

Distributed Scheduling for Throughput Maximization under Deadline Constraint in Wireless Mesh Networks

Xin Wang
Shanghai Jiao Tong University
Shanghai, China
xin.wang.ece@sjtu.edu.cn

Xudong Wang
Shanghai Jiao Tong University
Shanghai, China
wxudong@ieee.org

ABSTRACT

This paper studies the distributed scheduling of traffic flows with arbitrary deadlines that arrive at their source nodes and are transmitted to different destination nodes via multiple intermediate nodes in a wireless mesh network. When a flow is successfully delivered to its destination, a reward will be obtained, which is the embodiment of network performance and can be expressed by metrics such as throughput or network utility. The objective is to maximize the aggregate reward of all the deadline-constrained flows, which can be transformed into the constrained Markov decision process (CMDP). According to the transformation, a policy gradient-based distributed scheduling (PGDS) method is first proposed, where a primary reward and an auxiliary reward are designed to incentivize each node to independently schedule network resources such as power and subcarriers. The primary reward is generated when flows are successfully delivered to their destinations. The auxiliary reward, designed based on potential-based reward shaping (PBRS) using local information of data transmission, aims to accelerate the convergence speed. Inside this method, a reward feedback scheme is designed to let each node obtain the primary reward. Noting that each node selecting resources independently may cause interference and collision which leads to instability of data transmission, a policy gradient-based resource determination algorithm is proposed. Moreover, the optimality and convergence of the PGDS method are derived. Especially, when a policy obtained by the algorithm is not matched with the optimal policy but can better deal with the interference, an asymptotic optimum still exists and is further derived. The theoretical results are further validated through simulations. Taking throughput as a reward, simulations also show that the PGDS method improves the throughput by at least 70% and reduces the end-to-end delay.

CCS CONCEPTS

• **Networks** → **Network algorithms**; • **Theory of computation** → **Scheduling algorithms**.

KEYWORDS

Distributed scheduling, wireless mesh networks, deadline, reward

1 INTRODUCTION

Scheduling real-time traffic flows in wireless mesh networks has attracted wide attention due to emerging applications [3, 10, 34–36], e.g. in Internet of Things (IoT), autonomous driving, and cyber-physical systems, where time-sensitive packets need to be delivered from their sources to destinations across the whole networks [1, 22, 28]. When these packets are successfully transmitted to their destinations within the deadline, rewards will be generated. The

reward is an embodiment of the network performance and can be expressed in terms of throughput or network utility. The deadline means each flow should be delivered to its destination node on time before a given time requirement, otherwise it will be out-of-date and discarded. Unlike single-hop networks, where a central node can coordinate the traffic flows and perform scheduling methods [3, 4, 8, 9, 14, 17, 24], wireless mesh networks are difficult to centrally manage due to the large distance between source-destination pairs. At the same time, the time-frequency resources in the network are limited, which makes the simultaneous transmission of data flows by non-adjacent links interfere with each other and lead to transmission failure. Furthermore, different deadline-constrained flows fiercely compete for resources to achieve on-time delivery and thus be coupled with each other, which greatly increases the difficulty of distributed scheduling. Nowadays, some works of literature have been trying to solve these problems but with little success [15, 26, 31, 32], such as high-complexity scheduling methods with large overhead and cannot achieve the global optimum.

This paper makes important progress on distributedly scheduling deadline-constrained traffic flows in wireless mesh networks. The objective is to maximize the aggregate reward of flows that reach their destinations within their deadlines. To this end, this paper first formulates the reward maximization problem from a single-packet scheduling perspective considering interference in the network and establishes a relationship between scheduling network resources of packets at each node and the constrained Markov decision process (CMDP). A policy gradient-based distributed scheduling (PGDS) method based on the relationship is then designed, which quickly adapts to changes in network resources and converges to the global optimum. In this method, a primary reward is generated after a packet is successfully delivered to the destination node before its deadline, which is fed back to all intermediate nodes on the end-to-end path and used to guide the scheduling decision. Considering the sparsity of the reward generated by the destination node, the lag of feedback information will increase the convergence time. To improve the convergence speed, an auxiliary reward is designed using potential-based reward shaping (PBRS). To manage the interference, a resource determination algorithm is proposed, where each node maintains and updates the interference field and avoids collision resources to improve the stability of the PGDS method. To fully understand the performance of the PGDS method in a theoretical way and clearly know the influence of various parameters in the network, the optimality and convergence properties are derived theoretically as $O(1/T)$ which is verified to be fast enough to cope with channel variations by simulations. Finally, taking throughput as a reward, simulation results verify the performance of PGDS under throughput, delay, multiple network

topologies, and different channel conditions, and then the maximum throughput are improved by at least 70% compared with the existing methods. The main contributions of this paper are summarized as follows.

- For wireless mesh networks, the reward maximization model is theoretically established, which is then decomposed to propose a PGDS method. The relationship between the PGDS method and CMDP is analyzed in detail, and the method can be effectively used to schedule flows in wireless networks in a distributed manner.
- A PGDS method with PBRS is proposed. In this method, a reward feedback scheme is proposed to guide the scheduling on each node, an auxiliary reward based on PBRS is designed to accelerate the convergence, and the resource determination algorithm is proposed to manage the interference.
- The optimality and convergence are derived for the proposed PGDS method. These theoretical performances are verified and compared with existing methods by simulations.

The paper is structured as follows. In Section 2, the related work is discussed. In Section 3, the network model and problem formulation are performed. In Section 4, problem decomposition and the relationship with CMDP are analyzed. In Section 5, a PGDS method with PBRS is proposed. In Section 6, the performance of optimality and convergence are derived. In Section 7, some important simulation results are shown to illustrate the performance achieved by the proposed method, and the paper is concluded in Section 8.

2 RELATED WORK

There is a rich literature on the scheduling of traffic flows with deadline constraints in wireless networks to obtain maximal aggregate rewards, which contains scheduling methods in wireless single-hop networks, wired multi-hop networks, and wireless mesh or multi-hop networks. The reward can be network utility functions or throughput.

In wireless single-hop networks, where each node can communicate directly with a central node, a common approach is to use the node to coordinate the information of all flows and optimally schedule them under their deadline constraints to obtain maximal utility [3, 8, 14, 17, 24]. An error rate-based prediction model to improve the deadline-constrained throughput is proposed in [17], a bidding game algorithm between a central node and its neighbors to obtain maximal utility is designed in [3, 14], and some centralized methods based on Lyapunov optimization [8, 24]. However, these methods are difficult to extend to wireless mesh networks, since it is hard to find a central node that can collect all the information of flows from the whole network.

In wired multi-hop networks, flows are generated at source nodes and transmitted to their destination nodes via multiple intermediate nodes. If the flow exceeds its deadline, it is invalidated and discarded. The scheduling methods mainly contain heuristic algorithms without theoretical guarantees [5, 18], dynamic programming [20], a polynomial-time approximation algorithm based on the relaxation of the network utility maximization (NUM) problem [21], and distributed scheduling based on CMDP [31]. Since

these methods do not consider interference, they can not be directly extended to wireless mesh networks. However, a recent work in [31], which developed a connection with CMDP to design distributed scheduling to achieve throughput optimization, inspires us to devise PGDS method. It proposed a distributed link selection method for wired multi-hop networks with deadline-constrained flows, where intermediate nodes use a received reward from destination nodes to decide the next-hop transmission link.

In wireless mesh networks, the problem becomes more complicated considering interference. If the wireless channel is assumed to be stable, some centralized scheduling methods that can achieve maximal performance with deadline constraints can be used [11, 18, 19, 27, 41]. Note that these methods require large communication overhead to gather necessary information from all the nodes for a central node to obtain scheduling results. Distributed scheduling methods are preferred since they allow each node to make local scheduling decisions which is simpler and more efficient. These methods are mainly divided into three categories. First, the deadline is used to adjust the scheduling algorithm but not used as a constraint so that the delay can not be guaranteed, such as carrier sense multiple access (CSMA) scheduling algorithms with delay-based backoff value [13, 15, 26, 33], back pressure-type controllers adjusted by deadline [42], opportunistic scheduling with delay differentiation [39]. Second, the deadline is relaxed to the average delay constraint, such as a duality-based algorithm [12] and a maximum weight scheduling [40]. Thirdly, the deadline is constrained but scheduling cannot reach the maximal performance, such as opportunistic scheduling [23] and adaptive CSMA [32]. Therefore, devising a novel distributed scheduling method is an open and challenging problem.

3 NETWORK MODEL AND PROBLEM FORMULATION

3.1 Network Model

A general wireless mesh network is considered in this paper, where N nodes can generate, transmit, and receive flows. The topology of the network is denoted by a unidirectional graph $G = \{\mathcal{V}, \mathcal{L}\}$, where \mathcal{V} is the set of nodes in the network and \mathcal{L} is the set of links between adjacent nodes. Neighboring nodes communicate with each other through wireless links. There is no central node in the network that can collect information about all nodes and centrally schedule resources. Considering that orthogonal frequency division multiplexing (OFDM) is widely used in 6G and beyond Wireless networks, the subcarrier and time slot that can be scheduled are defined as the network resources in this paper. Therefore, nodes at each time slot can perform the scheduling of subcarriers to transmit flows. Each node is in half-duplex mode, that is, it can only receive or transmit flows in the same time slot. F real-time flows are generated at their source nodes and transmitted to their destination nodes through multiple intermediate nodes with each flow having end-to-end deadline constraint τ_f . The weight of flow f is w_f , which is determined by its priority. While transmitting flows in any time slot, each node needs to select subcarriers and transmission power for flows at that node. The total subcarriers in the network is C and the maximal transmission power at each node is P . The channel gain from node i to node j is h_{ij} ($0 \leq h_{ij} \leq 1$).

Wireless interference can prevent non-adjacent links from transmitting flows at the same time, which increases the difficulty of scheduling. For any node, it interferes with those nodes that can receive its signal power. Interference field of node i is $\mathcal{I}_i(t)$, which is defined as $\mathcal{I}_i(t) = \{j | j \in \mathcal{V}, P_i h_{ij}^2 > P_{\text{Th}}\}$. P_i is the transmission power at node i and P_{Th} is the detection threshold of interference. To avoid transmission failure due to interference, all links that are affected by node i cannot use more than C subcarriers.

