MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Learning Based Routing Link Scheduling in Heterogeneous Wireless IoT Networks

Wang, Zhiyang; Guo, Jianlin; Parsons, Kieran; Nagai, Yukimasa; Sumi, Takenori; Orlik, Philip V.

TR2024-069 June 11, 2024

Abstract

With the advent of 5G and beyond communication technologies, the consumer Internet of Things (IoT) devices are evolving from the current-generation to the next-generation. Next-generation IoT devices can support multiple communication interfaces and perform more functions. Accordingly, IoT network technologies must adapt to the emerging next- generation IoT devices. Routing is an inevitable technology in multi-hop IoT networks. However, as IoT devices become more and more diverse, IoT networks become more complex. As a result, the routing problem becomes more and more complicated for traditional protocols and mathematical optimization approaches to provide optimal solutions. Machine learning based routing techniques have been recently proposed and can outperform traditional routing methods in complex network environments. To that end, this paper presents a machine learning based routing link scheduling scheme for heterogeneous wireless IoT networks. We formulate the routing link scheduling problem as a combinatorial optimization problem, which is then parameterized for application of machine learning algorithm and the parameterized problem is solved using primal-dual approach with zero duality gap. A heterogeneous graph neural network (HetGNN) algorithm is proposed to update the primal- dual problems. We evaluate the proposed HetGNN model under networks with randomly deployed heterogeneous nodes. Compared with a convolutional neural network (CNN) model and a homogeneous GNN (HomGNN) model, the proposed HetGNN model can improve network throughput, reduce link capacity violation and interference link violation.

IEEE International Conference on Communications (ICC) 2024

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Learning Based Routing Link Scheduling in Heterogeneous Wireless IoT Networks

Zhiyang Wang*‡, Jianlin Guo*, Kieran Parsons*, Yukimasa Nagai†, Takenori Sumi† and Philip Orlik*
*Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
†Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa 2478501, Japan
‡ Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia PA 19104, USA

***Abstract*—With the advent of 5G and beyond communication technologies, the consumer Internet of Things (IoT) devices are evolving from the current-generation to the next-generation. Next-generation IoT devices can support multiple communication interfaces and perform more functions. Accordingly, IoT network technologies must adapt to the emerging next-generation IoT devices. Routing is an inevitable technology in multi-hop IoT networks. However, as IoT devices become more and more diverse, IoT networks become more complex. As a result, the routing problem becomes more and more complicated for traditional protocols and mathematical optimization approaches to provide optimal solutions. Machine learning based routing techniques have been recently proposed and can outperform traditional routing methods in complex network environments. To that end, this paper presents a machine learning based routing link scheduling scheme for heterogeneous wireless IoT networks. We formulate the routing link scheduling problem as a combinatorial optimization problem, which is then parameterized for application of machine learning algorithm and the parameterized problem is solved using primal-dual approach with zero duality gap. A heterogeneous graph neural network (HetGNN) algorithm is proposed to update the primal-dual problems. We evaluate the proposed HetGNN model under networks with randomly deployed heterogeneous nodes. Compared with a convolutional neural network (CNN) model and a homogeneous GNN (HomGNN) model, the proposed HetGNN model can improve network throughput, reduce link capacity violation and interference link violation.**

***Index Terms*—Routing link scheduling, heterogeneous IoT network, graph neural network, primal-dual method, combinatorial optimization.**

I. Introduction

With the advent of 5G and beyond communication technologies, consumer IoT devices are evolving from the current-generation to the next-generation. The current-generation IoT devices are equipped with fewer resources and perform simple functions. On the other hand, the next-generation IoT devices are installed with more resources and can perform more functions. Take smart meters for example, current-generation meters support one communication interface such as Wi-Fi (IEEE 802.11) or Wi-SUN (IEEE 802.15.4) and collect metering data only. Next-generation meters can support multiple communication interfaces, such as both Wi-Fi/Wi-SUN and 5G, collect metering data, and sense power supply information, which is critical for smart grid maintenance. However, it is impractical to completely remove the deployed current-generation devices during the migration phase. Accordingly, next-generation IoT networks will consist of mixed current and next-generation devices. Customers expect more from next-generation IoT networks. Therefore, new network technologies need to be developed for next-generation IoT networks to meet consumers' expectations.

