}$$

To avoid the simultaneous activation of interfering links, let us define $\mathcal{I}_{n-m}$ as the set of identified interfering links when n transmits to m. For every S-BS n, we evaluate whether interfering links in $\mathcal{I}_{n-m}$ are simultaneously activated (line 4). In case there are, the state of the S-BS is considered and only the link associated to the S-BS that has the highest queueing

Algorithm 1 Feasibility control function

1: observe $\mathbf{s}_i$ and $\mathbf{x}_n$ for $n = 1, ..., N_{\mathrm{BS}}$
2: compute $\mathbf{X}_i$ ▷ Eq. (9)
3: **for** $n = 1, ..., N_{\mathrm{BS}}$ **do**
4: **if** $\mathcal{I}_{n-m} \neq \emptyset$ **then**
5: Find all the transmitting S-BS $l \in \mathcal{I}_{n,m}$ such that $\{s_{l,i} = \max(s_{k,i}) | (k = 1, ..., N_{\mathrm{BS}}) \cap (k \in \mathcal{I}_{n,m})\}$.
6: Deactivate all the other Tx S-BS $k \neq l$, $k, l \in \mathcal{I}_{n,m}$
7: **end if**
8: **end for**
9: **for** $m = 1, ..., N_{\mathrm{BS}}$ **do**
10: **if** $\sum_{n=1}^{N_{\mathrm{BS}}} x_{n,m} > 1$ **then**
11: Find all the transmitting S-BS l such that $\{s_{l,i} = \max(s_{k,i}) |_{(k=1,...,N_{\mathrm{BS}}) \cap (x_{k,m}=1)}\}$.
12: Deactivate all the other transmitting S-BS $k \neq l$
13: **end if**
14: **end for**
15: **for** $n = 1, ..., N_{\mathrm{BS}}$ **do**
16: **if** $x_{n-m} \neq x_{m,n}$ **then**
17: **if** $s_{n,i} \geq s_{m,i}$ **then**
18: Switch S-BS m to receiving mode
19: **else**
20: Switch S-BS n to receiving mode
21: **end if**
22: **end if**
23: **end for**

time $t_{n,i}^{\mathrm{queue}}$ is kept active (line 5). The remaining interfering links are deactivated, i.e., the interfering S-BS are not allowed to transmit in time slot i and switch to receiving mode (line 6). In the following step, the feasibility control function checks if two or more S-BS want to transmit to the same S-BS (line 10). Similar to the previous case, we break the ties by keeping active the link associated to the S-BS with the highest queueing time and switching the remaining conflicting S-BS to receiving mode (lines 11 and 12). In a similar fashion, we ensure that for every transmitting S-BS, the target neighboring S-BS is in receiving mode, i.e., $x_{n,m} = -x_{m,n}$. If this is not the case, then ties are broken according to the state of the S-BS (lines 18 and 20).

C. Summary of SCAROS Algorithm

So far, we have described the system model, the overall design, and different components of SCAROS. In this section, we provide a detailed step-by-step algorithmic overview of SCAROS, as summarized in Algorithm 2. First, the learning parameters are initialized at each learning agent (line 1). Then, for every time slot, the state of the network is observed, i.e., the queueing time $t_{n,i}^{\mathrm{queue}}$ of each S-BS is collected in the state vector s_i (line 3). For each learning agent, the state of the corresponding S-BS is given as an input and it is used as a context information to determine the set of available actions (line 5). From this set, each learning agent decides the action $\mathbf{x}_n$ to be used in time slot i using the ϵ-greedy policy (line 6). In the ϵ-greedy policy each learning agent selects a random action $\mathbf{x}_n \in \mathcal{X}_n$ with probability ϵ, and with probability $1 - \epsilon$ it acts greedily with respect to what it considers to be the best action so far, i.e., the one that provides the minimum estimated end-to-end latency $\bar{t}_{n-m,i}$. The ϵ-greedy approach provides a trade-off between the exploration of new and potentially better transmission paths and the exploitation of the known ones [21]. The actions selected by the learning agents and the state $\mathbf{s}_i$ are used as the input of the feasibility control function. This function, described in Section IV-B, breaks ties in the actions selected by the learning agents and avoids the simultaneous activation of interfering links (line 8). The resulting route selection and scheduling solution is then applied in the network and the rewards associated with it are distributed (line 9). These rewards are calculated using (8) and are used to update the estimated end-to-end latency of each S-BS using (7) (line 10). The same procedure is repeated for all the time slots $i = 1, ..., I$.

Algorithm 2 SCAROS algorithm

1: initialize α_n, ϵ_n at every learning agent
2: **for** every $i = 1, ..., I$ **do**
3: observe state $\mathbf{s}_i$
4: **for** each learning agent $n = 1, ..., N_{\mathrm{BS}}$ **do**
5: observe $s_{n,i}$ and determine available actions
6: select $\mathbf{x}_n \in \mathcal{X}_n$ using ϵ-greedy
7: **end for**
8: call the feasibility control function ▷ Alg. 1
9: apply $\mathbf{X}_i$ and observe obtained rewards ▷ Eq. 8
10: update $\bar{t}_i(\mathbf{x}_n^{(i)})$ for $n = 1, ..., N_{\mathrm{BS}}$ ▷ Eq. 7
11: **end for**

D. Computational complexity

In this section, we evaluate the computational complexity of one iteration of the proposed SCAROS algorithm with respect to the network size, i.e., the number of S-BSs (denoted by N_{BS}) in the network, and show that the resulting complexity is much smaller than that of traditional RL algorithms such as Q-learning and SARSA. To this aim, we first evaluate the complexity of the feasibility control function and then use this result to calculate the worst-case computational complexity of SCAROS. From Algorithm 1, it is clear that the most computationally demanding operations in the feasibility control function are the ones inside the for loops in lines 3-8, 9-14 and 15-23. For each of these loops, the worst-case complexity is $O(N_{\mathrm{BS}}^2)$. This value is obtained from the calculation of the index of the S-BS that has the maximum queueing time. This operation requires finding the maximum value in the state vector $\mathbf{s}_i$, which has a complexity of $O(N_{\mathrm{BS}})$. As this operation is repeated, in the worst case, N_{BS} times, the resulting complexity of each of the loops is $O(N_{\mathrm{BS}}^2)$. However, as the three for loops are independent of each other, the worst-case complexity of the feasibility control function remains $O(N_{\mathrm{BS}}^2)$.