3.2 Reward Maximization Problem

Note that any flow f is composed of packets, and each packet has a deadline τ_f . Distributed scheduling for any packet of flow f is shown in Fig. 1. A primary reward $u(t)$ is generated after the packet is successfully delivered to the destination node within the deadline. $u(t)$ is a function of the delivered packet, such as throughput or utility functions. Define a unit packet to be denoted by σ and the set of all unit packets of flow f to be denoted by $\mathcal{P}_f(t)$ ($\sigma \in \mathcal{P}_f(t)$). Without loss of generality, all flows in a network can be represented as a combination of multiple unit packets. The wireless mesh network considered in this paper is a time-slotted system, the number of subcarriers and transmission power need to be selected for each unit packet at each time slot. When a unit packet is transmitted from the source node to the destination node, its position (the node i where it is located) and time-till-deadline (TTD) τ are constantly changing. These two varying parameters are used to define the state $s_\sigma(t)$ of the packet. $s_\sigma(t) \in \mathcal{S}$ and \mathcal{S} is denoted as

$$\mathcal{S} = \{(i, \tau) | i \in \mathcal{V}, 0 \leq \tau \leq \tau_f\}. \quad (1)$$

Scheduling the packet includes determining its next transmission

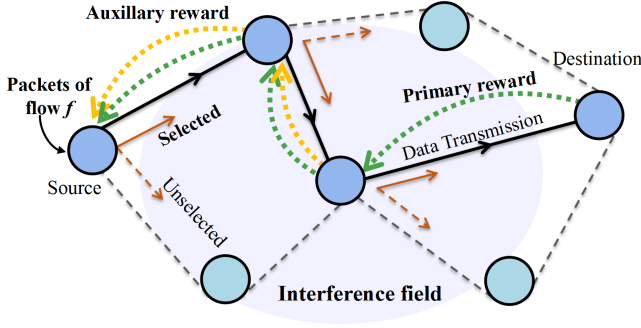


Figure 1: Distributed scheduling for packets of flow f .

node j and the number n_c and power n_p of subcarriers, which is denoted as $a_\sigma(t)$. $a_\sigma(t) \in \mathcal{A}$ and \mathcal{A} is given by

$$\mathcal{A} = \{(j, n_c, n_p) | j \in \mathcal{V}_i, 0 \leq n_c \leq C, 0 \leq n_p \leq P\}, \quad (2)$$

where \mathcal{V}_i denotes the set of neighbor nodes of node i , which is also a subset of \mathcal{V} . A scheduling policy $\pi[a_\sigma(t) | s_\sigma(t)]$ is defined as the probability of choosing schedule $a_\sigma(t)$ in state $s_\sigma(t)$. The policy is equal to 1 means definitely selects $a_\sigma(t)$. When the packet is successfully delivered to the destination node, the reward is generated, which is expressed as

$$r_D[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] = \begin{cases} u(t), & j \text{ is destination and } \tau \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The objective is to maximize the aggregate rewards of all the packets in the network. However, $r_D(\cdot)$ can only be obtained at the destination, which does not reflect the impact of the scheduling of packets in intermediate nodes and makes it difficult to determine the scheduling of intermediate nodes. Therefore, an auxiliary reward $\mathcal{H}[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]$ needs to be designed to reflect the global impact of the scheduling of intermediate nodes, and the new reward can be expressed as

$$r[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] = r_D[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] + \mathcal{H}[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]. \quad (4)$$

Using the new reward, the expected cumulative rewards of the packet with state $s_\sigma(t)$ is

$$R_r^\pi[s_\sigma(t)] = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\tau_f-1} r[s_\sigma(t+k), a_\sigma(t+k), s_\sigma(t+k+1)] \right\}. \quad (5)$$

Distinguished from the auxiliary reward, $r_D(\cdot)$ is called the primary reward. It should be emphasized that the role of the auxiliary reward is to facilitate the scheduling of packets by intermediate nodes, and it must not influence the scheduling policy π that maximizes the primary reward. This means that the scheduling policy obtained using the new reward is the same as the optimal scheduling policy obtained with the primary reward, that is, the policy-invariant auxiliary reward needs to be designed.

Moreover, constraints on the number and power of subcarriers need to be established. Define $c[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]$ is the number of subcarriers and $p[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]$ is the power selected from state $s_\sigma(t)$ to $s_\sigma(t+1)$, which are equal to n_c and n_p in $a_\sigma(t)$ in (2), respectively. The expected number of subcarriers under the policy π is

$$D_c^\pi[s_\sigma(t)] = \mathbb{E}_\pi \{c[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]\}. \quad (6)$$

The expected power under the policy π is

$$D_p^\pi[s_\sigma(t)] = \mathbb{E}_\pi \{p[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]\}. \quad (7)$$

Considering the constraint on the number of subcarriers and the transmit power of each node, the problem of maximizing the aggregate new reward of all the packets is formulated as

$$\max_{\pi} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f R_r^\pi[s_\sigma(t)], \quad (8)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i}(t) \cup \mathcal{P}_{\mathcal{I}_i}(t)} D_c^\pi[s_\sigma(t)] \leq C, \forall i \in \mathcal{V}, \quad (9)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} D_p^\pi[s_\sigma(t)] \leq P, \forall i \in \mathcal{V}, \quad (10)$$

where \mathcal{P}_{f_i} represents the set of packets of flow f at node i and $\mathcal{P}_{\mathcal{I}_i}$ represents the set of packets in the interference field of node i .

4 PROBLEM DECOMPOSITION

Note that the subcarrier in (8) is a discrete variable and the power is a continuous variable, the objective function can be either discrete

or continuous, thus this is a mixed integer optimization problem (MIOP). To solve the MIOP in a distributed way, the relationship between different states needs to be clarified first. Take the transmission of a unit packet of flow f as an example. In wireless mesh networks, a unit packet at state $s_\sigma(t) = (i, \tau)$ selects a scheduling way $a_\sigma(t) = (j, n_c, n_p)$ and wants to transmit to another state $s_\sigma(t+1) = (j, \tau - 1)$. This packet is not necessarily able to transmit to node j , but has a transition probability $\Pr(s_\sigma(t+1)|s_\sigma(t), a_\sigma(t))$. This probability is related to the channel characteristics of the current transmission link and the transmission technology of the physical layer, such as the coding and modulation methods. In the paper, the probability is expressed as $\Pr(SINR_{ij} \geq SINR_{Th})Pr_e(SINR_{ij})$, where $SINR_{ij}$ means the signal-to-interference-noise ratio (SINR) at node j and is denoted as

$$SINR_{ij} = \frac{p[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] h_{ij}^2}{N_0 p[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] \Delta_c + P_{I_i}}, \quad (11)$$

where N_0 is the power spectral density of additional white Gaussian noise (AWGN), Δ_c is the subcarrier space, and P_{I_i} is the interference power from interference field I_i . $P_{I_i} = \sum_{\sigma \in \mathcal{P}_{I_i}} p[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] h_{kj}^2$ where k is the node in the interference field I_i . P_e means the packet error rate under specific coding and modulation methods. Note that this transition probability is only related to the current link and has nothing to do with other links that the packet passes through, that is

$$\begin{aligned} \Pr[s_\sigma(t+1)|s_\sigma(0), a_\sigma(0), s_\sigma(1), a_\sigma(1), \dots, s_\sigma(t), a_\sigma(t)] \\ = \Pr[s_\sigma(t+1)|s_\sigma(t), a_\sigma(t)]. \end{aligned} \quad (12)$$

Therefore, the process of transmitting the packet from state $s_\sigma(t)$ to the next state $s_\sigma(t+1)$ is independent of the historical state, that is, it conforms to the Markov property.

It is stated above that the MIOP in (8) can be viewed as the constrained Markov decision process (CMDP), and then the Lagrangian decomposition can be used. The detailed derivation process is in Appendix A in [38] and the main result after the decomposition is

$$\mathfrak{L}(\pi, \lambda, \mu) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t)], \quad (13)$$

where the constant term is ignored, λ and μ are the Lagrange multipliers, and $R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t)]$ is to replace $r(\cdot)$ in (5) with $r_L(\cdot)$. $r_L(\cdot)$ is the reward under the Lagrange multipliers and is derived as

$$r_L^\sigma(t) = w_f r^\sigma(t) - \lambda_i c^\sigma(t) - \mu_i p^\sigma(t) - \frac{\lambda_i}{N_i} \sum_{\sigma' \in \mathcal{P}_{I_i}(t)} c^{\sigma'}(t), \quad (14)$$

where $r^\sigma(t)$, $c^\sigma(t)$, and $p^\sigma(t)$ are simplified representations of $r(\cdot)$ in (5), $c(\cdot)$ in (6), and $p(\cdot)$ in (7). N_i is the number of packets at node i . It's worth noting that $r_L^\sigma(t)$ can be viewed as two parts. One part is the reward obtained for successful transmission, which is the first three terms on the right-hand side of (22). The other part is the effect brought by the transmission of other packets in the interference field, which is the fourth term on the right side of (22). Recall the Lagrangian in (13), it decomposes the packet scheduling to each node, that is, the packets of each node can obtain and

constrain its own reward through the interference field, and then these rewards can be used to achieve distributed scheduling.

For the above-transformed problems, CMDP provides some standard algorithmic frameworks for solving them, such as value-based and policy-based algorithms [2, 6, 30]. However, these algorithms cannot be directly used to solve the transformed problem, and the two main challenges are as follows. One is that the feasible region is changing. Since the interference range changes with scheduling and channel gain, the number of available subcarriers is constantly changing. Therefore, when the scheduling policy is updated, the feasible region also needs to be updated synchronously. The other is the choice of scheduling policy. The collision occurs when the total number of subcarriers selected by packets in the interference field exceeds C . In the actual scheduling, it is necessary to avoid this collision situation, so as to reduce the probability of transmission errors and improve the stability of the network. Note that the distributed scheduling framework using value-based algorithms proposed in wired multi-hop networks inspires the design in this paper [31]. However, this framework cannot be applied to wireless multi-hop networks since it does not take into account the effects of wireless interference. Considering that the value-based algorithm may converge to a local optimum and it is difficult to find a global optimal policy when the action space and state space are large, policy-based algorithm framework is adopted in this paper. Therefore, this paper proposes a policy gradient-based distributed scheduling (PGDS) method and details are designed in section 5.

5 SCHEME AND ALGORITHMS IN PGDS METHOD WITH PBRS

In the PGDS method, when a packet reaches a state, it selects the next-hop link, number of subcarriers, and power of the next-hop transmission according to Algorithm 1. The packet then listens for the selection of other states in the interference field and determines whether the transmission needs to be performed. When the packet is successfully delivered to the destination node, the destination node will generate a reward. The reward is then fed back to each node along the end-to-end path and used to update scheduling policies according to a reward feedback scheme in section 5.1. When future packets arrive at these nodes, the number of subcarriers and power can then be selected based on the scheduling policy. The auxiliary reward, which is based on PBRS and can guarantee the policy optimality is derived in section 5.2. In summary, based on this method, any packet at any node in the network can decide the resource to transmit in the next hop according to the scheduling policy, so as to realize distributed scheduling. The PGDS framework consists of three key parts: reward feedback scheme, auxiliary reward design, and policy gradient-based resource determination algorithm. The main design methods for these parts are described in detail below.

5.1 Reward Feedback Scheme

The key to performing the PGDS method is that the rewards generated by the destination node need to affect all nodes along the end-to-end path. This effect can be described by the derivation of

$R_{r_L}^{\pi, \lambda, \mu}$ in (13), which is denoted by

$$R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t)] = \sum_{a_\sigma(t) \in \mathcal{A}_i} \pi[a_\sigma(t)|s_\sigma(t)] \{ \Pr(\cdot) [r_L^\sigma(t) + R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t+1)]] + [1 - \Pr(\cdot)] R_{r_L}^{\pi, \lambda, \mu}[s'_\sigma(t+1)] \}, \quad (15)$$

where $\Pr(\cdot) = \Pr[s_\sigma(t+1)|s_\sigma(t), a_\sigma(t)]$, $s_\sigma(t) = (i, \tau)$, $s_\sigma(t+1) = (j, \tau - 1)$, and $s'_\sigma(t) = (i, \tau - 1)$. The above derivation shows that in order to know the global impact of scheduling subcarriers and power in a state, it is necessary to know the expected cumulative reward in the next state, and so on until the end of the destination node. This means that the destination node needs to feed back the generated reward so that every intermediate node knows the global impact of its schedule.

The reward feedback scheme is designed to realize the above process as shown in Fig. 2. The scheme includes the establishment of a feedback table and the backtracking of rewards. When packets are transmitted from the source node to the destination node, feedback tables are created on each node. The feedback table records the current state, previous state, and next state of the packet shown in Fig. 2. Since the transfer between states has Markov property, these three states can effectively help the reward backtracking. More specifically, the transmission of packets between different nodes corresponds to the process of state transfer, and only packets belonging to the same flow can be transferred to each other.

