

Distributed Delay-Aware Link Scheduling and Route Selection in mmWave IAB Networks

Yekaterina Sadovaya, Olga Vikhrova, Wei Mao, Omid Semiari,
Shu-ping Yeh, Hosein Nikopour, Shilpa Talwar, and Sergey Andreev

Abstract—Integrated Access and Backhaul (IAB) represents a fast and cost-efficient network deployment technology that enhances the coverage of millimeter-wave (mmWave) 5G networks. In addition to the conventional challenges of wireless multi-hop relaying such as, e.g., increased interference and packet delays, traffic asymmetry can lead to significant delay degradation. While centralized coordination can mitigate these challenges, it may also lead to unnecessary overheads. In this paper, we propose an effective delay-aware distributed solution for joint access and backhaul link scheduling and route selection designed to function with limited information, which relies only on the knowledge collected from immediate neighbors. We formulate the joint upstream and downstream routing and scheduling problem, which is solved in a distributed manner for the IAB system with diverse delay requirements. To effectively tackle this problem, we employ deep reinforcement learning (DRL) algorithms. Our numerical results demonstrate that the proposed distributed solution provides improved scalability as compared to the centralized approach without a significant performance loss.

Index Terms—IAB, dynamic traffic, millimeter-wave, link scheduling

I. INTRODUCTION

The ongoing research on networks is advancing, which promises significant support for high mobile broadband traffic volumes. Nonetheless, the scope of these networks is limited due to inherent factors associated with millimeter-wave (mmWave) systems such as notably higher propagation and penetration losses. The conventional approach of increasing the numbers of 5G NR nodeBs (gNBs) per unit area proves to be financially challenging and presents complexities in certain geographical areas, primarily due to wired connectivity of each gNB to the core network. Therefore, the Third Generation Partnership Project (3GPP) introduced Integrated Access and Backhaul (IAB) relaying. This technology aims to significantly expand the network coverage in a cost-effective manner [1].

The IAB architecture introduces wireless relays, which are termed IAB nodes, as part of the system that supports multi-hop network design. The IAB concept is proposed as a replacement for conventional fixed access and backhaul setups, as it enables flexible allocation of access and backhaul resources. Such a dynamic allocation enhances the overall efficiency

of the network infrastructure [2]. Another notable advantage of the IAB architecture is in higher uniformity of hardware and technology across both access and backhaul domains. This uniformity results in substantial reductions of hardware expenses and the elimination of fiber cabling costs. Moreover, the design of IAB networks incorporates a division between the control unit (CU) and the distributed unit (DU), which aligns conveniently with the requirements of fifth generation and beyond (5G/B5G) networks as it provides the necessary functionality for handling the time-critical and non-time-critical operations separately [3].

The integration of IAB nodes introduces such benefits as expanding the network coverage and enhancing the overall network performance, as demonstrated by several research works [4], [5]. However, the optimization of such networks presents a formidable challenge due to multiple factors. The dynamic nature of IAB networks, which is represented by variations in traffic loads and user mobility, necessitates a continuous process of re-optimization. Additionally, IAB systems encounter familiar obstacles inherent to wireless multi-hop networks, including increased interference levels and adherence to the half-duplex communication constraint. While there are several methods for optimizing different aspects of the IAB system operation, including topology formation, power allocation, routing, and link scheduling that are available in the literature [6], [7], sub-optimal delay-aware solutions for mmWave IAB networks subject to the aforementioned constraints remain underexplored.

A substantial volume of the existing literature addressing the scheduling and routing problems in IAB networks tends to overlook the aforementioned factors. For example, in [6], a maximum weighted matching (MWM) strategy is adopted to tackle the scheduling operations. However, it is noteworthy that the problem formulation itself assumes NP-hard complexity, which necessitates the solving of the MWM problem in each discrete time slot. This computational demand poses inherent challenges to the real-time operation of the system. Over the past years, reinforcement learning (RL) algorithms have gained popularity to address various scheduling and routing problems. Typically, these problems are solved by utilizing single-agent RL, which demonstrates improved convergence as compared to other heuristic strategies [8]. For example, the authors of [9] propose a single-agent RL framework for link activation

Y. Sadovaya, O. Vikhrova, and S. Andreev are with Tampere University, Finland. Email: firstname.lastname@tuni.fi

W. Mao, O. Semiari, S.-p. Yeh, H. Nikopour, and S. Talwar are with Intel Corporation, Santa Clara, CA, USA. Email: firstname.lastname@intel.com

with the goal of latency minimization. Furthermore, due to the dynamic nature of the network, deep RL (DRL) algorithm demonstrates improved adaptability and ability to generalize the learned policies across different scenarios [10].