Following a similar procedure, we evaluate the computational complexity of one iteration of the SCAROS algorithm. In Algorithm 2, the most computationally demanding operations are the for loop in lines 4-7 and the feasibility control function. In the first case, the use of the ϵ-greedy policy requires, in the worst-case, finding the action that leads to the minimum estimated latency. Assuming the worst-case scenario for network complexity, i.e., full connectivity among the S-BS, this requires finding the minimum out of N_{BS} values. This operation has a complexity that grows as $O(N_{\mathrm{BS}})$. However, note that this operation is performed in each learning agent separately. Consequently, the complexity remains $O(N_{\mathrm{BS}})$. In the second case, the complexity grows as $O(N_{\mathrm{BS}}^2)$, as mentioned in the previous paragraph. Thus, the computational

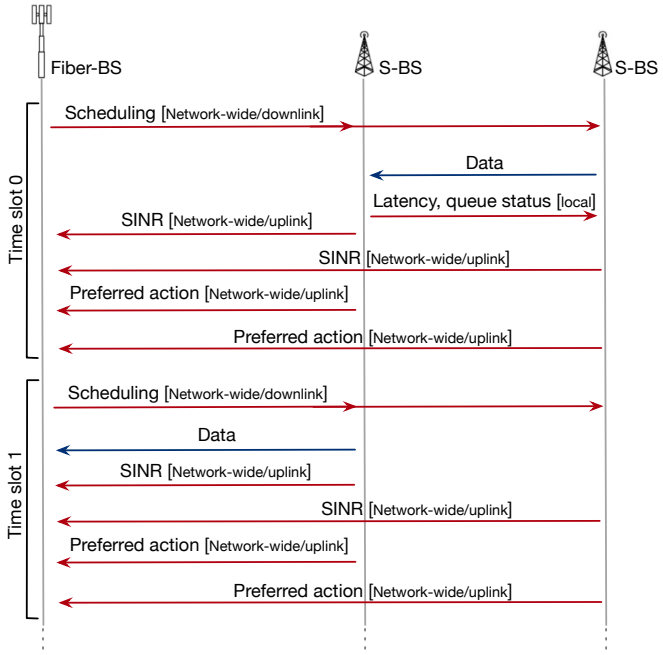


Fig. 3: The control and data exchange of SCAROS with reference to 3GPP TR 38.874.

complexity of SCAROS is also $O(N_{\text{BS}}^2)$. By a similar analysis, it can be shown that the computational complexity of the Q-learning and SARSA algorithms depends on the size of the action space $|\mathcal{X}|$ as defined in (4). This is because these two approaches solve a centralized learning problem in which all the possible route selection and scheduling solutions should be jointly considered. Therefore, as $\prod_{n=1}^{N_{\text{BS}}} |\mathcal{X}_n| \gg N_{\text{BS}}^2$, SCAROS achieves a much lower computational complexity than reference learning schemes.

V. Control Plane Aspects

Although mmWave backhauling has not yet been standardized, the 3GPP feasibility studies provide a roadmap towards the most plausible architecture. Given that technical feasibility has been the centerpiece of SCAROS's design, we based our design on 3GPP TR 38.874 self-backhauling study item [22]. The first generation of cellular mmWave deployments are expected to operate in non-standalone (NSA) mode in which mmWave base stations partially rely on 4G infrastructure. However, standalone (SA) deployments are expected to start from 2020 [5] in which 4G infrastructure is not required. Nevertheless, some level of separation between the control plane and the data plane is envisioned in both deployment modes [24], [25]. This separation allows for using sub-6GHz frequencies to carry part of control-plane messages since these frequencies provide higher communication range. The user data, on the other hand, is carried over mmWave frequencies. We follow the same architecture in this article. This is advantageous because *the achievable throughput and latency (1 ms in 5G NR) in sub-6GHz is sufficient for exchanging control messages among S-SBs and F-BSs.*

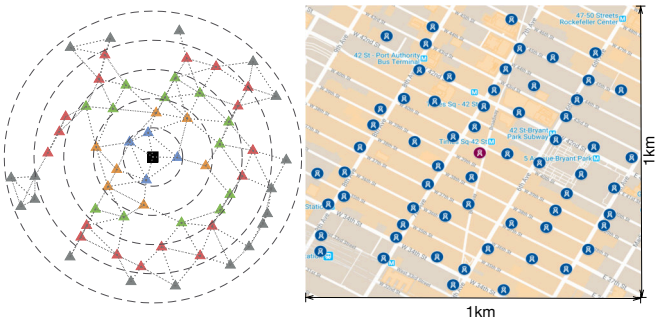


Fig. 4: The beam pattern and the evaluation scenario based on the base station deployment in Manhattan.

Figure 3 demonstrates how the required information for SCAROS (e.g., channel feedback, latency report) are communicated over the control plane (red arrows in the figure). The semi-distributed nature of SCAROS requires two types of messages: (i) Network-wide: the information exchanged between the Fiber-BS and its associated self-backhauled mmWave base stations. These messages carry scheduling information in downlink (similar to DCI in LTE), and SINR and local link activation preference in uplink; and (ii) Local: the control messages exchanged between a self-backhauled base station and its direct neighbors. The local control messages carry information regarding the estimated latency to the Fiber-BS and the queue status (similar to BSR in LTE). Note that the aforementioned information, with a pessimistic approximation, would be around 32 bits per active link per scheduling interval in uplink and 40 bits per scheduling interval in downlink. Assuming a network with 60 S-BS and scheduling interval of 5 ms, the total overhead would be 200 kbps. Such level of signaling overhead can be easily accommodated by 4G/5G NR in sub-6GHz bands.