When the packet is successfully delivered to the destination node, the reward is generated according to (22). The reward is then fed back to each node along the end-to-end path and the basic process is described below. The reward needs to carry information about which flow it belongs to and the state of the last hop. For example, the reward at node j carries information: flow f , state $(j, \tau - 1)$. When node i receives a reward from node j and needs to find its next feedback node in Fig. 2, node i needs to check the feedback table. According to the information carried by the reward, node i first matches the "Next state" to this information. Since the time of one-hop transmission is a unit time slot, node i can infer the "Current state" of the reward should be (i, τ) . After matching the "Next state" and "Current state" in the feedback table, node i can finally find the "Previous state". The identifier of the node in the previous state is the next feedback node for the reward. In order not to make the size of the feedback table too large, when the reward determines the node that needs feedback, the information belonging to the feedback path needs to be cleaned out at the node in time. Therefore, according to the above reward feedback process, the reward generated by the destination node can affect every node on the end-to-end path.

Note that this section designs the process of reward feedback from the perspective of scheme design. In practice, it is necessary to design a specific signaling transmission protocol to ensure the effective operation of the scheme, which will be studied in the future.

5.2 Auxiliary Reward Design based on PBRS

Since only the destination node can generate a reward, it takes some time for this reward to feedback to the intermediate node. During this period, the intermediate nodes lack the information to guide the scheduling, which makes the convergence speed slow. To

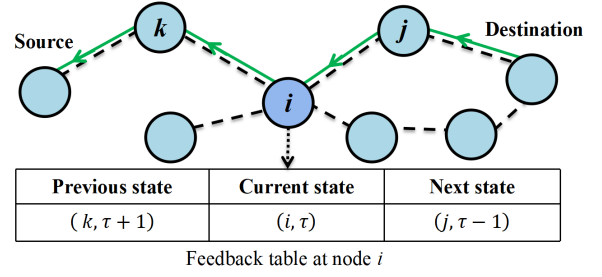


Figure 2: Reward feedback at node i .

speed up the convergence, this paper designs an auxiliary reward to assist the reward generated at the destination to complete fast scheduling. Note that the auxiliary reward should not affect the optimality of the scheduling policy, and only the information that can be collected locally can be exploited. Potential-based reward shaping (PBRS) has been proven to improve the convergence without changing the optimality, but how to design an effective reward shaping function needs to combine with the specific network situation. First, the representation of an auxiliary reward needs to be designed. According to PBRS, it has the following result [25, 37].

Lemma 1 Recall that the new reward in (4), the policy-invariant auxiliary reward should be formatted as

$$\mathcal{H}[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] = \phi[s_\sigma(t+1)] - \phi[s_\sigma(t)], \quad (16)$$

where $\phi(\cdot)$ is a potential function and belongs to \mathbb{R} .

Using the format of auxiliary reward in Lemma 1, the optimal policy will not be changed. Then, we need to determine the metric that will be used for designing the potential function. Noting that the primary reward in (4) produced by the destination node is a function of the successfully delivered packet, the auxiliary reward should also have a metric related to it. The number of packets received by the destination node is related to the number of packets transmitted by the intermediate node. The fewer the number of packets transmitted by the intermediate node, the fewer the number of packets successfully delivered to the destination node. From this perspective, the number of packets transmitted by the intermediate node is the bottleneck that limits the number of packets delivered by the destination node. Therefore, the link that transmits a large number of packets should be selected as far as possible, that is, the number of packets in the same state should be used as a metric. This metric is denoted as $u_L[s_\sigma(t)]$. In addition, to deliver the packet as soon as possible, it is necessary to select the intermediate node as close to the destination node as possible during scheduling, so the minimum number of hops from the intermediate node to the destination node should be used as a metric. This metric is denoted as $d_{\sigma i}$ where i is the same as that in $s_\sigma(t)$. To ensure the consistency of convergence, the potential function needs to have a consistent form with the primary reward, and the potential function also needs to have a tendency to change to the global optimum, that is, the maximum potential is at the destination node and the minimum potential is at the source node. Finally, the potential function is designed as

Theorem 1 Suppose the format of an auxiliary reward in Lemma 1, the auxiliary reward in wireless mesh networks is designed as

$$\mathcal{H}[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] = \frac{u_L[s_\sigma(t+1)]}{d_{\sigma j}} - \frac{u_L[s_\sigma(t)]}{d_{\sigma i}},$$

where j is the same as that in $s_\sigma(t+1)$.

$d_{\sigma i}$ and $d_{\sigma j}$ can be easily obtained by a simple distributed signaling interaction protocol in [29]. The proof of policy invariance with this potential function is derived in Appendix B in [38].

5.3 Determination of Resources

When one node receives feedback information using the reward feedback scheme in section 5.1, it can calculate the scheduling policy π and select the number and power of subcarriers. Note that the scheduling space is composed of the number and power of subcarriers and the next-hop transmission node. The power can be divided into different levels and it is a discrete value. The number of subcarriers may be large in wideband communication scenarios, which means that the size of the scheduling space may be large. However, most of the policy-based methods proposed to solve a CMDP problem are not yet applicable to such a large space. Policy gradient-based methods have the ability to deal with such a large scheduling space, and have the theoretical global convergence guarantee. Therefore, a policy gradient-based framework is used to design the resource determination algorithm in Algorithm 1. In this framework, a parameterized policy π_θ ($\theta \in \mathbb{R}$) is defined as $\pi_\theta(a|s) = \frac{\exp(\theta_{sa})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{sa'})}$, which is differentiable and tractable. The policy gradient of the Lagrangian in (13) is derived in Appendix C in [38] and the main result is

$$\nabla_\theta \mathcal{L}(\pi, \lambda, \mu) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} \nabla_\theta \log \pi_\theta[a_\sigma(t)|s_\sigma(t)] \left\{ D_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t), a_\sigma(t)] - R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t)] \right\}, \quad (17)$$

where $D_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t), a_\sigma(t)]$ is the expected cumulative reward obtained under the given $s_\sigma(t)$ and $a_\sigma(t)$. The parameterized gradient is given by

$$\theta_{sa}(t+1) = \theta_{sa}(t) + \eta_1 \nabla_\theta \mathcal{L}(\pi, \lambda, \mu)|_{s_\sigma(t), a_\sigma(t)}, \quad (18)$$

where η_1 is a step size. Consider the relationship between $\pi_\theta(a|s)$ and θ_{sa} , the policy gradient is obtained and shown in Algorithm 1. The algorithm is used at any node i , n_σ denotes the bit number of the unit packet, $(\cdot)_+$ denotes the projection onto a positive real set \mathbb{R}_+ , Δ_t denotes the unit time slot, and $s_\sigma(t) = (i, \tau)$.

Note that when the number of subcarriers scheduled by all states within the interference range exceeds C , the collision of packet transmission will occur, which leads to packet transmission failure. Since there is no reward for this packet transmission failure, it is actually equivalent to the packet giving up the transmission at the current time slot, so Algorithm 1 uses the capacity unavailable condition in line 13 to make these packets give up the transmission in advance. When these packets finish scheduling resources independently, they will broadcast their selection result and the reward value in the history to all the nodes in the interference field. If a packet finds that the sum of all subcarriers in its interference range

Algorithm 1: Policy Gradient-based Resource Determination.

Input: $R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t-1)]$, policy $\pi_{\theta_{sa}}(t-1)$, n_σ .

Output: $R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t)]$, $\pi_{\theta_{sa}}(t)$, $\tilde{c}[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]$, and $\tilde{p}[s_\sigma(t), a_\sigma(t), s_\sigma(t+1)]$.

```

1 for  $s_\sigma(t) \in \mathcal{S}_i$  and  $t > 0$  do
2   Calculate  $\nabla_\theta \mathcal{L}(\pi, \lambda, \mu)|_{s_\sigma(t-1), a_\sigma(t-1)}$ .
3   Policy gradient update:
      
$$\pi_{\theta_{sa}}(t) = \frac{\pi_{\theta_{sa}}(t-1) e^{\eta_1 \nabla_\theta \mathcal{L}(\pi, \lambda, \mu)|_{s_\sigma(t-1), a_\sigma(t-1)}}}{\sum_{a \in \mathcal{A}_i} \pi_{\theta_{sa}}(t) e^{\eta_1 \nabla_\theta \mathcal{L}(\pi, \lambda, \mu)|_{s_\sigma(t-1), a}}}$$

4   Scheduling  $a_\sigma(t)$  with the probability  $\pi_{\theta_{sa}}(t)$ : obtain  $c^\sigma(t)$  and  $p^\sigma(t)$ .
5   Calculate the number of bits that can be transmitted under the channel gain  $h_{ij}$ :
      
$$\tilde{n} = c^\sigma(t) \Delta_c \log_2 \left( 1 + \frac{p^\sigma(t) h_{ij}^2}{N_0 c^\sigma(t) \Delta_c} \right) \Delta_t.$$

6   while  $|\tilde{n} - n_\sigma| > n_\epsilon$  or  $\tilde{n} \neq 0$  do
7     Repeat the above probabilistic scheduling and calculate  $\tilde{n}$ .
8     if the number of repetitions reaches the limit then
9        $c^\sigma(t) = 0$  and  $p^\sigma(t) = 0$ .
10    end
11  end
12 end
13 Calculate total number of subcarriers selected in node  $i$  and  $\mathcal{I}_i$ ; /* Capacity unavailable condition */
      
$$C_i = \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i}(t) \cup \mathcal{P}_{\mathcal{I}_i}(t)} c^\sigma(t)$$

15 if  $C_i > C$  then
16   Remove the scheduling with the smallest  $R_{r_L}^{\pi, \lambda, \mu}[s_\sigma(t-1)]$  until  $C_i \leq C$ .
17   Final resource for each state is  $\tilde{c}^\sigma(t)$  and  $\tilde{p}^\sigma(t)$ .
18 end
19 Using the selected resources to update Lagrangian multipliers:
      
$$\lambda_i(t+1) = \left( \lambda_i(t) + \eta_2 \left\{ C - \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i}(t) \cup \mathcal{P}_{\mathcal{I}_i}(t)} D_c^\pi[s_\sigma(t)] \right\} \right)_+,$$

20
      
$$\mu_i(t+1) = \left( \mu_i(t) + \eta_3 \left\{ P - \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} D_p^\pi[s_\sigma(t)] \right\} \right)_+.$$


```

exceeds C , it will compare the reward values, and let the scheduling with the minimum reward value give up the transmission in this time slot, so as to ensure no interference. The above resource

determination process needs to use a distributed signaling interaction protocol, which can be quickly implemented by some existing methods [7, 16], such as nucleus-based methods, which can allocate the subcarriers required by each packet to orthogonal frequencies. The non-interference transmission will make some packets give up the transmission in advance, the state of these packets is also converted at the current node, and no reward will be obtained. Therefore, the same effect as the interference-based transmission will not affect the convergence of the distributed method to the optimal scheduling policy. However, this will lead to a change in scheduling policy and bring some errors, and detailed analysis and derivation are shown in Appendix E in [38].

6 OPTIMALITY AND CONVERGENCE

This section derives the optimality and convergence properties of the PGDS method. Considering that the algorithm in section 5.3 changes the scheduling results under capacity unavailable condition, it may lead to an error from the optimal scheduling. Therefore, this section analyzes the optimality without the capacity unavailable condition first, and then considers the error caused by changing the scheduling results to further derive the optimality.

6.1 Without Capacity Unavailable Condition

Define the optimal policy as π^* and the optimal expected cumulative reward for each packet is $R_{r\sigma}^*$. The relationship between the optimality of the PGDS method and the network parameters is crucial, as this facilitates the dynamic management of the network. The network parameters considered in this paper include the number of nodes N , the number of flows F , the number of subcarriers C , and the maximum deadline constraint τ_{\max} . The global convergence of PGDS method is derived in Appendix D in [38], and the main result is shown below.