However, single-agent and centralized RL solutions tend to face scalability issues [11]. Furthermore, while the formulation of a centralized scheduling and route selection problem may offer conceptual clarity, the actual implementation of centralized scheduling algorithms is challenging within practical systems due to continuous and resource-intensive control overheads required. On the other hand, distributed DRL methods demonstrated their capability to overcome the scalability challenge without significant performance loss [12]. Therefore, recent advancements in IAB research focus on scalable semi-centralized and distributed approaches, as evidenced by works such as, e.g., [13], [14]. However, it is essential to highlight that the goal of these studies lies primarily in data rate maximization.

Within this context, the objective of the present study is to introduce a feasible distributed algorithm designed to address the joint route selection and scheduling problem to satisfy various delay requirements. This includes the problem formulation, which is further addressed via DRL techniques. Moreover, the proposed approach assumes an in-band IAB system with non-negligible interference and distinct traffic demands of individual users in both uplink (UL) and downlink (DL) directions. Further, our distributed solution is benchmarked against the centralized setup to demonstrate the scalability benefits of the distributed approach and the associated trade-offs.

II. SYSTEM MODEL

A. Main Assumptions

We examine an IAB network with n_I IAB nodes, n_u UEs, and a single DgNB. At the same time, we denote the set of all nodes in the system as \mathcal{N} . We consider a system, where access and backhaul resources are shared, i.e., an in-band IAB network, which is commonly adopted due to interference limitations [15]. An example of target network deployment is provided in Fig. 1. The feasible communication links between IAB nodes and user equipment (UE) are determined by the network topology. Currently, IAB networks support both directed acyclic graph (DAG) and spanning tree topologies. In this study, we consider DAG topology as it is more generic and may include spanning tree topology as its special case.

For each source–destination pair in UL and DL, we consider a logical path that may traverse through multiple nodes, which we refer to as a flow. Therefore, there exists a total number of F traffic flows $F \leq 2N$, which encompass both UL and DL streams. Out of these flows, $\lfloor \delta F \rfloor$ flows are characterized as delay-sensitive (denoted by \mathcal{F}_1), while $\lceil (1 - \delta)F \rceil$ flows are delay-tolerant (denoted by \mathcal{F}_0). This partitioning of flows into delay-sensitive and delay-tolerant categories assists in managing the network performance across various use cases and traffic demands. It is worth noting that UEs are considered

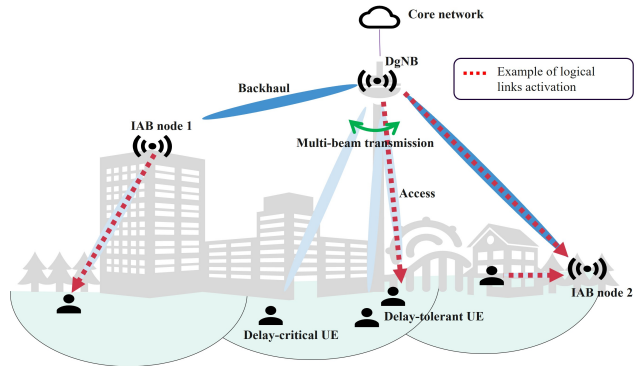


Fig. 1: Example of IAB network deployment.

to be mobile in our system. Moreover, we assume that new traffic arrivals follow the Poisson traffic model.

B. Problem Formulation

We require slotted time t in our framework, where the duration of a single slot is determined based on the selected 3GPP numerology. Within the centralized problem formulation, scheduling decisions are indicated by binary variables $a_{i,j}(t) \in \{0, 1\}$, where individual vector elements represent the activation of logical links from intermediate or end source i to destination j of flow f at time t . It is worth noting that each activation pattern within the centralized problem formulation accounts for the half-duplex constraint. Specifically, the obtained vector $a_{i,j}(t)$ does not include the activation of conflicting links, i.e., the use of incoming and outgoing flows at the same time.