VI. Evaluation

To date, there are no experimental platforms available for mmWave self-backhauling. The only available commercial platforms are the 802.11ad WiFi routers equipped with a 32-element phased antenna array[1,2]. However, the built-in energy detector in the firmware and CSMA-CA mechanism of 802.11ad does not allow for simultaneous transmissions to model the interference caused by different base stations. The other alternative is using a wide-band SDR platform [26] with commercial phased antenna arrays[3], whose price tag exceeds a few million euros even for a small setup with eight base stations. Nevertheless, we take the followings step to approach a realistic evaluation as much as possible:

- **Realistic beam patterns**. Instead of assuming perfect narrow pencil beams, we use the actual beam patterns obtained from our measurement campaign using TALON AD7200 mmWave routers equipped with a 32-element

[1]https://www.tp-link.com/en/home-networking/wifi-router/
[2]https://www.netgear.com/landings/ad7200/default.aspx
[3]https://www.anokiwave.com/products/index.html

phased antenna array. The measurements were performed in an anechoic chamber to ensure the beam patterns are not distorted by reflections and interference from other sources. We further elaborate on our 3D beam pattern measurement in Section VI-A.

- **Realistic deployment model**. Instead of using theoretical base station placement such as uniform and Poisson, we leverage the actual deployment in very dense urban areas. In particular, we parsed the location of the base stations in a 1 km^2 area in Manhattan through cellmapper website[4]. This area includes 60 base stations. Considering the expected effective range of mmWave base stations ($\sim$ 100 m), we divided this area into 5 tiers. Figure 4 shows the location of the base stations on the map and their separation into 5 tiers in our simulation.
- **Interference**. We refrain from the *street canyon* deployment model, which assumes that the majority of base stations are in NLOS and do not interfere with each other due to blockage. Looking at actual deployment in dense cities (e.g., New York, London), we observe that the rooftops are the most common deployment location for base stations in dense urban areas. Our evaluation considers the impact of interference between base stations, according to their location and selected beam pattern. Essentially, we compute the received signal power at each scheduled receiver by deducting the actual received power with the total interference power from all the other scheduled transmitters.
- **Latency measurement**. We timestamp every packet transmitted from the UE to the S-BS. These packets are then buffered at every base station and transmitted to the next hop on a FIFO manner. If the base station buffer is full, the newly arrived packet is dropped (i.e., tail drop). We measure the end-to-end latency when the packet is received at the Fiber-BS.
- **Periods**. For the simulations, we define a period as a set of consecutive time slots in which the number of S-BS and their respective average load remains constant. We use this separation to evaluate the adaptability of SCAROS to variations in the network size and load balance by having a known point of change, as described in Sections VI-C and VI-D. Note that this does not mean that the scenario remains static for the duration of the period. The variations due to noise, interference, mobility, blockage, and user arrival are still considered and impact the delays and throughputs achieved within the period.

We design three distinct scenarios (SC1 to SC3) in which we evaluate the performance of SCAROS against two benchmark schemes [1], [21], using the parameters listed in Table I. The first scenario focuses on benchmarking SCAROS in terms of throughput, latency, and packet drop as well as demonstrating the scalability of SCAROS with the network size. The second and third scenarios are designed to analyze the self-backhaul configurations and system performance under load imbalance and base station failure, respectively. Each simulation is repeated 10 times, and the simulation parameters for the mmWave path loss model are chosen according to 3GPP guidelines [27]. Note that the simulations are based on the 2D beam patterns. To provide intuition into the performance of centralized learning approaches as well as optimization-based solutions, SCAROS is benchmarked against the following schemes:

- **Reinforcement learning (RL)**. RL provides a solution of the MDP presented in Section III by means of the well known RL algorithm state-action-reward-state-action (SARSA) [21]. To handle the infinite number of states, we combine SARSA with linear function approximation. Specifically, we consider tile coding as an approximator of the state space. Additionally, to ensure the feasibility of the route selection and scheduling solutions, we break any possible ties randomly.
- **MTFS**. As described in Section I-B, MTFS is a linear programming-based approach proposed by Yuan et al. in [1]. By making decision centrally, MTFS provides the routes and their associated activation schedule that maximize the overall network throughput. However, MTFS does not account for interference, and it assumes fully backlogged queues. We chose MTFS to shed light on the performance of throughput-optimal approaches in the presence of changes in the networks. Otherwise, MTFS is an innovative solution which achieves an optimal result

[4]www.cellmapper.net/

TABLE I: Parameters used in the evaluation

Parameter	Value
Radio parameters	
Maximum number of S-BS	60
Maximum number of Fiber-BS	1
Maximum number of tiers	5
TX power	14.9 dBm
Carrier frequency	28GHz
Bandwidth	1GHz
Thermal noise power	-174 dBm/Hz
Noise figure S-BSs	4
Noise figure UEs	7
Learning parameters	
Number of realizations	10
Exploration probability ϵ	$1000/(1000+i)$
Learning rate α_n	0.1

Fig. 5: The measurement setup in the anechoic chamber (top) and the close-up view of the routers and the antenna array (bottom).