Theorem 2 Without capacity unavailable condition, each state will directly perform the selected number and power of subcarriers. The optimality gap between the objective and optimal aggregate reward R_r^* is

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{R_r^\pi [s_\sigma(t)] - R_{r\sigma}^*\} \leq \frac{g(N, F, C, \tau_{\max})}{T},$$

where g is the function of the network parameters shown in Appendix D in [38], and it also has a limited value.

Since g is limited, if the iteration time T is large enough, it is obtained that

$$\lim_{T \rightarrow \infty} \frac{g(N, F, C, \tau_{\max})}{T} = 0. \quad (19)$$

Therefore, the maximal aggregate reward will be obtained, and the convergence rate is $\mathcal{O}(\frac{1}{T})$.

6.2 With Capacity Unavailable Condition

In wireless mesh networks, capacity unavailable condition in Algorithm 1 is used to avoid collisions. As analyzed in section 5.3, the

collision of packet transmissions will lead to an error with the optimal scheduling result. Define δ_c and δ_p are errors between Algorithm 1 and optimal scheduling without capacity unavailable condition. Detailed expressions are provided in Appendix E in [38], and the main result is shown below.

Theorem 3 Without capacity unavailable condition, δ_c and δ_p occur, and the optimality gap between the objective and optimal aggregate reward R_r^* is

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{R_r^\pi [s_\sigma(t)] - R_r^*\} \leq \frac{g(N, F, C, \tau_{\max}) + q(\delta_c, \delta_p)}{T}. \quad (20)$$

where q is the function of the errors shown in Appendix E in [38], and it also has a limited value. It is worth noting that optimality can still be achieved in the presence of errors.

7 PERFORMANCE EVALUATIONS

In this section, the optimality and convergence of the PGDS method are first verified, then is to compare the performance difference between this approach and the current best-performing distributed scheduling methods.

7.1 Verification of Optimality and Convergence

To clearly show the performance of the network, the reward function takes the form of throughput. Take the mesh topology with 16 nodes and 4 flows as an example. The relationship between the throughput and different arrival rates is shown in Fig. 3(a). Arrival rates 1,2,3, and 4 represent the sum rate of all the flows are 1, 2, 3, and 4 unit packet/slot, respectively. Each flow divides all the sum arrival rates equally. It can be seen that when the arrival rate is small, the convergence rate is faster, and throughput is proportional to the arrival rate. When the arrival rate is large, throughput tends to be saturated, and is limited by the maximum link capacity, that is, the maximum transmit power per node, the number of subcarriers, and the channel gain. In addition, the figure can also reflect the rapid convergence, even under saturated network utility, it only takes around 100 units of time slot to converge.

After that, the performance with and without auxiliary reward is shown in Fig 3(b). At different arrival rates, the auxiliary reward can significantly speed up the convergence. Note that the effect of the auxiliary reward on improving the convergence speed does not increase much as the arrival rate increases. The main reason is that the auxiliary reward is only effective when the primary reward does not return in the initial stage. Regardless of the arrival rate, the effect of the auxiliary reward will continue to decrease as soon as the primary reward starts to generate.

Next, the convergence time under different numbers of nodes in the wireless network is carried out in Fig. 3(c). Flows are generated from the node in the lower left corner and need to be delivered to the destination node in the upper right corner. Since all the paths of the data flow to the destination node can run through all nodes, it can effectively reflect the influence of the number of network nodes. It can be seen that when the number of nodes increases, the convergence time increases similarly to the log function form, which shows the correctness of the conclusion in Theorem 2. Note

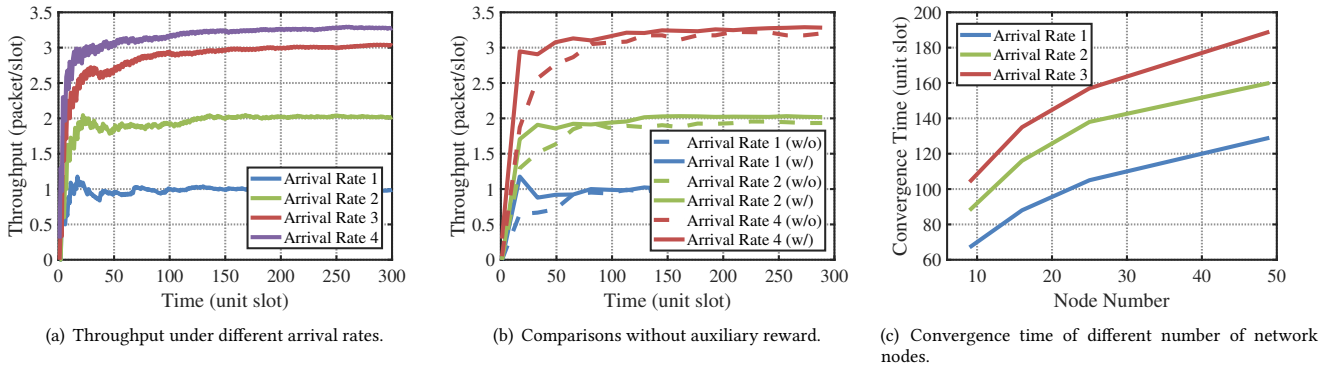


Figure 3: Verification of optimality and convergence of the PGDS method.

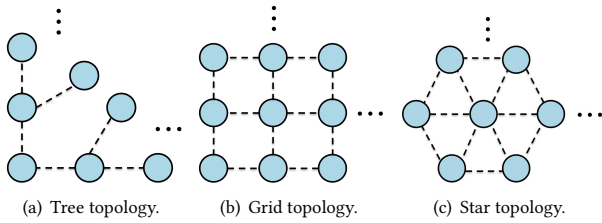


Figure 4: Three typical topologies of wireless mesh networks.

that the increase in the number of flows is similar to the increase in the arrival rate, so increasing the arrival rate can reflect the impact of increasing the number of data flows to a certain extent. As the arrival rate increases, the convergence time increases less than the linear increase, in a manner similar to the log function.

7.2 Performance under Different Topologies

Tree, grid, and star topologies are classic wireless mesh network topologies, which are considered in the simulation and shown in Fig. 4. The difference between the three topologies lies in the number of neighbors of each node. A larger number of neighbors means a larger scheduling space, which will lead to an increase in the convergence time. A time-slotted system is used, where the unit time slot is set as 0.25 ms and the length of the unit packet is 100 bits. The total number of subcarriers in the network is 20, the maximum transmit power of each node is 20 dBm. Links adopt the channel model of the 3GPP standard. The channel changes for a time greater than twice the maximum end-to-end delay. Each flow has its own arrival rate. The maximal deadline for flows to reach their destination is assumed to be 10 unit time slots. The source and destination nodes of data flow need to cover the edge nodes of the network, so as to reflect the impact of the increase of network size.

The performance of the PGDS method in different topologies is shown in Fig. 5. The result of throughput is shown in Fig. 5(a). Since the tree topology has the smallest action space, it has a fast convergence speed. With the help of auxiliary rewards, the action Spaces

of grid and star topologies are also reduced a lot, but the interference range becomes larger in these two topologies, which leads to a decrease in the convergence speed. The cumulative distribution function (CDF) of the delay under the three topologies is shown in Fig. 5(b). It can be observed that when the arrival rate increases, more packets will stay in the intermediate nodes due to the limited number of available bandwidth in the interference field, increasing the delay. After that, the delay violation probability is simulated in Fig. 5(c), where the horizontal coordinate represents the sum of the arrival rates of all data flows. It can be seen that when the sum of all data flow arrival rates does not exceed the maximum link capacity, PGDS method can effectively ensure the delay, that is, the delay violation probability is about 0. As the arrival rate increases, more and more packets are stuck at the source node or intermediate node. The TTD of these packets will continue to decrease until it drops to 0, that is, the packet is invalid. Therefore, packets are continuously discarded, which leads to an increase in the delay violation probability. The increase in delay violation probability means that the source node needs to reduce its data generation rate, which belongs to the field of congestion control and is not considered in this paper.

7.3 Comparisons with Existing Methods

Finally, the performance of the PGDS method is compared with that of existing methods. To illustrate the improved performance of the proposed method, the simulation takes traditional CSMA as a baseline and compares it with the most recent best-performing distributed scheduling method, that is adaptive CSMA in [32]. In the traditional CSMA method, the node listens to the channel at each unit slot, and performs backoff if the channel is idle, so that the counter time is reduced by 1 unit slot, and the data is transmitted when the counter time is 0. If a conflict occurs, double the counter time. In the Adaptive CSMA method, each node has a counter for its connected links, and the time generation of the counter follows the exponential distribution of the mean of a parameter as a unit time slot, and the adjustment of this parameter is in [32]. The performance comparison of throughput is shown in Fig. 6(a), where PGDS method outperforms the existing methods. The reason is analyzed as follows. Since CSMA only transmits data according to the

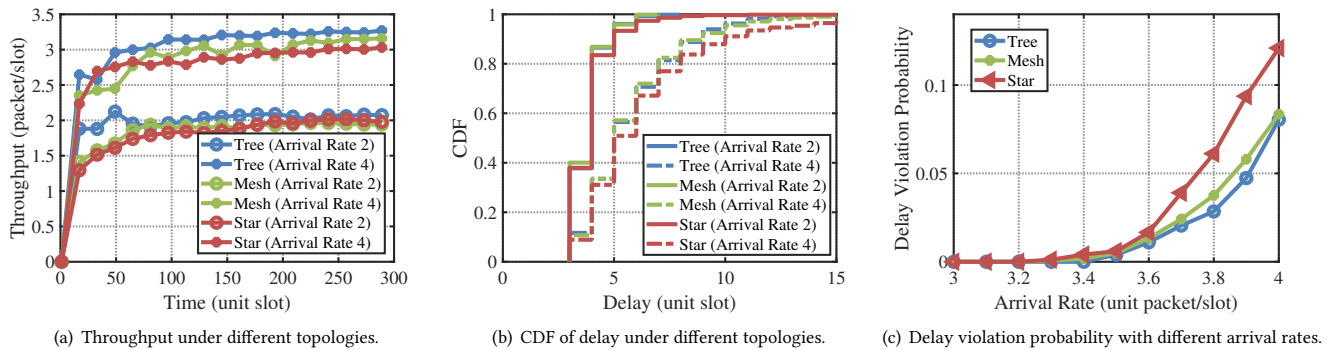


Figure 5: Throughput and delay performance under different topologies using the PGDS method.