To effectively transmit diverse traffic volumes from UEs, it is essential to note that packets can be distributed unevenly across the nodes. Therefore, the buffer states at each individual node as well as the capacities of the links should be estimated. To achieve this, we introduce the variable $Q_i^f(t)$, which represents the length of the queue associated with flow f at time t . Moreover, we denote the vector of queue lengths at time t as $\mathbf{Q}(t) = (Q_i^f(t))_{i \in \mathcal{N}}$. Hence, the link transmission rate is calculated as

$$b_{i,j}^f(t) = a_{i,j}^f(t) \min \left[Q_i^f(t), e_{i,j}(t) c_{i,j}(t) \right], \quad (1)$$

where $c_{i,j}(t)$ is the channel state, which represents the estimated link capacity between nodes i and j , $e_{i,j}(t)$ is the estimated channel loss at time t between node i and node j . The evaluated channel at time t for all the nodes is denoted as vector $\mathbf{E}(t) = (e_{i,j}(t))_{i,j \in \mathcal{N}}$. The evolution of the queues over time is captured by the following equation:

$$Q_v^f(t+1) = Q_v^f(t) - \sum_i b_{i,v}(t) + \sum_j b_{v,j}(t) + \Lambda_v^f(t), \quad (2)$$

where $\Lambda_v^f(t)$ denotes the new traffic arrivals.

Let $\mathbf{r} = (r_1, \dots, r_F)$ be the vector that characterizes the perceived flow rates. The objective is to find a strategy that satisfies the requirements $\mathbf{r} = (r_1^*, \dots, r_F^*)$, if such solutions are feasible. To reduce the packet delay of delay-critical flows when the traffic arrival rate is low, we introduce a penalty $u_f(t)$ that is expressed by

$$u_f(t) = \begin{cases} 0, & Q_d^f(t+1) = r_f^* \\ 1, & \text{otherwise,} \end{cases} \quad (3)$$

where Q_d^f is the destination queue of flow f . This objective can be realized by addressing the optimization problem as

$$\text{Minimize}_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \left(\sum_{f \in \mathcal{F}} (r_f^* - r_f(t_k))^2 + \sum_{f \in \mathcal{F}_1} u_f(t) \right) \right], \quad (4)$$

where t_k is the time that indicates the end of the frame. Policy π seeks to minimize the cumulative sum of two components: the initial term quantifies the difference between the desired flow rates \mathbf{r}^* and the achieved flow rates r_f for all flows $f \in \mathcal{F}$, while the subsequent term accounts for the scheduling delay penalties for delay-critical flows ($f \in \mathcal{F}_1$).

III. APPROXIMATE ALGORITHMS

A. Single-Agent RL

In Markov decision process (MDP) terminology, the methodology for single-agent RL can be described in the following way:

1) *Actions*: The actions correspond to the scheduling and routing decisions made in a centralized manner, i.e., $a_t \triangleq a_{i,j}(t)$.

2) *States*: The state at each time step t is represented by the statistics of queue states and channel matrix $\mathbf{E}(t)$ for all flows. Therefore, $s_t^n \triangleq (\mathbf{Q}(t), \mathbf{E}(t))$.

3) *Reward*: The reward computed in a centralized way is expressed as

$$R_t = \sum_{f \in \mathcal{F}} (r_f^* - r_f(t_k))^2 + \sum_{f \in \mathcal{F}_1} u_f(t). \quad (5)$$

B. Multi-Agent RL

Further, we adapt our formulation to multi-agent RL, where each IAB node and the DgNB utilize localized knowledge and act as individual agents by making autonomous decisions. Therefore, it leads to a more adaptive and scalable decision-making process. Similarly to the centralized case, we introduce the queue state $Q_n^f(t)$, which represents the queue length associated with node n . As compared to the centralized model, each individual IAB node and the DgNB make decisions. These decisions are indicated by the discrete actions $a_{i,j} \in \{1, 2, \dots, A(n)\}$, where $A(n)$ represents all possible combinations of simultaneously activated links for node n . Moreover, a continuous exchange of information

TABLE I: Parameters employed in our simulations.