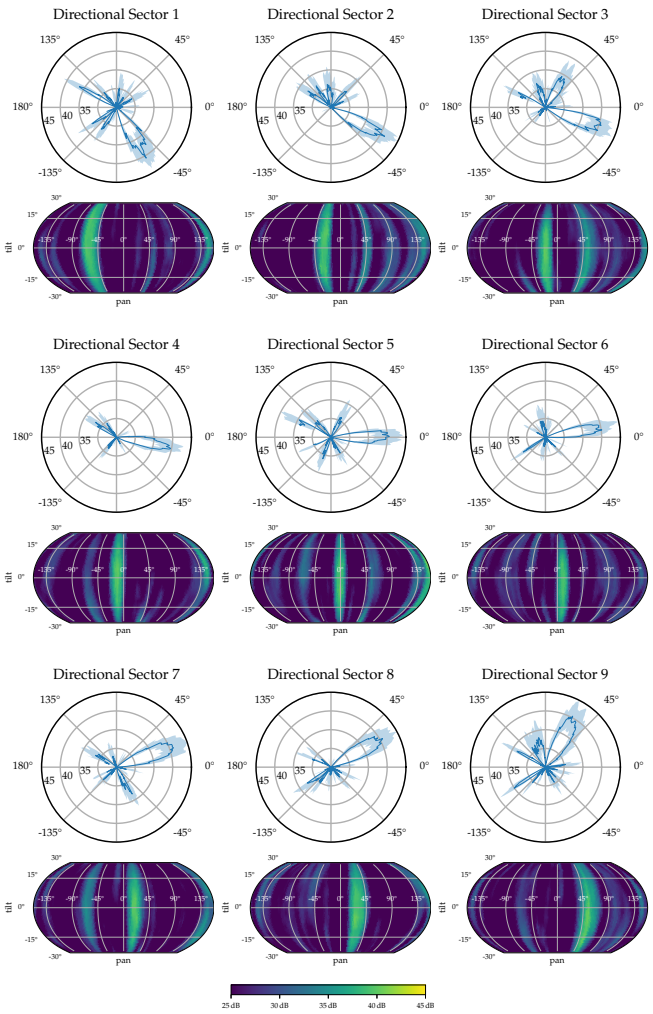


Fig. 6: Measured beam patterns of TALON AD7200.

in a short time within static networks.

A. Beam pattern measurement

The measurement setup used for measuring the radiation patterns consists of two Talon AD7200 devices in an anechoic chamber. As shown in Figure 5, we mount one device on a pan-tilt unit that uses two precise step-motors to orient it in different directions. In particular, we use a FLIR PTU-E46-70 that achieves an accuracy of up to $0.003°$. The Talon AD7200 is mounted with a custom adapter plate directly above the center of the rotation axis. The second device is placed three meters away on a tripod facing the first one. The walls, floor, and ceiling of the room are covered with radio-wave absorbing foam to omit disturbing reflections and multi-path effects. For each measurement, we establish a connection between both devices and perform frequent sector level sweeps. During this, both devices record the Signal-to-Noise Ratio (SNR) for received frames in all sectors. Using this setup, we measure the beam patterns both in 2D and 3D. To map the antenna radiation patterns in 3D space, we repeat our measurements and additionally tilt the rotation head from $-30°$ to $30°$. We use the same setup as before, take measurements at different pan angles from $-159°$ to $159°$, but decrease the accuracy to $2.25°$.

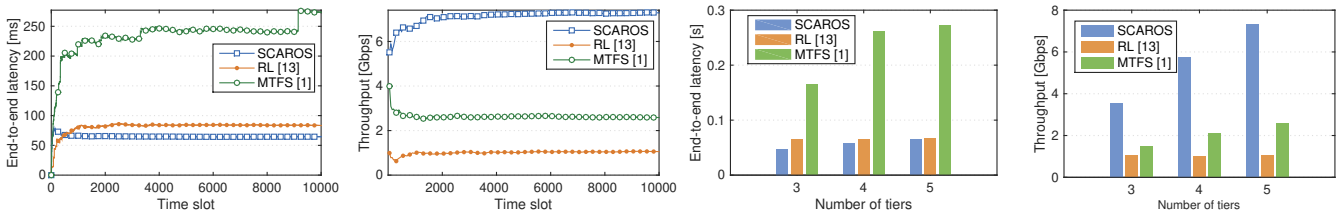
Figure 6 shows the measured beam patterns. Due to space constraint, we did not show all the 36 beam patterns. However, we intend to make these results publicly available in a Matlab file format so that other researcher can use our measurements for more realistic evaluations. Note that this is different from the beam patterns in [10] as we, through joint work with the leading authors in [10], include an algorithm that optimizes for the SNR at the direction of interest. As seen, although we have optimized the beamforming codebook to achieve the highest level directionality using commercial hardware, we can observe that the beam patterns are far from the theoretical narrow pencil beams.

B. SC1: Learning speed and scalability

We start the evaluation by analyzing the learning speed of SCAROS in Figure 7a. The figure shows the average end-to-end latency observed over time in a 5-tier network with 60 base stations. We observe that SCAROS outperforms all benchmark schemes both in average latency (up to 210 ms less than MTFS) and the convergence speed. Specifically, SCAROS achieves a latency of 67 ms only after 501 scheduling slots ($\sim 2.5s$ assuming 5 ms frame length). Figure 7b corroborate the superiority of SCAROS in terms of throughput. Figures 7c and 7d show latency and throughput performance of SCAROS in a 3-, 4-, and 5-tier network. SCAROS's ability to *maintain low latency while increasing the throughput as the network size increases is an indication of its scalability*.

Concerning the benchmarks, RL demonstrates comparable performance in terms of latency, but it achieves very low throughput. Theoretically, given infinite time, RL eventually matches the SCAROS's performance, but we did not observe this within our simulations due to the enormity of RL's state space. For example, in our implementation of RL with tile coding, we consider a precision of 0.1 in the approximation. This translates into approximately 10^{40} possible different states.

The poor latency performance of MTFS is due to the greedy approach toward maximizing throughput and ignoring the impact of interference and queue dynamics. The latter has a significant impact on MTFS's throughput since it activates as many links as possible simultaneously without accounting for interference. The reason behind the high performance of SCAROS is better illustrated in Figure 8. In this figure, we visualize the utilization of each link (i.e., the thickness of each link is proportional to traffic passing through it) and eventually the most selected routes. We can clearly observe that SCAROS selects shorter routes towards the Fiber-BS in comparison to the benchmark schemes and on average activates fewer links simultaneously, thus minimizing the impact of interference. In terms of throughput loss due to overhead, since the overhead is 200 kbps, this essentially means that the throughput loss is only 0.0027% of the average throughput.