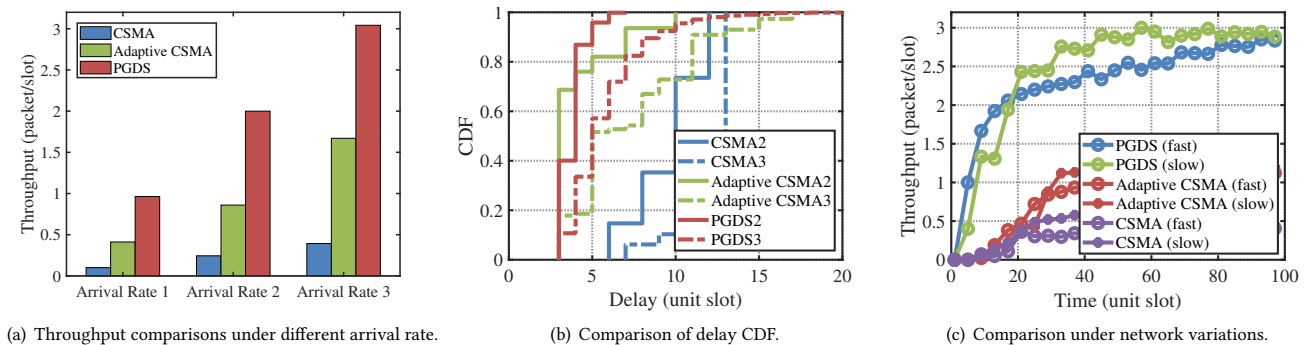


Figure 6: Comparisons of throughput and delay with existing methods.

parameter of the timer, this parameter does not reflect the performance of transmitting the data flow from source to destination, so the performance is the worst. Adaptive CSMA can adjust the timer size through the interaction of local nodes, but cannot obtain end-to-end instructions, so a large number of packets are lost when the delay constraint is very tight, resulting in low throughput. The CDF of the delay under different arrival rates is shown in Fig. 6(b). In CSMA, a large number of data packets are stuck in intermediate nodes, or fall into some kind of cycle and can not reach the destination node, resulting in a long delay. Adaptive CSMA can reduce this effect by adjusting the window with local information, to reduce the delay. However, only using local information cannot know the end-to-end transmission situation, so this problem cannot be fundamentally solved. The proposed PGDS method can find the best transmission link according to the end-to-end feedback information, so it can effectively guarantee the end-to-end delay. Furthermore, the effect of network variation is compared in Fig. 6(c). In the simulation, the average channel quality is the same for each link. "fast" means that the duration of the channel quality variation on each link is less than the maximum end-to-end delay, and "slow" means that the channel quality variation on each link changes larger than twice the maximum end-to-end delay. It can be seen that the rapid change of the channel will slow down the convergence speed, the main reason is that the feedback reward

information has a lag, which is no longer applicable to the current channel. However, since the statistical characteristics of the channel are not changed, it is still able to converge to the optimal value in the end. Note that the PGDS method does not consider signaling overhead. If signaling overhead is 20%, then the performance gain will decrease 20%, which will be studied in the future.

8 CONCLUSION

In this paper, a policy gradient-based distributed scheduling (PGDS) method with potential-based reward shaping (PBRS) for wireless networks with end-to-end deadline constraints was proposed. The method can achieve maximal throughput for wireless mesh networks and outperform the existing methods. Theoretically, the optimality and convergence rate of the method were derived, and their relationship to network parameters was analyzed in detail. Following the proposed distributed method, an interesting problem for future research is how to apply the method to wireless mobile networks. How to integrate other technologies, such as beamforming, to further enhance the network performance is also an important topic to study.

REFERENCES

- [1] Vahid Khalilpour Akram and Onur Ugurlu. 2023. Detecting the Most Vital Articulation Points in Wireless Multi-Hop Networks. *IEEE/ACM Transactions on*

- Networking* 31, 5 (2023), 2389–2402.
- [2] Eitan Altman. 1999. *Constrained Markov Decision Processes* (1st ed.). Routledge, New York, USA.
 - [3] I-Fen Chao and Wei-Sheng Hsu. 2023. A MAC Protocol Design for Maximizing End-to-End Throughput and Fairness Guarantee in Chain-Based Multi-Hop Wireless Backhaul Networks. *IEEE Transactions on Mobile Computing* 22, 7 (2023), 3743–3756.
 - [4] Kun Chen and Longbo Huang. 2018. Timely-Throughput Optimal Scheduling With Prediction. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2457–2470.
 - [5] Han Deng, Tao Zhao, and I-Hong Hou. 2019. Online Routing and Scheduling With Capacity Redundancy for Timely Delivery Guarantees in Multihop Networks. *IEEE/ACM Transactions on Networking* 27, 3 (2019), 1258–1271.
 - [6] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. 2020. Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 8378–8390.
 - [7] Sahar Hoteit, Stefano Secchi, Rami Langar, and Guy Pujolle. 2013. A Nucleolus-Based Approach for Resource Allocation in OFDMA Wireless Mesh Networks. *IEEE Transactions on Mobile Computing* 12, 11 (2013), 2145–2154.
 - [8] I-Hong Hou and P. R. Kumar. 2010. Utility Maximization for Delay Constrained QoS in Wireless. In *2010 Proceedings IEEE INFOCOM*. 1–9.
 - [9] I-Hong Hou and P. R. Kumar. 2010. Utility-optimal scheduling in time-varying wireless networks with delay constraints. In *Proceedings of the Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Chicago, Illinois, USA) (*MobiHoc '10*). Association for Computing Machinery, New York, NY, USA, 31–40.
 - [10] Hung-Yun Hsieh and Raghupathy Sivakumar. 2001. Performance comparison of cellular and multi-hop wireless networks: a quantitative study. In *Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (Cambridge, Massachusetts, USA) (*SIGMETRICS '01*). Association for Computing Machinery, New York, NY, USA, 113–122.
 - [11] Cheng Huang, Xin Wang, and Xudong Wang. 2022. Effective-Capacity-Based Resource Allocation for End-to-End Multi-Connectivity in 5G IAB Networks. *IEEE Transactions on Wireless Communications* 21, 8 (2022), 6302–6316.
 - [12] Soroush Jahromizadeh and Veselin Rakocevic. 2014. Joint Rate Control and Scheduling for Providing Bounded Delay With High Efficiency in Multihop Wireless Networks. *IEEE/ACM Transactions on Networking* 22, 5 (2014), 1686–1698.
 - [13] Hyeryung Jang, Se-Young Yun, Jinwoo Shin, and Yung Yi. 2014. Distributed learning for utility maximization over CSMA-based wireless multihop networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 280–288.
 - [14] Jiequ Ji, Xiangyu Ren, Lin Cai, and Kun Zhu. 2023. Downlink Scheduler for Delay Guaranteed Services Using Deep Reinforcement Learning. *IEEE Transactions on Mobile Computing* (2023), 1–14.
 - [15] Libin Jiang and Jean Walrand. 2010. A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks. *IEEE/ACM Transactions on Networking* 18, 3 (2010), 960–972.
 - [16] Ji-Young Jung and Jung-Ryun Lee. 2021. Distributed Fair Resource Allocation Algorithm in Wireless Multihop Network Systems. *IEEE Systems Journal* 15, 4 (2021), 4713–4724.
 - [17] Igor Kadota, Abhishek Sinha, and Eytan Modiano. 2018. Optimizing Age of Information in Wireless Networks with Throughput Constraints. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 1844–1852.
 - [18] Ruogu Li and Atilla Eryilmaz. 2012. Scheduling for End-to-End Deadline-Constrained Traffic With Reliability Requirements in Multihop Networks. *IEEE/ACM Transactions on Networking* 20, 5 (2012), 1649–1662.
 - [19] Kang Liu, Wei Quan, Nan Cheng, Xue Zhang, Liang Guo, Deyun Gao, and Hongke Zhang. 2023. Deadline-Constrained Multi-Agent Collaborative Transmission for Delay-Sensitive Applications. *IEEE Transactions on Cognitive Communications and Networking* 9, 5 (2023), 1370–1384.
 - [20] Qingyu Liu, Haibo Zeng, and Minghua Chen. 2020. Network Utility Maximization Under Maximum Delay Constraints and Throughput Requirements. *IEEE/ACM Transactions on Networking* 28, 5 (2020), 2132–2145.
 - [21] Xin Liu, Weichang Wang, and Lei Ying. 2019. Spatial-temporal routing for supporting end-to-end hard deadlines in multi-hop networks. *Performance Evaluation* 135 (2019), 102007.
 - [22] Jiadong Lou, Xu Yuan, Purushottam Sigdel, Xiaoqi Qin, Sastry Kompella, and Nian-Feng Tzeng. 2023. Age of Information Optimization in Multi-Channel Based Multi-Hop Wireless Networks. *IEEE Transactions on Mobile Computing* 22, 10 (2023), 5719–5732.
 - [23] Michael J. Neely. 2011. Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks. In *2011 Proceedings IEEE INFOCOM*. 1728–1736.
 - [24] Michael J. Neely. 2013. Delay-Based Network Utility Maximization. *IEEE/ACM Transactions on Networking* 21, 1 (2013), 41–54.
 - [25] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (Washington, DC, USA) (*ICML '99*). AAAI Press, 278–287.
 - [26] Jian Ni, Bo Tan, and R. Srikant. 2012. Q-CSMA: Queue-Length-Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks. *IEEE/ACM Transactions on Networking* 20, 3 (jun 2012), 825–836.
 - [27] Matteo Pagin, Tommaso Zugno, Michele Polese, and Michele Zorzi. 2022. Resource Management for 5G NR Integrated Access and Backhaul: A Semi-Centralized Approach. *IEEE Transactions on Wireless Communications* 21, 2 (2022), 753–767.
 - [28] Claudio E. Palazzi. 2018. High Bandwidth and Low Delay over Wireless Multihop Networks (*MobiHoc '18*). Association for Computing Machinery, New York, NY, USA, 306–307.
 - [29] R. D. Poor. 2000. Gradient Routing in Ad Hoc Networks. *Massachusetts Institute of Technology* (2000).
 - [30] Harsh Satija, Philip Amortila, and Joelle Pineau. 2020. Constrained Markov Decision Processes via Backward Value Functions. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 8502–8511.
 - [31] Rahul Singh and P. R. Kumar. 2019. Throughput Optimal Decentralized Scheduling of Multihop Networks With End-to-End Deadline Constraints: Unreliable Links. *IEEE Trans. Automat. Control* 64, 1 (2019), 127–142.
 - [32] Rahul Singh and P. R. Kumar. 2021. Adaptive CSMA for Decentralized Scheduling of Multi-Hop Networks With End-to-End Deadline Constraints. *IEEE/ACM Transactions on Networking* 29, 3 (2021), 1224–1237.
 - [33] Peruru Subrahmanya Swamy, Radha Krishna Ganti, and Krishna Jagannathan. 2017. Adaptive CSMA Under the SINR Model: Efficient Approximation Algorithms for Throughput and Utility Maximization. *IEEE/ACM Transactions on Networking* 25, 4 (2017), 1968–1981.
 - [34] Vishrant Tripathi, Rajat Talak, and Eytan Modiano. 2023. Information Freshness in Multihop Wireless Networks. *IEEE/ACM Transactions on Networking* 31, 2 (2023), 784–799.
 - [35] Christos Tsanikidis and Javad Ghaderi. 2022. Online Scheduling and Routing with End-to-End Deadline Constraints in Multihop Wireless Networks (*MobiHoc '22*). Association for Computing Machinery, New York, NY, USA, 11–20.
 - [36] Christos Tsanikidis and Javad Ghaderi. 2023. Near-Optimal Packet Scheduling in Multihop Networks with End-to-End Deadline Constraints. 7, 3, Article 50 (dec 2023), 32 pages.
 - [37] Eric Wiewiora, Garrison Cottrell, and Charles Elkan. 2003. Principled Methods for Advising Reinforcement Learning Agents. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (Washington, DC, USA) (*ICML '03*). AAAI Press, 792–799.
 - [38] Wang Xin and Xudong Wang. [n. d.]. Distributed Scheduling for Throughput Maximization under Deadline Constraint in Wireless Mesh Networks. *accessible at <https://wanglab.sjtu.edu.cn/userfiles/files/Article.pdf>* ([n. d.]).
 - [39] Haozhi Xiong, Ruogu Li, Atilla Eryilmaz, and Eylem Ekici. 2011. Delay-Aware Cross-Layer Design for Network Utility Maximization in Multi-Hop Networks. *IEEE Journal on Selected Areas in Communications* 29, 5 (2011), 951–959.
 - [40] Dongyue Xue and Eylem Ekici. 2013. Delay-Guaranteed Cross-Layer Scheduling in Multihop Wireless Networks. *IEEE/ACM Transactions on Networking* 21, 6 (2013), 1696–1707.
 - [41] Shizhen Zhao and Xiaojun Lin. 2016. Design of Scheduling Algorithms for End-to-End Backlog Minimization in Wireless Multi-Hop Networks Under K -Hop Interference Models. *IEEE/ACM Transactions on Networking* 24, 2 (2016), 1265–1278.
 - [42] Anfu Zhou, Min Liu, Zhongcheng Li, and Eryk Dutkiewicz. 2012. Cross-Layer Design for Proportional Delay Differentiation and Network Utility Maximization in Multi-Hop Wireless Networks. *IEEE Transactions on Wireless Communications* 11, 4 (2012), 1446–1455.