IoT networks are typically multi-hop. As a result, routing is inevitable. Routing consists of route discovery and route scheduling. Both route discovery and route scheduling are high-complexity problems. The route discovery can be NP-complete [1], e.g., maximizing throughput in a multi-hop wireless network is proved to be NP-hard as a result of wireless interference [2]. It has also been proved that both centralized and distributed route scheduling problems are NP-complete in 2D mesh topology [3].

Route scheduling is more complex than route discovery, especially in carrier sense multiple access (CSMA) based multi-hop wireless networks, where wireless interference presents great challenges. Route scheduling in wireless networks is a spectrum resource allocation problem. It aims to schedule channel access for data transmission to avoid mutual transmission interference and channel access delay, thus improving network efficiency. For example, the time synchronized channel hopping (TSCH) is a scheduling mechanism provided in IEEE 802.15.4 standard. However, the conventional scheduling mechanisms such as TSCH require clock synchronization, which is intractable to be realized in wireless IoT networks, especially in multi-hop IoT networks. Machine learning based routing techniques have been recently proposed and can outperform traditional routing methods in complex network environments [4], which makes them promising approaches for next-generation IoT networks.

Besides wireless interference, the emerging heterogeneous IoT networks add more complexity into routing problems. The heterogeneity such as device heterogeneity, data heterogeneity and communication capability heterogeneity must be considered for next-generation IoT network technology development since more capable devices can play an important role to improve network performance, e.g., the devices with multiple communication interfaces can significantly improve IoT network performance [5]. However, most existing routing

This work was done while Zhiyang Wang was working at Mitsubishi Electric Research Laboratories (MERL) as an intern.

technologies are designed for current-generation homogeneous networks without considering network heterogeneity.

This paper focuses on route scheduling in multi-hop heterogeneous wireless IoT networks. We present a machine learning based routing link scheduling scheme by formulating a routing link scheduling problem as an optimization problem, which is then parameterized for applying graph neural network techniques. The parameterized problem is solved using the primal-dual approach with a zero duality gap.

The rest of this paper is organized as follows. Section II presents related works. Section III provides system model and problem formulation. The scheduling policy parameterization is given in Section IV. Section V introduces the primal-dual problem formulation. Performance evaluation is conducted in Section VI. Finally, we conclude our paper in Section VII.

II. Related Works

Efficient scheduling of transmissions is a key problem in wireless networks. The main challenge stems from the fact that optimal link scheduling involves solving a maximum weighted independent set problem, which is known to be NP-hard [6]. No efficient global optimal algorithm is available yet for routing link scheduling in device-to-device networks, especially for a densely deployed network with a large number of mutually interfering links [7].

Paper [8] studies the problem of joint routing, link scheduling and power control to support high data rates for broadband wireless multi-hop networks. The authors address the problem of finding an optimal link scheduling and power control policy that minimizes the total average transmission power in a wireless multi-hop network. It is found that optimum allocations do not necessarily route traffic over minimum energy paths. Work [9] proposes a joint routing and scheduling optimization in time-sensitive networks using graph convolutional network (GCN) based deep reinforcement learning for time-sensitive applications. Numerical simulations demonstrate that the proposed algorithm has good convergence and outperforms the benchmark methods in terms of the average end-to-end delay. Authors in [10] introduce a number of routing scheduling algorithms that, using certain knowledge about the network structure, guarantee stability for certain injection rates. The authors provide some results regarding both the maximum latency and queue length and also evaluate how the lack of global knowledge about the system topology affects the performance of the routing scheduling algorithms. Paper [11] proposes a TSCH based scheduling method for multi-hop time sensitive networks. A scheduling scheme is designed to minimize the schedule length and the maximum end to end delay. Link level simulations verify the performance improvement of the proposed scheme over the existing schemes.