Parameter Name	Parameter Value
Carrier frequency	30 GHz
Bandwidth	400 MHz
Number of UEs	10
Number of IAB nodes	3
Cell radius	1000 m
DgNB transmit power	40 dBm
IAB transmit power	33 dBm
UE transmit power	23 dBm
Noise figure of DgNB and IAB	7 dB
Noise figure of UE	13 dB
Power spectral density of noise	-173.93 dBm/Hz
UE antenna array size	4x4
DgNB/IAB antenna array size	16x16
Gain of a single antenna element	8 dBi
Maximum UE velocity	1 m/s
Minimum UE velocity	0.05 m/s
DgNB height	25 m
IAB height	10 m
UE height	1.5 m

occurs between child and parent nodes regarding the activated links. This exchange is designed to circumvent the constraints imposed by half-duplex communication, thereby ensuring that nodes do not transmit and receive simultaneously over the same link. In case of a conflict, the activation decision is based on the queue states corresponding to the flows, i.e., the most loaded link is activated.

The strategy employed for distributed scheduling and routing represents a sequence of scheduling decisions made by each individual node. The total number of these strategies is $N + 1$, which reflects the combination of strategies originating from node n . The queue and channel dynamics are captured in a similar way as in the centralized formulation. The main difference between the centralized and the distributed approaches is in the availability of the information. In the centralized model, it is assumed that the channel estimation matrix $\mathbf{E}(t)$ contains the estimated channel states for all the nodes and it is globally available to the network controller. On the other hand, in the distributed framework, the agents are only aware of the UEs, parent, and child nodes directly connected to them.

1) *Actions*: The actions in this setup correspond to the scheduling and routing decisions made by each node $a_t \in \{1, 2, \dots, A(n)\}$. These decisions determine how the traffic flows are routed through the network.

2) *States*: At each step, an agent observes the queue states of flows, which traverse through node (agent) n $Q^n(t)$. Therefore, the state at time slot t is $s_t^n \triangleq (\mathbf{Q}^n(t), \mathbf{E}_n(t))$.

3) *Reward*: As compared to the centralized formulation, each agent computes the reward individually for those flows, which traverse through node n . Therefore, each agent observes its individual reward, i.e., for node n it holds

$$R_t = \sum_{f \in \mathcal{F}} (r_{nf}^* - r_{nf}(t_k))^2 + \sum_{f \in \mathcal{F}_1} u_{nf}(t). \quad (6)$$

TABLE II: Average evaluated demand dissatisfaction rate.

Centralized PPO	Centralized PPO, $T = 5$	Centralized SAC	Distributed PPO
0.16	0.19	0.18	0.21

C. Distributed vs. Centralized Models

While it may be beneficial for the multi-agent environment to observe different rewards per agent as it makes the learning and evaluation processes more localized, this approach creates a challenge in comparing the performance levels of the distributed and the centralized models. This is because the individual rewards do not represent the overall satisfaction of the demands r . Therefore, we introduce the term that is named *demand dissatisfaction rate*, which is observed during the training phase for both single-agent and multi-agent RL frameworks. We denote the difference between the target and the achieved traffic requirements as

$$R_d = \sum_{f \in \mathcal{F}} [(r_f^* - r_f(t))H(r_f^* - r_f(t))]^2, \quad (7)$$

where $H(\cdot)$ is the Heaviside step function.

The demand dissatisfaction rate can be computed as $\frac{R_d}{|\mathcal{R}_d|}$. Specifically, this metric is the sum of the squared difference between the number of packets of flow f successfully delivered within T slots under policy π and the demand requirement r_f^* computed over all flows. As the number of delivered packets may be higher than the demand requirement, the formulation is multiplied by the Heaviside step function to ensure that only unsatisfied flows contribute to the computation of the demand dissatisfaction rate. By observing this parameter across all flows in both centralized and distributed models, we can fairly compare their performance. Moreover, this metric captures both delay and throughput performance. It is worth noting that this parameter reflects the overall convergence of the traffic requirements to the desired values.