A PROBLEM DECOMPOSITION

Recall the problem formulation in (8), the Lagrangian decomposition is denoted as

$$\begin{aligned} \mathfrak{Q}(\pi, \lambda, \mu) &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f R_r^\pi [s_\sigma(t)] - \\ &\sum_{i \in \mathcal{V}} \lambda_i \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i(t)} \cup \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)] - \\ &\sum_{i \in \mathcal{V}} \mu_i \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} D_p^\pi [s_\sigma(t)] + \sum_{i \in \mathcal{V}} \lambda_i C + \sum_{i \in \mathcal{V}} \mu_i P, \end{aligned}$$

where λ_i and μ_i are the Lagrangian multipliers. Ignore the constant value, it can be further derived as

$$\begin{aligned} \mathfrak{Q}(\pi, \lambda, \mu) &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \left\{ \sum_{\sigma \in \mathcal{P}_f(t)} w_f R_r^\pi [s_\sigma(t)] - \right. \\ &\left. \sum_{i \in \mathcal{V}} \lambda_i \sum_{\sigma \in \mathcal{P}_{f_i(t)} \cup \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)] - \sum_{i \in \mathcal{V}} \mu_i \sum_{\sigma \in \mathcal{P}_{f_i}(t)} D_p^\pi [s_\sigma(t)] \right\} \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \left\{ \sum_{\sigma \in \mathcal{P}_{f_i}(t)} w_f R_r^\pi [s_\sigma(t)] - \right. \\ &\left. \lambda_i \sum_{\sigma \in \mathcal{P}_{f_i(t)} \cup \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)] - \mu_i \sum_{\sigma \in \mathcal{P}_{f_i}(t)} D_p^\pi [s_\sigma(t)] \right\} \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \left\{ \sum_{\sigma \in \mathcal{P}_{f_i}(t)} [w_f R_r^\pi [s_\sigma(t)] - \right. \\ &\left. \lambda_i D_c^\pi [s_\sigma(t)] - \mu_i D_p^\pi [s_\sigma(t)]] - \lambda_i \sum_{\sigma \in \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)] \right\}. \end{aligned}$$

Note that the last term is not related to σ in $\mathcal{P}_{f_i}(t)$. Define the number of packet at node i is N_i , then we can get

$$\sum_{\sigma \in \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)] = \frac{1}{N_i} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} \sum_{\sigma \in \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)]. \quad (21)$$

Therefore, the Lagrangian is derived as

$$\begin{aligned} \mathfrak{Q}(\pi, \lambda, \mu) &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} [w_f R_r^\pi [s_\sigma(t)] - \\ &\lambda_i D_c^\pi [s_\sigma(t)] - \mu_i D_p^\pi [s_\sigma(t)] - \frac{\lambda_i}{N_i} \sum_{\sigma \in \mathcal{P}_{I_i}(t)} D_c^\pi [s_\sigma(t)]] \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} R_{r_L}^{\pi, \lambda, \mu} [s_\sigma(t)], \end{aligned}$$

where $R_{r_L}^{\pi, \lambda, \mu} [s_\sigma(t)]$ is to replace $r(\cdot)$ in (5) with $r_L(\cdot)$. $r_L(\cdot)$ is the reward under the Lagrange multipliers and is given by

$$r_L^\sigma(t) = w_f r^\sigma(t) - \lambda_i c^\sigma(t) - \mu_i p^\sigma(t) - \frac{\lambda_i}{N_i} \sum_{\sigma' \in \mathcal{P}_{I_i}(t)} c^{\sigma'}(t), \quad (22)$$

where $r^\sigma(t)$, $c^\sigma(t)$, and $p^\sigma(t)$ are simplified representations of $r(\cdot)$ in (5), $c(\cdot)$ in (6), and $p(\cdot)$ in (7).

B PROOF OF THEOREM 1

Without considering the auxiliary reward, the cumulative reward of any state $s_\sigma(t)$ is $R_{r_D}^\pi [s_\sigma(t)]$ and

$$R_{r_D}^\pi [s_\sigma(t)] = \sum_{a_\sigma(t) \in \mathcal{A}} \pi [a_\sigma(t) | s_\sigma(t)] D_{r_D}^\pi [s_\sigma(t), a_\sigma(t)]. \quad (23)$$

Under the optimal policy, $D_{r_D}^* [s_\sigma(t), a_\sigma(t)]$ should satisfy the Bellman equation, that is

$$\begin{aligned} D_{r_D}^* [s_\sigma(t), a_\sigma(t)] &= \mathbb{E}_\pi \{ r_D [s_\sigma(t), a_\sigma(t), s_\sigma(t+1)] + \\ &\max_{a_\sigma(t+1) \in \mathcal{A}} D_{r_D}^* [s_\sigma(t+1), a_\sigma(t+1)] \}. \end{aligned}$$

Suppose the potential function is $\frac{u_L[s_\sigma(t)]}{d_{\sigma i}}$. For the convenience of expression, let $s = s_\sigma(t) = (i, \tau)$ and $s' = s_\sigma(t+1) = (j, \tau-1)$. It can be derived that

$$\begin{aligned} D_{r_D}^* [s, a] - \frac{u_L[s]}{d_{\sigma i}} &= \mathbb{E}_\pi \left\{ r_D [s, a, s'] + \max_{a' \in \mathcal{A}} D_{r_D}^* [s', a'] \right\} - \frac{u_L[s]}{d_{\sigma i}} \\ &= \mathbb{E}_\pi \left\{ r_D [s, a, s'] + \frac{u_L[s']}{d_{\sigma j}} + \max_{a' \in \mathcal{A}} D_{r_D}^* [s', a'] - \frac{u_L[s']}{d_{\sigma j}} \right\} - \frac{u_L[s]}{d_{\sigma i}} \\ &= \mathbb{E}_\pi \left\{ r_D [s, a, s'] + \frac{u_L[s']}{d_{\sigma j}} - \frac{u_L[s]}{d_{\sigma i}} + \max_{a' \in \mathcal{A}} D_{r_D}^* [s', a'] - \frac{u_L[s']}{d_{\sigma j}} \right\} \\ &= \mathbb{E}_\pi \left\{ r [s, a, s'] + \max_{a' \in \mathcal{A}} D_{r_D}^* [s', a'] - \frac{u_L[s']}{d_{\sigma j}} \right\}. \end{aligned}$$

Let $D_r^\pi [s, a] = D_{r_D}^* [s, a] - \frac{u_L[s]}{d_{\sigma i}}$, then we can get

$$D_r^\pi [s, a] = \mathbb{E}_\pi \left\{ r [s, a, s'] + \max_{a' \in \mathcal{A}} D_r^\pi [s', a'] \right\}. \quad (24)$$

Therefore, for any state, $D_r^\pi [s, a]$ still satisfies the Bellman equation then it is also optimal. Finally, the optimal policy is obtained as that without auxiliary reward.

C POLICY GRADIENT DERIVATION

Recall that the Lagrangian is given by

$$\mathfrak{Q}(\pi, \lambda, \mu) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} R_{r_L}^{\pi, \lambda, \mu} [s_\sigma(t)]. \quad (25)$$

A parameterized policy π_θ is used to obtain the gradient, such as $\pi_\theta(a|s) = \frac{\exp(\theta_{sa})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{sa'})}$, which is differentiable and tractable. Utilizing the policy gradient theorem, the gradient for the objective

function is

$$\begin{aligned} & \nabla_{\theta} \mathfrak{L}(\pi, \lambda, \mu) \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{V}} \sum_{\sigma \in \mathcal{P}_{f_i}(t)} \nabla_{\theta} \log \pi_{\theta}[a_{\sigma}(t) | s_{\sigma}(t)] \\ & \quad D_{r_L}^{\pi, \lambda, \mu}[s_{\sigma}(t), a_{\sigma}(t)], \end{aligned}$$

where

$$\begin{aligned} & D_{r_L}^{\pi, \lambda, \mu}[s_{\sigma}(t), a_{\sigma}(t)] \\ &= \mathbb{E}_{\pi} \left\{ \sum_{k=0}^{\tau_f-1} r_L^{\pi, \lambda, \mu}[s_{\sigma}(t+k), a_{\sigma}(t+k), s_{\sigma}(t+k+1)] \middle| s_{\sigma}(t), a_{\sigma}(t) \right\}. \end{aligned} \quad (26)$$

Then, a general policy gradient updating method is $\theta_{sa}(t+1) = \theta_{sa}(t) + \eta_1 \nabla_{\theta} \mathfrak{L}(\pi, \lambda, \mu) |_{s,a}$. Nowadays, the natural policy gradient (NPG) method is popular and has a faster convergence rate than other policy gradient methods, in which the policy is updated as

$$\theta_{sa}(t+1) = \theta_{sa}(t) + \eta_1 \mathcal{F}^{\dagger}[\theta_{sa}(t)] \nabla_{\theta} \mathfrak{L}(\pi, \lambda, \mu) |_{s,a}, \quad (27)$$

where $\mathcal{F}(\theta)$ is the Fisher information matrix and given by

$$\mathcal{F}(\theta) = \mathbb{E}_{\pi} [\nabla_{\theta} \log \pi_{\theta}(a|s) (\nabla_{\theta} \log \pi_{\theta}(a|s))^{\top}]. \quad (28)$$

\mathcal{F}^{\dagger} takes the Moore-Penrose inverse of matrix \mathcal{F} . Using the compatible function approximation error theorem, the policy gradient is derived as

$$\begin{aligned} & \mathcal{F}^{\dagger}[\theta_{sa}(t)] \nabla_{\theta} \mathfrak{L}(\pi, \lambda, \mu) |_{s,a} \\ &= D_{r_L}^{\pi, \lambda, \mu}[s_{\sigma}(t) = s, a_{\sigma}(t) = a] - R_{r_L}^{\pi, \lambda, \mu}[s_{\sigma}(t) = s]. \end{aligned} \quad (29)$$

By using the exponential relationship between policy π and θ , the update of the policy can be obtained as

$$\pi_{\theta_{sa}}(t) = \frac{\pi_{\theta_{sa}}(t-1) e^{\eta_1 \nabla_{\theta} \mathfrak{L}(\pi, \lambda, \mu) |_{s_{\sigma}(t-1), a_{\sigma}(t-1)}}}{\sum_{a \in \mathcal{A}_i} \pi_{\theta_{sa}}(t) e^{\eta_1 \nabla_{\theta} \mathfrak{L}(\pi, \lambda, \mu) |_{s_{\sigma}(t-1), a}}}. \quad (30)$$

D PROOF OF THEOREM 2

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{R_r^* [s_{\sigma}(t)] - R_r^{\pi} [s_{\sigma}(t)]\} \leq \frac{g(N, F, C, \tau_{\max})}{T},$$

Recall the objective function is

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f R_r^{\pi} [s_{\sigma}(t)].$$

The optimal reward is $R_r^* [s_{\sigma}(t)]$. The optimality gap is

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{R_r^* [s_{\sigma}(t)] - R_r^{\pi} [s_{\sigma}(t)]\} \\ &= \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right], \end{aligned} \quad (31)$$

where $A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)]$ is derived by the performance difference theorem and is denoted by

$$A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] = D_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] - R_r^{\pi} [s_{\sigma}(t)]. \quad (32)$$