Owing to the complexity of routing scheduling problems, machine learning techniques have been recently applied to routing scheduling. Work [6] presents link scheduling

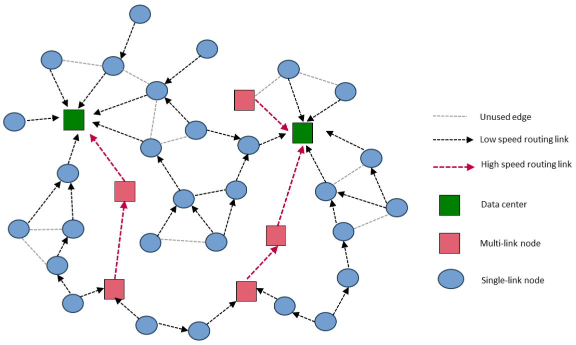

Figure 1: Heterogeneous Wireless IoT Network System Model

technologies using graph neural networks. The fast heuristics based on GCN are proposed to be implemented in centralized and distributed manners. Test results show that the proposed centralized heuristic can reach a near-optimal solution quickly, and the distributed heuristic based on a shallow GCN can reduce by nearly half the suboptimality gap of the distributed greedy solver with minimal increase in complexity. Authors in [4] propose a deep reinforcement learning based dynamic routing optimization algorithm for delay-sensitive applications featuring the proximal policy optimization method and the front-convergent actor-critic network technique. Authors consider the packet survival time to make up for the shortage of the time-to-live mechanism in conventional IP networks. Experimental results show that the proposed algorithm outperforms two traditional routing protocols in terms of minimizing delay and packet loss rate. However, the aforementioned works deal with homogeneous wireless networks.

This paper studies routing link scheduling in heterogeneous IoT networks. We assume that network topology and routes are given. Our objective is to schedule efficient data transmission along the given routes. To that end, we propose machine learning based routing link scheduling techniques to improve heterogeneous IoT network performance.

III. System Model and Problem Formulation

In this section, we present our system model and routing link scheduling problem formulation.

A. System Model

This paper assumes symmetric communication connectivity, i.e., edge (link) is directionless. However, a routing link is directional. We consider a heterogeneous wireless IoT network consisting of single-link data node set $\mathcal{S}$, multi-link data node set $\mathcal{M}$ and data centers as illustrated in Fig. 1, where some edges are used in routing and others are not involved in routing. Single-link data nodes support one communication interface, multi-link data nodes support multiple communication interfaces and data centers are considered as multi-link nodes. A single-link data node can communicate with neighboring single-link nodes, multi-link nodes and data centers using a low-speed communication interface. On the other hand, a multi-link data node can communicate with neighboring single-link data nodes using the same low-speed communication interface and with neighboring multi-link data nodes and data centers using a high-speed communication interface.

The total data node set is $\mathcal{V} = \mathcal{S} \cup \mathcal{M}$ indexed with $k = \{1, 2, \cdots, |\mathcal{S}|, |\mathcal{S}|+1, \cdots, |\mathcal{S}|+|\mathcal{M}|\}$. The data sets for data nodes are denoted as $\mathbf{x} = \{x_1, x_2, \cdots x_{|\mathcal{S}|+1}, \cdots, x_{|\mathcal{S}|+|\mathcal{M}|}\}$, where the data of a data node include self data and potential relay data in the routing process. The edge set is denoted as $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ with $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$, where $\mathcal{E}_2$ contains the edges between nodes belonging to $\mathcal{M}$ and data centers, and $\mathcal{E}_1$ represents the rest of the edges.

The network topology and routes from all data nodes to data centers are given as illustrated in Fig. 1. The route set is denoted as $\mathcal{R} = \{R_1, R_2, \cdots, R_{NoR}\}$ with each route containing a sequence of ordered nodes, i.e., links, where NoR is the number of routes and $NoR \geq |\mathcal{S}| + |\mathcal{M}|$ since a data node may have multiple routes. It can be seen that each routing link overlaps an edge. Without loss of generality, we assume that $\mathcal{E}_2$ links are high-speed long range communication links such as 5G links and $\mathcal{E}_1$ links are low-speed short range communication links such as Wi-Fi/Wi-SUN links. A route can consist of $\mathcal{E}_1$ links only or $\mathcal{E}_2$ links only or both $\mathcal{E}_1$ and $\mathcal{E}_2$ links. We assume that $\mathcal{E}_2$ links such as 5G links do not interfere with each other and $\mathcal{E}_1$ links such as Wi-Fi/Wi-SUN links can however interfere with each other. Our objective is to schedule routing links for efficient data transmission to avoid transmission collision and random backoff delay incurred by the CSMA mechanism in Wi-Fi/Wi-SUN channel access mechanisms.