(a) (b) (c) (d)

Fig. 7: The learning rate over time for 5 tier networks and the average throughput and latency for different network sizes (3, 4, 5 tiers). SCAROS demonstrates very fast learning rate while achieving the lowest end-to-end latency and the highest throughput.

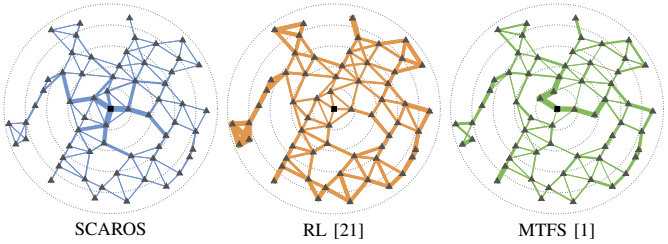
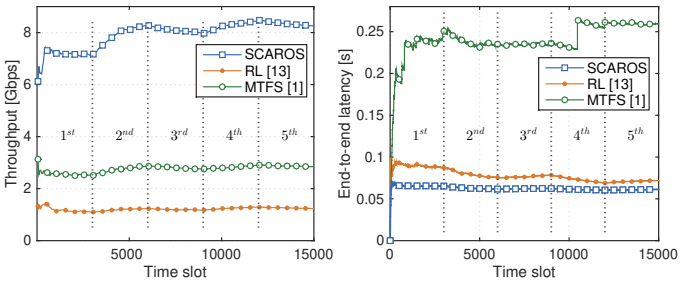


Fig. 8: Selected route and link activation. The thickness of each link is proportional to traffic passing through it.



(a) (b)

Fig. 9: The throughput and end-to-end latency of all schemes. SCAROS shows the highest robustness against load imbalance.

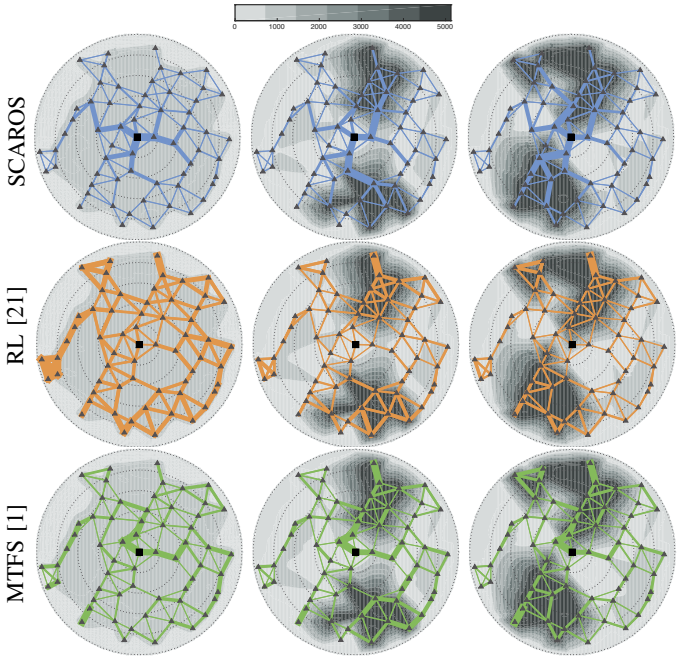


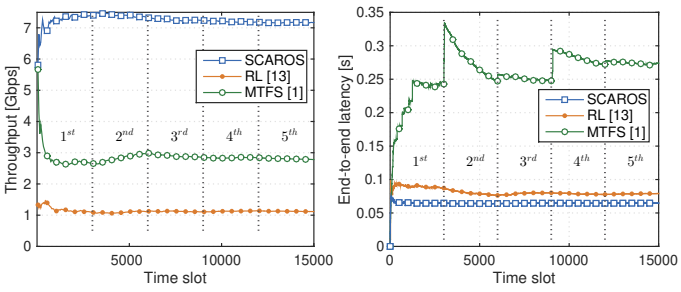
Fig. 10: The impact load imbalance on the selected routes under different algorithms. The overlaid heat map shows the traffic load, i.e., the intensity of gray shade increases with the load at the base station.

The *key insights* from this scenario are: (i) Machine learning is not a default solution for self-backhauling problems. Our observation showed that RL significantly reduces the throughput and requires a long time to converge to a steady-state; (ii) Throughput optimality under static and interference-free scenarios may not hold in realistic evaluations. MTFS is throughput optimal in the absence of interference and queue dynamics. However, it achieves lower throughput than SCAROS and leads to high end-to-end latency; (iii) SCAROS outperforms the benchmark schemes due to a tailored model designed to reduce the action space while accounting for network dynamics which hampered the performance of benchmark schemes.

C. SC2: Robustness against load imbalance

Here, we study the impact of load-awareness in self-backhauled networks. We divide this scenario into five periods, as shown in Figure 9. The load across the network is balanced in the first, third, and fifth period (1000 packet/s per base station), while we enforce load imbalance within the second and fourth periods. The heat map overlaid on the network topology in Figure 10 shows the load distribution during these periods.

The impact of load imbalance on throughput is shown in Figure 9a. Comparing this figure with the selected routes in Figure 10, we observe that SCAROS adapts to the new load imbalance quickly by selecting suitable paths, whereas RL fails to adapt to the changes, resulting in high packet drop. SCAROS not only achieves 5.35 Gbps higher throughput than MTFS but also it maintains steady latency performance against load variation, see Figure 9b. More specifically, we



(a) 7.42Gbps (b) 340ms

Fig. 11: The throughput and end-to-end latency of all schemes. SCAROS shows the highest robustness against base station failure.

can observe sharp latency increment (up to 35 ms) under MTFS, whereas such behavior is not seen in the learning-based approaches. The stability of SCAROS is due to the fact that when a change in the network occurs, SCAROS reacts by triggering more exploration of route selection and scheduling solutions. However, if the network topology does not change, the probability of exploration is decreased over time in order to exploit the gained knowledge. Besides, the algorithm's overhead only caused a throughput loss 0.0025% of the average throughput.