According to the PGDS method in Algorithm 1, it should be noted that

$$\pi_{\theta_{sa}}(t+1) = \pi_{\theta_{sa}}(t) \exp \left\{ \eta_1 A_{r_L}^{\pi, \lambda, \mu} [s_{\sigma}(t), a_{\sigma}(t)] \right\} Z^{-1} [s_{\sigma}(t)], \quad (33)$$

$$\begin{aligned} A_{r_L}^{\pi, \lambda, \mu} [s_{\sigma}(t), a_{\sigma}(t)] &= A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] + \lambda_i(t) A_c^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] + \\ & \quad \mu_i(t) A_p^{\pi} [s_{\sigma}(t), a_{\sigma}(t)], \end{aligned} \quad (34)$$

where Z is the normalized function, $A_c^{\pi} [s_{\sigma}(t), a_{\sigma}(t)]$ is the function related to the subcarriers and $A_p^{\pi} [s_{\sigma}(t), a_{\sigma}(t)]$ is the function related to the power constraint, which are defined in the same way as $A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)]$. Then, it is derived that

$$\begin{aligned} A_r^{\pi, \lambda, \mu} [s_{\sigma}(t), a_{\sigma}(t)] &= \frac{1}{\eta_1} \log \frac{\pi_{\theta_{sa}}(t+1) Z [s_{\sigma}(t)]}{\pi_{\theta_{sa}}(t)} - \\ & \quad \lambda_i(t) A_c^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] - \mu_i(t) A_p^{\pi} [s_{\sigma}(t), a_{\sigma}(t)]. \end{aligned} \quad (35)$$

Following the derivation in (31), it is obtained that

$$\begin{aligned} & \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right] \\ &= \frac{1}{\eta_1} \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] \log \frac{\pi_{\theta_{sa}}(t+1) Z [s_{\sigma}(t)]}{\pi_{\theta_{sa}}(t)} \right] - \\ & \quad \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \lambda_i(t) \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_c^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right] - \\ & \quad \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \mu_i(t) \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_p^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right] \end{aligned} \quad (36)$$

Using the performance difference theorem for two constraints, it can be obtained that

$$\begin{aligned} & \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \lambda_i(t) \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_c^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right] \\ &= \mathbb{E}_{\pi^*} [\lambda_i(t) c [s_{\sigma}(t), a_{\sigma}(t)]] - \mathbb{E}_{\pi} [\lambda_i(t) c [s_{\sigma}(t), a_{\sigma}(t)]], \\ & \quad \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \mu_i(t) \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_p^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right] \\ &= \mathbb{E}_{\pi^*} [\mu_i(t) p [s_{\sigma}(t), a_{\sigma}(t)]] - \mathbb{E}_{\pi} [\mu_i(t) p [s_{\sigma}(t), a_{\sigma}(t)]]. \end{aligned}$$

Then (31) is derived as

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] A_r^{\pi} [s_{\sigma}(t), a_{\sigma}(t)] \right] \\ &= \frac{1}{\eta_1 T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_{\sigma}(t) \in \mathcal{A}} \pi^* [a_{\sigma}(t) | s_{\sigma}(t)] \log \frac{\pi_{\theta_{sa}}(t+1) Z [s_{\sigma}(t)]}{\pi_{\theta_{sa}}(t)} \right] - \\ & \quad \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} [\lambda_i(t) c [s_{\sigma}(t), a_{\sigma}(t)]] - \mathbb{E}_{\pi} [\lambda_i(t) c [s_{\sigma}(t), a_{\sigma}(t)]] - \\ & \quad \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} [\mu_i(t) p [s_{\sigma}(t), a_{\sigma}(t)]] - \mathbb{E}_{\pi} [\mu_i(t) p [s_{\sigma}(t), a_{\sigma}(t)]] . \end{aligned} \quad (37)$$

The first term is derived as

$$\begin{aligned}
& \frac{1}{\eta_1 T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_\sigma(t) \in \mathcal{A}} \pi^*[a_\sigma(t)|s_\sigma(t)] \log \frac{\pi_{\theta_{sa}}(t+1)Z[s_\sigma(t)]}{\pi_{\theta_{sa}}(t)} \right] \\
&= \frac{1}{\eta_1 T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_\sigma(t) \in \mathcal{A}} \pi^*[a_\sigma(t)|s_\sigma(t)] \log \frac{\pi_{\theta_{sa}}(t+1)}{\pi_{\theta_{sa}}(t)} \right] + \\
& \frac{1}{\eta_1 T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_\sigma(t) \in \mathcal{A}} \pi^*[a_\sigma(t)|s_\sigma(t)] \log Z[s_\sigma(t)] \right] \\
&= \frac{1}{\eta_1 T} \mathbb{E}_{\pi^*} \left[\log \frac{\pi_{\theta_{sa}}(T)}{\pi_{\theta_{sa}}(0)} \right] + \frac{1}{\eta_1 T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} [\log Z[s_\sigma(t)]] \\
&\leq \frac{\log NFC\tau_{\max}}{\eta_1 T} \tag{38} \\
&+ \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [r[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [r[s_\sigma(t), a_\sigma(t)]] \} \\
&+ \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [\lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)]] - \\
& \mathbb{E}_{\pi^*} [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]] \} \\
&+ \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [\mu_i(t+1)p[s_\sigma(t+1), a_\sigma(t+1)]] - \\
& \mathbb{E}_{\pi^*} [\mu_i(t)p[s_\sigma(t), a_\sigma(t)]] \} \\
&\leq \frac{\log NFC\tau_{\max}}{\eta_1 T} + \frac{\bar{r}_T}{T} + \frac{\bar{\lambda}c + \bar{\mu}p}{T} - \frac{\mathbb{E}_\pi (\lambda_i^2(T)) - \epsilon_\lambda}{2\eta_2 T} - \frac{\mathbb{E}_\pi (\mu_i^2(T)) - \epsilon_\mu}{2\eta_3 T}.
\end{aligned}$$

where the fact that $\mathbb{E}(\log \frac{p(\pi)}{q(\pi)}) \leq \log |\pi|$ is used and $|\pi|$ means the size of π . The maximal size of π for flow f at node i is $C\tau_{\max}$, then the total policy for the whole network is $NFC\tau_{\max}$. The inequality in (38) uses the expansion of $\mathbb{E}_{\pi^*} [\log Z[s_\sigma(t)]]$, which is derived as

$$\begin{aligned}
& \frac{1}{\eta_1} \mathbb{E}_{\pi^*} [\log Z[s_\sigma(t)]] \leq \frac{1}{\eta_1} \mathbb{E}_{\pi^*} [\log Z[s_\sigma(t)]] + \\
& \frac{1}{\eta_1} \mathbb{E}_{\pi^*} \left[\sum_{a_\sigma(t) \in \mathcal{A}} \pi^*[a_\sigma(t)|s_\sigma(t)] \log \frac{\pi_{\theta_{sa}}(t+1)}{\pi_{\theta_{sa}}(t)} \right] \\
&= \frac{1}{\eta_1} \mathbb{E}_{\pi^*} \left[\sum_{a_\sigma(t) \in \mathcal{A}} \pi^*[a_\sigma(t)|s_\sigma(t)] \log \frac{\pi_{\theta_{sa}}(t+1)Z[s_\sigma(t)]}{\pi_{\theta_{sa}}(t)} \right] - \\
& \mathbb{E}_{\pi^*} [\lambda_i(t)A_c^\pi[s_\sigma(t), a_\sigma(t)]] - \mathbb{E}_{\pi^*} [\mu_i(t)A_p^\pi[s_\sigma(t), a_\sigma(t)]] + \\
& \mathbb{E}_{\pi^*} [\lambda_i(t)A_c^\pi[s_\sigma(t), a_\sigma(t)]] + \mathbb{E}_\pi [\mu_i(t)A_p^\pi[s_\sigma(t), a_\sigma(t)]] \\
&= \mathbb{E}_{\pi^*} [A_r^\pi[s_\sigma(t), a_\sigma(t)]] + \mathbb{E}_{\pi^*} [\lambda_i(t)A_c^\pi[s_\sigma(t), a_\sigma(t)]] + \\
& \mathbb{E}_{\pi^*} [\mu_i(t)A_p^\pi[s_\sigma(t), a_\sigma(t)]] \\
&= \mathbb{E}_{\pi^*} [r[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [r[s_\sigma(t), a_\sigma(t)]] \\
&+ \{ \mathbb{E}_{\pi^*} [\lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]] \} \\
&+ \{ \mathbb{E}_{\pi^*} [\mu_i(t+1)p[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [\mu_i(t)p[s_\sigma(t), a_\sigma(t)]] \}
\end{aligned}$$

The third term of the inequality in (38) is derived by

$$\begin{aligned}
& \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [\lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]] \} \\
&\tag{39} \\
&= \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \{ \lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)] - \lambda_i(t)c[s_\sigma(t), a_\sigma(t)] \} \\
&= \frac{1}{T} \mathbb{E}_{\pi^*} \left\{ \sum_{t=0}^{T-1} \lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)] - \lambda_i(t)c[s_\sigma(t), a_\sigma(t)] \right\} \\
&= \frac{1}{T} \mathbb{E}_{\pi^*} \{ \lambda_i(T)c[s_\sigma(T), a_\sigma(T)] - \lambda_i(0)c[s_\sigma(0), a_\sigma(0)] \} \\
&= \frac{\mathbb{E}_{\pi^*} \{ \lambda_i(T)c[s_\sigma(T), a_\sigma(T)] \}}{T}.
\end{aligned}$$

The fourth term of the inequality in (38) is similar to the third term. Following the same derivation process, it can be derived that

$$\begin{aligned}
& \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [\mu_i(t+1)p[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [\mu_i(t)p[s_\sigma(t), a_\sigma(t)]] \} \\
&= \frac{\mathbb{E}_{\pi^*} \{ \mu_i(T)p[s_\sigma(T), a_\sigma(T)] \}}{T}. \tag{40}
\end{aligned}$$

Before deriving the second term of (37), it should be noted that the update of Lagrangian multiplier λ is

$$\lambda_i(t+1) = (\lambda_i(t) + \eta_2 \mathbb{E}_\pi [C - c[s_\sigma(t), a_\sigma(t)]])_+, \tag{41}$$

where $()_+$ denotes the projection into an interval Λ_1 on the positive real set \mathbb{R}_+ and η_2 is the step size. Since the projection makes the range of λ smaller, it is obtained that $\lambda_i(t+1) - \lambda_i(t) \leq \eta_2 \mathbb{E}_{\pi^*} [C - c[s_\sigma(t), a_\sigma(t)]]$. Note that $\mathbb{E}_{\pi^*} [c[s_\sigma(t), a_\sigma(t)]] = C$, the second term of (37) is derived as

$$\begin{aligned}
& \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]] - \mathbb{E}_\pi [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]] \} \\
&\geq \frac{1}{\eta_2 T} \sum_{t=0}^{T-1} \mathbb{E}_\pi \{ \lambda_i(t) [\lambda_i(t+1) - \lambda_i(t)] \} \\
&= \frac{1}{\eta_2 T} \left[\frac{1}{2} \sum_{t=0}^{T-1} \mathbb{E}_\pi (\lambda_i^2(t+1) - \lambda_i^2(t)) - \frac{1}{2} \sum_{t=0}^{T-1} \mathbb{E}_\pi (\lambda_i(t+1) - \lambda_i(t))^2 \right] \\
&= \frac{1}{\eta_2 T} \left[\frac{1}{2} \mathbb{E}_\pi (\lambda_i^2(T)) - \frac{1}{2} \sum_{t=0}^{T-1} \mathbb{E}_\pi (\lambda_i(t+1) - \lambda_i(t))^2 \right] \\
&= \frac{\mathbb{E}_\pi (\lambda_i^2(T)) - \epsilon_\lambda}{2\eta_2 T}, \tag{42}
\end{aligned}$$

where ϵ is the quadratic difference of the iterative step. As the iterations go on, $\lambda_i(t+1)$ and $\lambda_i(t)$ tend to be equal, so ϵ_λ is a small enough finite value.