IV. NUMERICAL RESULTS

In this section, we describe the adopted RL algorithms and present our numerical results. The parameters that were utilized by the simulation framework are provided in Table I. It is worth noting that the choice of the parameters is mostly impacted by the 3GPP recommendations for modeling IAB deployments [15]. The recommendation for carrier frequency is based on the use-case scenarios for IAB technology. Specifically, the relays are deployed primarily in mmWave spectrum, where network densification is needed. As demonstrated below in this section, the scalability of the centralized formulation limits the number of UEs and IAB nodes in the network. In the numerical results, we first calculate the demand dissatisfaction rate of the considered solutions during training and evaluation procedures. Then, we proceed with an estimation of wireless performance metrics such as delay and throughput.

A. Employed RL Algorithms

To numerically solve the formulated problem, we employ the proximal policy optimization (PPO) and soft actor-critic (SAC) algorithms. We selected these two options to compare the performance of the on-policy (PPO) and off-policy (SAC) methods. On the one hand, PPO is known for its stability, which makes it suitable for training DRL policies in dynamic environments. On the other hand, off-policy methods are generally more sample-efficient as compared to on-policy methods. However, both algorithms demonstrate the ability to generalize learned policies across different scenarios [16].

For both of the considered algorithms, the actor network generates a policy while the critic network is trained to evaluate this policy. However, there is a major difference in how the policy is learned by these methods. Specifically, the SAC algorithm learns a policy and two Q-functions in an off-policy manner. On the other hand, the PPO algorithm is on-policy, which means that it always aims to improve the current policy. Unlike conventional policy gradient methods, PPO employs a trust region approach by restricting policy updates to prevent drastic changes that could disrupt learning stability.

We first assess the performance of these algorithms under the single-agent RL case. Then, we choose the best-performing one for the multi-agent RL case. In the multi-agent setup, we assume that each agent learns its own individual policy. Therefore, at the end of the training, the number of the obtained policies is $n_I + 1$. Since the implementation of the idealized centralized algorithm is not possible in practical IAB systems due to the overheads associated with the centralized solution, we also consider a version with delayed feedback of the best-performing centralized algorithm. As compared to the idealized case, the delayed implementation samples the environment with a delay of T time slots, i.e., the states are updated after T slots. This means that the agent needs to generate a sequence of actions (a_0, \dots, a_T) instead of a single action a_t over a time interval between the updates. As a result, the agent learns the outcome of its actions by estimating the correlation between the states, traffic arrivals, and channel dynamics processes [17]. In our work, a fixed value of T is utilized while in reality the periodicity of the updates may be limited by the system implementation and the scale of network topology.

B. Demand Dissatisfaction Rate

We start with a comparison of the centralized algorithms. As can be seen in Fig. 2, the average demand dissatisfaction rates of the idealized centralized PPO and SAC after reaching the local minima are smaller than those for the distributed alternatives. This is observed because the channel states and the buffer lengths are known to all nodes in the centralized solution. However, when the feedback is delayed by $T = 5$ slots, the performance of the centralized PPO degrades. Moreover, the number of training steps required for it to reach a local minimum increases as well, since the algorithm needs

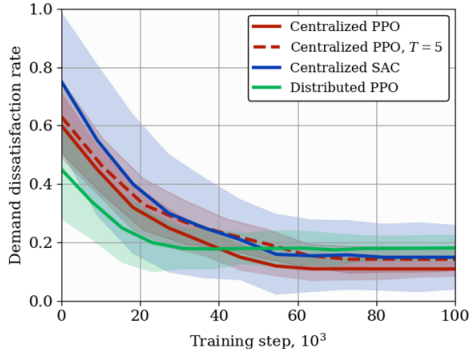


Fig. 2: Average demand dissatisfaction rate during training.

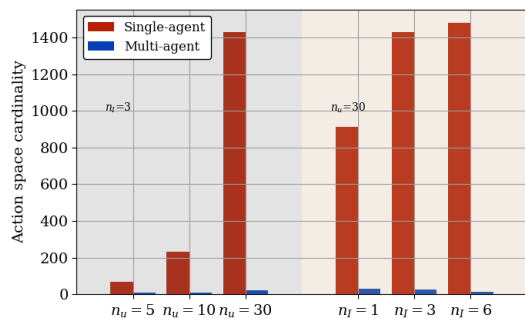


Fig. 3: Action space cardinality for different deployments.

to make a decision in advance before the next update occurs. On the other hand, the distributed solution achieves a local minimum much faster, as indicated by the earlier saturation in Fig. 2. This is because the action space is smaller for the distributed solution, since the set of actions comprises of only the activation combinations for flows, which traverse through the corresponding nodes. Moreover, it can be noted that among the considered centralized policies, i.e., PPO and SAC, the former provides more stable results for the formulated problem. This effect is observed because SAC employs entropy regularization as a key component of its objective function. This encourages the agent to explore more diverse actions, which leads to improved exploration strategies. Additionally, PPO reaches a lower value of the demand dissatisfaction rate and demonstrates reduced variability.