B. Routing Link Scheduling Problem Formulation

The route scheduling in shared medium wireless networks can be divided into two categories: whole route scheduling and routing link scheduling. In whole route scheduling, the non-interfering routes can be scheduled to transmit simultaneously. In routing link scheduling, the non-interfering links can be scheduled to transmit simultaneously. However, whole route scheduling is less efficient than routing link scheduling, especially in data centric networks, where all routes have one of the data centers as the destination node, as a result, the routes that could be scheduled to transmit simultaneously are the routes heading towards different data centers. On the other hand, it is possible that multiple links on different routes or on same route can be scheduled to transmit simultaneously without causing interference. Therefore, this paper focuses on the routing link scheduling.

Before formulating the routing link scheduling problem, we briefly introduce the link interference in shared medium wireless communication. A set of links $L_1 \triangleq a_1 \rightarrow b_1$, $L_2 \triangleq a_2 \rightarrow b_2$, $\cdots$, $L_n \triangleq a_n \rightarrow b_n$ are interfering links if and only if there exists at least one b_j such that b_j is a neighbor of at least two a_is. There are different ways a node can discover its neighbors, e.g., via probe and response mechanism. For two $\mathcal{E}_1$ links L_i and L_j, we use notation $L_i \cap L_j = \emptyset$ to indicate interference free.

Denote as $\mathcal{L} = \{L_1, L_2, \cdots, L_N\}$ the set of $\mathcal{E}_1$ and $\mathcal{E}_2$ links that are on routes $R_1, R_2, \cdots, R_{NoR}$. The link capacity, i.e., channel capacity, for link L_i is denoted as C_i. The vector $\mathbf{q} \in \{0,1\}^{N\times 1}$ shows if the links in $\mathcal{L}$ have data to transmit, i.e., the starting nodes of links have data to transmit, with $q_i = 1$ indicating yes and $q_i = 0$ indicating no. Our goal is to determine a scheduling decision vector $\mathbf{p} \in \{0,1\}^{N\times 1}$ to transmit as much data as possible with $p_i = 1$ indicating link L_i scheduled to transmit and $p_i = 0$ indicating otherwise.

We formulate the routing link scheduling problem in heterogeneous wireless IoT networks as an optimization problem. More specifically, the routing link scheduling problem is formulated as following optimization problem

$$\mathbf{p}(\mathcal{L},\mathbf{x}) = \arg\max_{\mathbf{P}} \sum_{i=1}^{N} p_i x_i \tag{1}$$

$$\begin{aligned} s.t. \quad & L_i \cap L_j = \emptyset, \text{ for all } L_i \ \& \ L_j \in \mathcal{E}_1, \\ & p_i = p_j = 1, \ 1 \leq i, \ j \leq N, \ i \neq j && (2) \\ & x_i \leq \mathbb{E}_{\mathbf{H}}[C_i], \text{ for all } p_i = 1 && (3) \\ & p_i \in \{0,1\}, \ 1 \leq i \leq N && (4) \\ & p_i = 1 \text{ only if } q_i = 1, \ 1 \leq i \leq N && (5) \end{aligned}$$

where condition (2) is the interference constraint such that the scheduled $\mathcal{E}_1$ links do not interfere with each other, condition (3) is the link capacity constraint such that the amount of data scheduled for a link does not exceed the expected link capacity, condition (4) is the scheduling decision constraint such that a link is either scheduled or not scheduled and condition (5) is the data availability constraint such that a link is scheduled only if the starting node of the link has data to transmit. For the link capacity constraint (3), we use the expected link capacity to take dynamic channel conditions into account. The expected link capacity cab be dynamically estimated using a wireless channel model.

IV. Link Scheduling Policy Parameterization for Heterogeneous Graph Neural Network

The routing link scheduling policy parameterization and graph neural network embedding are introduced in this section.