D. SC3: Robustness against base station failure

In Section I-A, we elaborate on how the complexity of self-backhauling problem is tied to the multitude of available routes toward Fiber-BS. However, if harnessed, this can be used as an opportunity to improve the robustness of the network against link/base station failures. In this scenario, we evaluate the impact of base station failure on the performance and the robustness of SCAROS and the benchmark schemes under such conditions, see Figure 11. Similar to the previous cases, this scenario is divided into five periods. In the first, third, and fifth periods, the network is in normal working conditions, and we enforce several base stations failures within the second and fourth periods. In Figure 12, the failed base stations are shown in red triangles.

In Figure 11, we observe the throughput and latency performance of all schemes in different periods. To be fair to MTFS (non-learning benchmark), we recompute the selected route before each failure period. Otherwise, MTFS would have experienced steep throughput drops due to routing data toward inactive base stations and allocating transmission times to links which cannot be activated. Nevertheless, we can still observe the impact of changes in routing and scheduling policy of MTFS on end-to-end latency, which results in sudden latency surge up to 96 ms. Nonetheless, SCAROS outperforms all the benchmark schemes in terms of throughput and latency. It can be observed, in Figure 12, that SCAROS quickly adapts to new scenarios by choosing alternative routes. We also see that SCAROS chooses to activate fewer links and schedule them longer, which contrasts the strategy of MTFS and RL. This

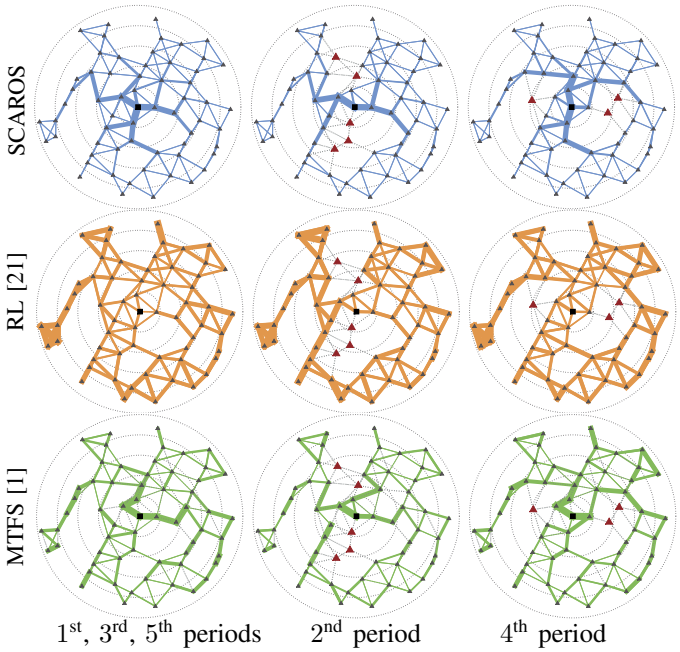


Fig. 12: The impact base station failure on the selected routes under different algorithms. The base station in red color and larger size are the failed base stations.

tendency towards activating fewer links reduces the overall interference in the network, thus enhancing the throughput. Noteworthily, SCAROS manages to keep the throughput loss at only 0.0034% of the average throughput even in the worst-case scenario (i.e., in the period with the minimum throughput).

VII. Discussion

In this section, we shed light on our choices when designing SCAROS and elaborate on its limitations. Furthermore, we provide insights into feasible solutions that can be potentially used in future work.

A. Learning speed

SCAROS was designed with a strong emphasis on achieving a high learning speed as well as robustness to network dynamics. Indeed, we observed in Section VI that SCAROS achieved our goals. However, we believe that the convergence speed and adaptability can be even further improved by training the algorithm before deployment. Considering the fact that the deployment topology is a priori known to the operators, it is not unreasonable to assume that the operators can simulate an expected set of link rates, traffic patterns and even link failures to collect training information for the algorithm prior to deployment. Furthermore, data collected from locations with similar characteristics (e.g., building density, height) can be used as training data. Note that we neither claim that the real network dynamics can be completely simulated nor propose to duplicate learning matrices from similar areas. Clearly, none of these approaches eliminates the need for

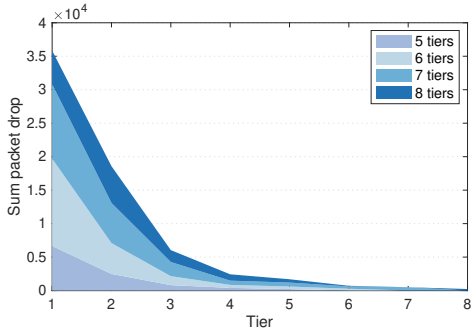


Fig. 13: The average packet drop per tier for different networks sizes (i.e., 5, 6, 7, and 8-tier scenarios).

online learning. Our suggestion merely enables the algorithm to make more informed decisions from day zero.

When reinforcement learning techniques are used, the learning agent learns how to act by interacting with the environment and receiving feedback about the selected action. This means that trying different route selection and scheduling solutions and evaluating the end-to-end latency. Even if some of these solutions lead to poor performance, it should be noted that only by allowing SCAROS to try all the different options, i.e., explore the action space, we can learn what solutions lead to the optimum performance. Nevertheless, taking into account that in a real implementation some of these sub-optimal solutions might not be allowed due to regulations about minimum performance, the prior training of SCAROS will reduce the requirement of in-situ exploration of the route selection and scheduling solutions. As a consequence, a minimum performance could be ensured from the beginning of the operation. Note that as the time evolves, the different solutions will still be evaluated according to local observations.