Fig. 3 depicts the action space cardinality for both distributed and centralized formulations. For the latter, the total number of actions is counted to compute the cardinality, while for the distributed case, the cardinality is calculated for each agent. Hence, Fig. 3 shows the average cardinality of the action space among the agents. As can be noted based on Fig. 3, the complexity of the centralized formulation grows significantly when more UEs and/or IAB nodes are added to the scenario.

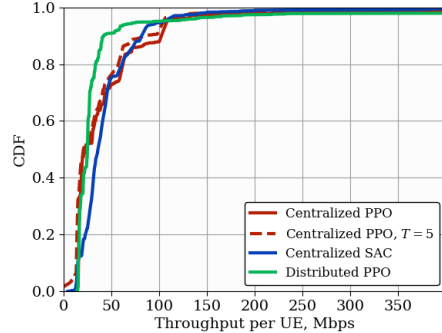


Fig. 4: Throughput performance on evaluation data.

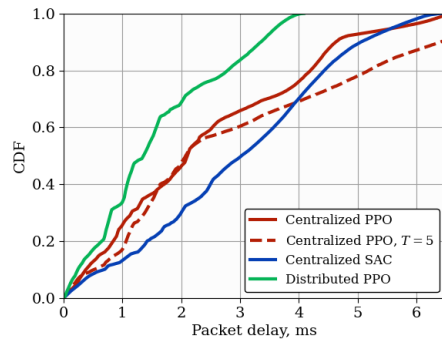


Fig. 5: Delay performance on evaluation data.

For example, as the number of UEs increases from 5 to 30, the action space cardinality of the centralized formulation grows by more than 10 times. On the other hand, the complexity of the distributed formulation remains the same. Moreover, an increase in the number of IAB nodes may even cause a decrease in the average cardinality per agent. This is because the distributed formulation only considers the links that pass through a certain node. Therefore, with a larger number of IAB nodes, the share of such links decreases as the load is distributed among them.

It is worth noting that the above results were obtained during training to demonstrate the computational complexity of the algorithms. Therefore, it is essential to consider the evaluation data set to estimate the performance of target solutions. The average demand dissatisfaction rate value during evaluation are summarized in Table II. It can be observed that the trend is similar to the one during training. Specifically, both centralized approaches outperform the distributed case. However, the difference between the average performance levels of the distributed and centralized PPO is about 3–5%.

C. Throughput and Delay

Further, we assess the system throughput and delay of the centralized and distributed algorithms. For this purpose, Fig.

4 illustrates the achieved throughput under the centralized and distributed strategies. It is important to note that comparing these algorithms solely in terms of throughput may yield limited insights, as throughput is inherently constrained by the specific traffic flow requirements. Consequently, given that all algorithms directly aim at meeting these requirements, discerning substantial differences in their throughput performance proves to be challenging.

The delay performance is reported in Fig. 5. It can be observed that the distributed PPO achieves the lowest delay as compared to all of the considered centralized algorithms. Specifically, it demonstrates two times lower delay as compared to the centralized SAC. On the other hand, the average difference between the delayed and the idealized PPO is insignificant. While the reward does not directly capture the delay, this effect can be attributed to the indirect influence of the penalty associated with the satisfaction of delay-critical requirements.

While acknowledging the superior performance of centralized solutions, it is essential to recognize the impracticality of their implementation in real-world systems. Specifically, it is worth noting that within a centralized formulation, it is assumed that information about the buffer and channel states is available to the centralized controller in every time slot. However, in reality, it will take time to collect the statistics. Moreover, perfect knowledge of the queue lengths and channel states cannot be made available.