The third term of (37) has a similar format as the second term, so it can be derived that

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \{ \mathbb{E}_{\pi^*} [\mu_i(t) p[s_\sigma(t), a_\sigma(t)]] - \mathbb{E}_\pi [\mu_i(t) p[s_\sigma(t), a_\sigma(t)]] \} \\ \geq \frac{\mathbb{E}_\pi (\mu_i^2(T)) - \epsilon_\mu}{2\eta_3 T}. \end{aligned} \quad (43)$$

It should be noted that ϵ_μ is similar as ϵ_c . Finally, it can be obtained

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{ R_r^\pi [s_\sigma(t)] - R_{r\sigma}^* \} &\leq \frac{\log N F C \tau_{\max}}{\eta_1 T} + \frac{\bar{r}_T}{T} + \\ \frac{\mathbb{E}_{\pi^*} \{ \lambda_i(T) c[s_\sigma(T), a_\sigma(T)] + \mu_i(T) p[s_\sigma(T), a_\sigma(T)] \}}{T} &- \\ \frac{\mathbb{E}_\pi (\lambda_i^2(T)) - \epsilon_\lambda}{2\eta_2 T} - \frac{\mathbb{E}_\pi (\mu_i^2(T)) - \epsilon_\mu}{2\eta_3 T}. \end{aligned} \quad (44)$$

For simplicity, the third term of the above expressions on the right-hand side is denoted as $\frac{\bar{\lambda}\epsilon + \bar{\mu}\bar{\rho}}{T}$. To clearly reflect the influence of network parameters on optimality, the above formula can be expressed as

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{ R_r^\pi [s_\sigma(t)] - R_{r\sigma}^* \} \leq \frac{g(N, F, C, \tau_{\max})}{T} \quad (45)$$

Therefore, as T approaches infinity, the optimality gap goes to zero, that is

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{ R_r^\pi [s_\sigma(t)] - R_{r\sigma}^* \} = \mathcal{O} \left(\frac{1}{T} \right). \quad (46)$$

E PROOF OF THEOREM 3

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{ R_r^\pi [s_\sigma(t)] - R_{r\sigma}^* \} &\leq \\ \frac{g(N, F, C, \tau_{\max}) + q(\delta_c, \delta_p)}{T}. \end{aligned} \quad (47)$$

Recall the number and power of subcarriers with capacity unavailable condition is $\tilde{c}^\sigma(t)$ and $\tilde{p}^\sigma(t)$, the policy is denoted as $\tilde{\pi}$ and the parameterized policy is denoted as $\hat{\theta}$. The optimality gap between the optimal policy π^* and the policy π with capacity unavailable condition is

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}} \sum_{\sigma \in \mathcal{P}_f(t)} w_f \{ R_r^\pi [s_\sigma(t)] - R_{r\sigma}^* [s_\sigma(t)] \} \\ = \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a_\sigma(t) \in \mathcal{A}} \pi^* [a_\sigma(t) | s_\sigma(t)] A_r^\pi [s_\sigma(t), a_\sigma(t)] \right], \end{aligned} \quad (48)$$

For the convenience of derivation, using $s = s_\sigma(t) = (i, \tau)$ and $a = a_\sigma(t)$. After that, it can be derived that

$$\begin{aligned} \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] A_r^\pi [s, a] \right] &= \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] A_r^\pi [s, a] \right] + \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\lambda}_i(t) \pi^* [a | s] A_c^\pi [s, a] \right] &- \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\lambda}_i(t) \pi^* [a | s] A_c^\pi [s, a] \right] + \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\mu}_i(t) \pi^* [a | s] A_p^\pi [s, a] \right] &- \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\mu}_i(t) \pi^* [a | s] A_p^\pi [s, a] \right] + \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \log \pi(t+1) \right] &- \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \log \pi(t+1) \right] \\ = \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] A_{r_L}^\pi [s, a] \right] &- \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\lambda}_i(t) \pi^* [a | s] A_c^\pi [s, a] \right] - \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\mu}_i(t) \pi^* [a | s] A_p^\pi [s, a] \right] &+ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \log \pi(t+1) \right] - \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \log \pi(t+1) \right]. \end{aligned} \quad (49)$$

Consider the Taylor expansion of policy π , it can be obtained that

$$\begin{aligned} \log \pi_\theta(t+1) &= \log \pi_\theta(t) + \nabla_\theta \log \pi_\theta |_{\theta=\theta(t)} [\theta(t+1) - \theta(t)] + \\ &\gamma [\theta(t)] [\theta(t+1) - \theta(t)], \end{aligned} \quad (50)$$

where $\lim_{t \rightarrow \infty} \gamma [\theta(t)] = 0$. Therefore, we can derive that

$$\begin{aligned} \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] A_r^\pi [s, a] \right] &= \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] A_{r_L}^\pi [s, a] \right] - \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\lambda}_i(t) \pi^* [a | s] A_c^\pi [s, a] \right] &- \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \mu_i(t) \pi^* [a | s] A_p^\pi [s, a] \right] + \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \log \pi(t+1) \right] &- \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \log \pi(t) \right] - \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \nabla_\theta \log \pi_\theta |_{\theta=\theta(t)} [\theta(t+1) - \theta(t)] \right] &- \\ \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^* [a | s] \gamma [\theta(t)] [\theta(t+1) - \theta(t)] \right]. \end{aligned}$$

Merging partial terms of them can be obtained that

$$\begin{aligned} & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] A_r^\pi[s, a] \right] = \\ & \mathbb{E}_{\pi^*} \left\{ \sum_{a \in \mathcal{A}} \pi^*[a|s] \left[A_{r_L}^\pi[s, a] - \nabla_\theta \log \pi_\theta |_{\theta=\theta(t)} [\theta(t+1) - \theta(t)] \right] \right\} + \\ & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] \frac{\log \pi(t+1)}{\log \tilde{\pi}(t)} \right] - \\ & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\lambda}_i(t) \pi^*[a|s] A_c^\pi[s, a] \right] - \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\mu}_i(t) \pi^*[a|s] A_p^\pi[s, a] \right] - \\ & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] \gamma [\theta(t)] [\theta(t+1) - \theta(t)] \right]. \end{aligned}$$

Using the transformation and the performance difference theorem, it can be obtained that

$$A_{r_L}^\pi[s, a] = \nabla_\theta \log \pi_\theta |_{\theta=\theta(t)} [\theta(t+1) - \theta(t)], \quad (51)$$

$$\begin{aligned} & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\lambda}_i(t) \pi^*[a|s] A_c^\pi[s, a] \right] = \\ & \mathbb{E}_{\pi^*} [\lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]], \quad (52) \\ & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \tilde{\mu}_i(t) \pi^*[a|s] A_p^\pi[s, a] \right] = \\ & \mathbb{E}_{\pi^*} [\mu_i(t+1)p[s_\sigma(t+1), a_\sigma(t+1)]] - \mathbb{E}_{\pi^*} [\mu_i(t)p[s_\sigma(t), a_\sigma(t)]]. \quad (53) \end{aligned}$$

It can be then derived that

$$\begin{aligned} & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] A_r^\pi[s, a] \right] = \mathbb{E}_{\pi^*} \left\{ \sum_{a \in \mathcal{A}} \pi^*[a|s] \left[A_{r_L}^\pi[s, a] - A_{r_L}^\pi[s, a] \right] \right\} - \\ & \mathbb{E}_{\pi^*} [\lambda_i(t+1)c[s_\sigma(t+1), a_\sigma(t+1)]] + \mathbb{E}_{\pi^*} [\lambda_i(t)c[s_\sigma(t), a_\sigma(t)]] - \\ & \mathbb{E}_{\pi^*} [\mu_i(t+1)p[s_\sigma(t+1), a_\sigma(t+1)]] + \mathbb{E}_{\pi^*} [\mu_i(t)p[s_\sigma(t), a_\sigma(t)]] + \\ & \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] \frac{\log \pi(t+1)}{\log \pi(t)} \right] - \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] \gamma [\theta(t)] [\theta(t+1) - \theta(t)] \right]. \end{aligned}$$

Therefore, it can be obtained that

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] A_r^\pi[s, a] \right] \leq \\ & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left\{ \sum_{a \in \mathcal{A}} \pi^*[a|s] \left[A_{r_L}^\pi[s, a] - A_{r_L}^\pi[s, a] \right] \right\} - \frac{\tilde{\lambda}\bar{c}}{T} - \frac{\tilde{\mu}\bar{p}}{T} + \\ & \frac{\log N F C \tau_{\max}}{\eta_1 T} - \frac{\gamma [\theta(T)] \bar{\theta}_T}{T}, \end{aligned}$$

where the second, third, and fourth term in the right-hand side use the same derivation process in Appendix D. The last term has the similar expression as that of the second and third term, so it is derived using the same process. Using the expansion of $A_r^\pi[s, a]$ in

(34) of Theorem 2 to $A_r^\pi[s, a]$, it can be derived that

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left\{ \sum_{a \in \mathcal{A}} \pi^*[a|s] \left[A_{r_L}^\pi[s, a] - A_{r_L}^\pi[s, a] \right] \right\} \leq \\ & \frac{\mathbb{E}_{\pi^*} \left[\log \frac{\tilde{r}_T}{\pi_T} \right]}{\eta_1 T} + \frac{1}{\eta_1 T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left\{ \sum_{a \in \mathcal{A}} \pi^*[a|s] \left[\log \tilde{Z}(s) - \log Z(s) \right] \right\} \\ & = \frac{\mathbb{E}_{\pi^*} \left[\log \frac{\tilde{r}_T}{\pi_T} \right]}{\eta_1 T} + \frac{1}{T} \{ \mathbb{E}_{\pi^*} [\tilde{r}[s_\sigma(T), a_\sigma(T)]] - \mathbb{E}_{\pi^*} [r[s_\sigma(T), a_\sigma(T)]] \} \\ & + \frac{1}{T} \{ \mathbb{E}_{\pi^*} [\tilde{\lambda}_i(T)\tilde{c}[s_\sigma(T), a_\sigma(T)]] - \mathbb{E}_{\pi^*} [\lambda_i(T)c[s_\sigma(T), a_\sigma(T)]] \} \\ & + \frac{1}{T} \{ \mathbb{E}_{\pi^*} [\tilde{\mu}_i(T)\tilde{p}[s_\sigma(T), a_\sigma(T)]] - \mathbb{E}_{\pi^*} [\mu_i(T)p[s_\sigma(T), a_\sigma(T)]] \} \end{aligned}$$

It should be noted that the third and fourth term of the above expressions are affected by the difference of number of subcarriers and power. Suppose the third term is denoted by δ_c and the fourth term is denoted by δ_p , it can be obtained that

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \pi^*[a|s] A_r^\pi[s, a] \right] \leq \frac{g(N, F, C, \tau_{\max}) + q(\delta_c, \delta_p)}{T}, \quad (54)$$

where

$$\begin{aligned} q(\delta_c, \delta_p) &= \frac{\mathbb{E}_{\pi^*} \left[\log \frac{\tilde{r}_T}{\pi_T} \right]}{\eta_1} + \tilde{r}_T - 2\bar{r}_T + \delta_c + \delta_p - 2\tilde{\lambda}\bar{c} - 2\tilde{\mu}\bar{p} - \\ & \gamma [\theta(T)] \bar{\theta}_T + \frac{\mathbb{E}_\pi (\lambda_i^2(T)) - \epsilon_\lambda}{2\eta_2} + \frac{\mathbb{E}_\pi (\mu_i^2(T)) - \epsilon_\mu}{2\eta_3} \quad (55) \end{aligned}$$

It should be noted that $q(\delta_c, \delta_p)$ is still a finite value. Therefore, as T approaches infinity, the optimality gap goes to zero.