B. Scalability issues due to intermediary bottlenecks

Within the course of our evaluation, we observed that computational complexity is not exactly the only bottleneck for scalability of self-backhauling networks. Indeed, while SCAROS quickly learns the best strategy for large network sizes (up to 160 S-BSs in eight tiers), the performance appeared to be capped by a different factor, that is, the architecture itself. The self-backhauling architecture leads to a tributary effect, which automatically increases the probability of having bottlenecks in the inner tiers. This effect is better illustrated in Figure 13, where we show the average number of packet drops in a 5, 6, 7, and 8-tier network with 60, 97, 134, and 160 base stations, respectively. Note that the x-axis shows the number of packets dropped at every tier for different network sizes. We see in the figure that as the network size increases the number of packet drops increases dramatically. This result demonstrates that assuming Fiber-BS as the only bottleneck in the system, which is a common assumption in the literate [4], is in fact inaccurate. The majority of the base station in inner tiers will eventually become a bottleneck.

Although addressing this problem was out of the scope of this article, there are solutions which can alleviate this problem. A potential solution is increasing the number of RF chains at the base stations. More RF chains coupled with frequency duplexing will allow simultaneous transmission and reception, which can significantly improve the network capacity. From an economic aspect, this increases the end cost of each base station. A feasible solution to this problem is a non-uniform distribution for the number of RF chains per BS in which the number of RF chains is higher for the base stations closer to the Fiber-BS. The suitable distribution (e.g., Gaussian) is still an open problem, but we expect the number of tiers and distance to the Fiber-BS to be good candidates as characterization metrics.

C. Beamforming accuracy

In our evaluation, we intentionally refrained from using idealized beam pattern to emphasize the impact of interference. However, we expect the accuracy of beamforming techniques to improve, which brings us closer to idealized beam patters. We shed light on the impact of beamforming accuracy in Figure 14. In particular, the figure shows how fast the number of interfering link in our Manhattan scenario (5 tiers) drops as we reduce the beamwidth. Narrower beams mitigate the impact of interference and enhance the spatial reuse, which in turn increases the overall network capacity and reduces the queueing waiting times.

D. Multiple RF chains

In this paper, we have addressed the route selection and scheduling problem when only one RF chain is considered. Nevertheless, the semi-distributed design of SCAROS allows its seamless extension to the case when multiple RF chains are available at the S-BS and the Fiber-BS. In such a case, every S-BS learns the estimated set of times required to reach the Fiber-BS via the different available paths. As a consequence, the action set of each S-BS is no longer a vector but turns into a matrix whose dimensions depend on the number of available RF chains. Furthermore, the constraints given by (5) and (6) should be considered for each of the available RF chains independently.

VIII. Conclusions

In this paper, we present SCAROS, a machine learning approach to solve route selection and scheduling in self-backhauled mmWave networks under realistic network dynamics (e.g., channel variation, interference, mobility). SCAROS tackles the computational complexity of self-backhauling through a so-called symptom-based modeling coupled with a semi-distributed algorithm design. We benchmark SCAROS not only against a widely used learning approach but also a state-of-the-art linear optimization method. Furthermore, for a thorough evaluation, we set up a scenario based on the actual deployments in the Manhattan area and use the actual beam patterns obtained from our measurement of off-the-shelf mmWave devices. The evaluation demonstrated that SCAROS

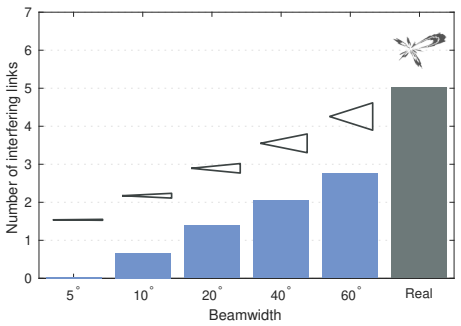


Fig. 14: The impact of beamwidth on the number of interfering links. We label the beam pattern obtained from our measurement campaign as *Real*.

improves the throughput by at least $1.8\times$ while achieving the lowest latency and providing the highest robustness against load imbalance and base station failures.

This is the first attempt towards using an online learning approach for solving self-backhauling problem. This work can be extended in several directions. For instance, we assume that the self-backhauled base stations (i.e., S-BSs) are a priori associated to a Fiber-BS, which is the conventional approach in today's networks. However, solving this problem in a network with arbitrary S-BS–Fiber-BS association is an interesting research avenue. Service guarantee is also an interesting future work in which the operators can ensure a certain level of quality of service in all tiers of their networks.

IX. Acknowledgements

This work has been performed in the context of the DFG Collaborative Research Center (CRC) 1053 MAKI - projects C1, A3, and B3 - and the Center emergenCITY of the LOEWE initiative (Hessen, Germany).

References

[1] D. Yuan, H.-Y. Lin, J. Widmer, and M. Hollick, "Optimal Joint Routing and Scheduling in Millimeter-Wave Cellular Networks." *IEEE INFOCOM*, pp. 1205–1213, 2018.

[2] Q. Hu and D. M. Blough, "Relay Selection and Scheduling for Millimeter Wave Backhaul in Urban Environments," in *International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 206–214.

[3] R. Taori and A. Sridharan, "Point-to-Multipoint In-band mmwave Backhaul for 5G Networks," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 195–201, 2015.

[4] M. N. Kulkarni, A. Ghosh, and J. Andrews, "How Many Hops Can Self-Backhauled Millimeter Wave Cellular Networks Support?" *arxiv.org*, 2018.

[5] A. Ometov, D. Moltchanov, M. Komarov, S. V. Volvenko, and Y. Koucheryavy, "Packet Level Performance Assessment of mmWave Backhauling Technology for 3GPP NR Systems," *IEEE Access*, 2019.

[6] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "Towards Scalable and Ubiquitous Millimeter-Wave Wireless Networks," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2018.

[7] M. K. Haider and E. W. Knightly, "Mobility resilience and overhead constrained adaptation in directional 60 ghz wlans: Protocol design and system implementation," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM, 2016, pp. 61–70.

[8] G. H. Sim, S. Klos, A. Asadi, A. Klein, and M. Hollick, "An online context-aware machine learning algorithm for 5g mmwave vehicular communications," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2487–2500, 2018.