V. CONCLUSIONS

In summary, our findings suggest that while the distributed PPO provides 7–8% worse demand dissatisfaction rate as compared to its idealized centralized implementation, it reaches a local minimum faster by avoiding the overheads and scalability issues associated with centralized solutions. Moreover, if the feedback for the centralized PPO is delayed, a loss in the demand dissatisfaction rate reduces to 3–4%. In addition, the average delay performance of the distributed approach is two times better as compared to the centralized solutions, while the difference in terms of the average throughput is insignificant. These trade-offs confirm the applicability of the employed algorithms, which can be considered when designing mmWave IAB networks. For example, system operators may utilize the distributed formulation to satisfy the traffic flow requirements more efficiently. Subsequently, the centralized approach may be useful to further enhance the performance after the distributed algorithm has reached its local minimum.

ACKNOWLEDGMENT

This work was supported by Intel Corporation and by the Research Council of Finland (Projects RADIANT, ECO-NEWS, SOLID, and ALL-ON).

REFERENCES

- [1] B. Tezergil and E. Onur, "Wireless backhaul in 5G and beyond: Issues, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2579–2632, 2022.
- [2] Y. Sadovaya, D. Moltchanov, W. Mao, O. Orhan, S.-p. Yeh, H. Nikopour, S. Talwar, and S. Andreev, "Integrated access and backhaul in millimeter-wave cellular: Benefits and challenges," *IEEE Communications Magazine*, vol. 60, no. 9, pp. 81–86, 2022.
- [3] C.-K. Wen, L.-S. Tsai, A. Shojaeifard, P.-K. Liao, K.-K. Wong, and C.-B. Chae, "Shaping a smarter electromagnetic landscape: IAB, NCR, and RIS in 5G standard and future 6G," *IEEE Communications Standards Magazine*, vol. 8, no. 1, pp. 72–78, 2024.
- [4] P. Fabian, G. Z. Papadopoulos, P. Savelli, and B. Cousin, "Performance evaluation of integrated access and backhaul in 5G networks," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2021, pp. 88–93.
- [5] A. Toskala and H. Holma, "5G-advanced overview," *5G Technology: 3GPP Evolution to 5G-Advanced*, pp. 485–503, 2024.
- [6] F. Gómez-Cuba and M. Zorzi, "Optimal link scheduling in millimeter wave multi-hop networks with MU-MIMO radios," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1839–1854, 2019.
- [7] R. Singh and P. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 127–142, 2018.
- [8] L. Wang, B. Ai, Y. Niu, H. Jiang, S. Mao, Z. Zhong, and N. Wang, "Joint user association and transmission scheduling in integrated mmWave access and terahertz backhaul networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 15 930–15 940, 2023.
- [9] N. Yarkina, D. Moltchanov, and Y. Koucheryavy, "Counter waves link activation policy for latency control in in-band IAB systems," *IEEE Communications Letters*, vol. 27, no. 11, pp. 3108–3112, 2023.
- [10] M. M. Sande, M. C. Hlophe, and B. T. S. Maharaj, "A backhaul adaptation scheme for IAB networks using deep reinforcement learning with recursive discrete choice model," *IEEE Access*, vol. 11, pp. 14 181–14 201, 2023.
- [11] G. B. Stone, D. A. Talbert, and W. Eberle, "A survey of scalable reinforcement learning," *International Journal of Intelligent Computing Research*, vol. 13, pp. 1118–1124, 2022.
- [12] Q. Cheng, Z. Wei, and J. Yuan, "Deep reinforcement learning-based spectrum allocation and power management for IAB networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.
- [13] M. Pagin, T. Zugno, M. Polese, and M. Zorzi, "Resource management for 5G NR integrated access and backhaul: A semi-centralized approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 753–767, 2022.
- [14] S. Gopalam, S. V. Hanly, and P. Whiting, "Distributed and local scheduling algorithms for mmWave integrated access and backhaul," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1749–1764, 2022.
- [15] 3GPP, "Study on integrated access and backhaul (Release 16)," *3GPP TR 38.874 V16.0.0*, 2018.
- [16] M. Aleksandrowicz and J. Jaworek-Korjakowska, "Metrics for assessing generalization of deep reinforcement learning in parameterized environments," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 14, no. 1, pp. 45–61, 2023.
- [17] M. Gupta, A. Rao, E. Visotsky, A. Ghosh, and J. G. Andrews, "Learning link schedules in self-backhauled millimeter wave cellular networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8024–8038, 2020.