[9] S. Niknam and B. Natarajan, "On the Regimes in Millimeter Wave Networks: Noise-Limited or Interference-Limited?" in *Proceedings of IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[10] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.1 lad Devices," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. ACM, 2017, pp. 414–425.

[11] J. Du, E. Onaran, D. Chizhik, S. Venkatesan, and R. A. Valenzuela, "Gbps User Rates Using mmWave Relayed Backhaul With High-Gain Antennas," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1363–1372, 2017.

[12] Y. Zhu, Y. Niu, J. Li, D. O. Wu, Y. Li, and D. Jin, "QoS-Aware Scheduling for Small Cell Millimeter Wave Mesh Backhaul," in *IEEE International Conference on Communications (ICC)*, 2016.

[13] M. E. Rasekh, D. Guo, and U. Madhow, "Interference-Aware Routing and Spectrum Allocation for Millimeter Wave Backhaul in Urban Picocells," in *Conference on Communication, Control and Computing (Allerton)*. IEEE, 2015, pp. 1–7.

[14] G. H. Sim, M. Mousavi, L. Wang, A. Klein, M. Hollick *et al.*, "Joint relaying and spatial sharing multicast scheduling for mmwave networks," *arXiv preprint arXiv:1907.13085*, 2019.

[15] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: Enabling Robust 60 GHz Links Under Blockage," in *USENIX NSDI*, 2016, pp. 193–206.

[16] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *IEEE INFOCOM*, 2017, pp. 1–9.

[17] A. Zhou, L. Wu, S. Xu, H. Ma, T. Wei, and X. Zhang, "Following the Shadow: Agile 3-D Beam-Steering for 60 GHz Wireless Networks," in *IEEE INFOCOM*, 2018, pp. 2375–2383.

[18] M. K. Haider, Y. Ghasempour, D. Koutsonikolas, and E. W. Knightly, "LiSteer: MmWave Beam Acquisition and Steering by Tracking Indicator LEDs on Wireless APs," in *ACM MobiCom*, 2018, pp. 273–288.

[19] H. Hassanieh, O. Abari, M. Rodriguez, M. Abdelghany, D. Katabi, and P. Indyk, "Fast millimeter wave beam alignment," in *ACM SIGCOMM*, 2018, pp. 432–445.

[20] M. S. Sumesh Krishnan, *Concepts in Engineering Design*. Notion Press, 2016. [Online]. Available: https://books.google.de/books?id=MP5BDQAAQBAJ

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (2nd edition)*. MIT Press, 2017.

[22] 3GPP, "Technical Specification Group Radio Access Network: Study on Integrated Access and Backhaul," Tech. Rep., Dec 2018.

[23] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[24] A. Tagami, K. Yokota, C. Sasaki, and K. Yamaoka, "Splitting Control-User Plane on Communication Protocol for Spotty Network," in *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, 2015, pp. 26–31.

[25] K. Yokota, A. Tagami, and K. Yamaoka, "Utilization of Content-Centric Networking for 60 GHz Spotty Networks," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking (ICN)*, 2016, pp. 223–224.

[26] S. K. Saha, Y. Ghasempour, M. K. Haider, T. Siddiqui, P. De Melo, N. Somanchi, L. Zakrajsek, A. Singh, R. Shyamsunder, O. Torres *et al.*, "X60: A Programmable Testbed for Wideband 60 Ghz Wlans with Phased Arrays," *Computer Communications*, vol. 133, pp. 77–88, 2019.

[27] 3GPP, "Technical Specification Group Radio Access Network: Study on channel model for frequencies from 0.5 to 100 GHz ," Tech. Rep., Mar 2017.

Andrea Ortiz (S'14) received the B.S. degree in electronic engineering from Universidad del Norte, Barranquilla, Colombia, in 2008. In 2013 she obtained the M.S. degree in Information and Communication Engineering from Technische Universität Darmstadt, Darmstadt, Germany. Currently she is pursuing the Ph.D. degree at the Communications Engineering Lab, Technische Universität Darmstadt, Germany. Her research interests include reinforcement learning for wireless communications, signal processing for wireless communications and energy harvesting communications.

Dr. Arash Asadi is appointed as Athena Young Investigator of Technische Universität Darmstadt. His research interests include wireless communications both in sub-6Ghz and millimeter-wave bands as well as their applications in emerging areas such as vehicular communication and industry 4.0. He is a recipient of several awards including outstanding PhD and master thesis awards from UC3M. Some of his papers on D2D communication has appeared in IEEE COMSOC best reading topics on D2D communication and in IEEE COMSOC Tech Focus.

Dr. Daniel Steinmetz received his Ph.D. in Computer Science from Technische Universität Darmstadt in 2019. His research focused on physical layer security and millimeter-wave communication systems. He was part of the Secure Mobile Networking Lab in Technische Universität Darmstadt and is currently working in ITK Engineering GmbH, Germany.

Dr. Gek Hong (Allyson) Sim is currently a Post-Doctoral researcher at the Department of Computer Science, Technische Universität Darmstadt, Germany. Allyson received her Bachelor Degree (Engineering majoring in Telecommunication) and first Master Degree (Engineering Science) from Multimedia University, Malaysia in 2007 and 2011, respectively. She was awarded with Master Degree and PhD Degrees in Telematics Engineering from University Carlos III Madrid in 2012 and 2015, respectively. Her research interest are multicast scheduling and MAC layer optimization for millimeter-wave networks.

Prof. Dr.-Ing. Matthias Hollick is currently heading the Secure Mobile Networking Lab in the Computer Science Department of Technische Universität Darmstadt, Germany. After receiving the Ph.D. degree from TU Darmstadt in 2004, he has been researching and teaching at TU Darmstadt, Universidad Carlos III de Madrid, and the University of Illinois at Urbana Champaign. His research focus is on resilient, secure, privacy-preserving, and quality-of-service-aware communication for mobile and wireless systems and networks.