A. Routing Link Scheduling Policy Parameterization

The optimization problem (1) is a combinatorial optimization problem, which is generally NP-hard [7], [12]. Directly solving optimization problem (1) can be intractable due to the discrete policy variables and constraints. To tackle the challenge, we parameterize the scheduling policy $\mathbf{p}(\mathcal{L}, \mathbf{x})$ for the application of machine learning techniques with a mapping function $\Psi(\mathbf{W}, \mathcal{L}, \mathbf{x})$, where the parameter set $\mathbf{W}$ is of controllable dimension. Accordingly, the optimization problem (1) can be rewritten as

$$\Psi(\mathbf{W}, \mathcal{L}, \mathbf{x}) = \arg\max_{\mathbf{W}} \sum_{i=1}^{N} \Psi_i(\mathbf{W}) x_i \tag{6}$$

$$\begin{aligned} s.t. \quad & L_i \cap L_j = \emptyset, \text{ for all } L_i \ \&\ L_j \in \mathcal{E}_1,\ \Psi_i(\mathbf{W}) \\ & \quad = \Psi_j(\mathbf{W}) = 1,\ 1 \le i,\ j \le N,\ i \neq j \quad (7) \\ & x_i \le \ \mathbb{E}_{\mathbf{H}}[C_i], \text{ for all } \Psi_i(\mathbf{W}) = 1 \quad (8) \\ & \Psi_i(\mathbf{W}) \in \{0, 1\},\ 1 \le i \le N \quad (9) \\ & \Psi_i(\mathbf{W}) = 1 \text{ only if } q_i = 1,\ 1 \le i \le N \quad (10) \end{aligned}$$

Denote in matrix form, let $\mathbf{A}_V : \{0, 1\}^{|\mathcal{V}| \times N}$ represent the presentation matrix of nodes involved in routing links. In the i-th column of $\mathbf{A}_V$, the element with entry 1 indicates that this node is used in link L_i as either the starting node or the ending node. Similarly, we can define $\mathbf{A}_S : \{0, 1\}^{|\mathcal{S}| \times N}$ to represent the usage of single-link nodes in all routing links and $\mathbf{A}_I : \{0, 1\}^{N \times N}$ to represent routing link interference matrix with the entry $[\mathbf{A}_I]_{ij} = 1$ indicating links L_i and L_j interfere with each other. With these notations, we can rewrite the optimization problem as

$$\Psi(\mathbf{W}, \mathcal{L}, \mathbf{x}) = \arg\max_{\mathbf{W}} \quad \mathbf{x}^{\mathsf{T}} \mathbf{A}_V \Psi(\mathbf{W}) \tag{11}$$

$$\begin{aligned} s.t. \quad & \mathbf{A}_S \mathbf{A}_I \Psi(\mathbf{W}) \le \mathbb{1}_{|\mathcal{S}|} \quad (12) \\ & x_i \le \mathbb{E}_{\mathbf{H}}[C_i], \text{ for all } \Psi_i(\mathbf{W}) = 1 \quad (13) \\ & \Psi_i(\mathbf{W}) \in \{0, 1\},\ 1 \le i \le N \quad (14) \\ & \Psi_i(\mathbf{W}) = 1 \text{ only if } q_i = 1,\ 1 \le i \le N \quad (15) \end{aligned}$$

B. Heterogeneous Graph Neural Network Embedding

Link scheduling is usually formulated as a non-convex combinatorial problem, which is generally difficult to get the optimal solution [7]. Traditional methods to solve this problem are mainly based on mathematical optimization techniques with high computation complexity. To overcome the high computational complexity, machine leaning based approaches have been introduced recently, e.g., the GNN techniques have been employed in link prediction, node classification and graph regression [13].

Our system model shown in Fig. 1 can be represented as a heterogeneous graph since there are different types of nodes and edges. Therefore, we propose the HetGNN technique for our optimization problem. The node features include node type (single-link or multi-link), amount of data, neighbors (single-link neighbors and multi-link neighbors), its involvement in routing links (number of links involved, starting node or ending node of links), etc. The edge features include edge type ($\mathcal{E}_1$ or $\mathcal{E}_2$), edge capacity, node pair, edge usability in routing, edge direction in routing, interference edges, etc. The given routes can be considered as meta-paths in graph neural networks.

For a node $k \in \mathcal{V}$, its neighbor set is denoted as $\mathcal{N}(k)$. The node embedding is defined as

$$\begin{aligned} \mathbf{h}_k^{\mathcal{V}} = \sigma(\mathbf{F}_k + & \\ \sum_{\substack{u \in \mathcal{N}(k) \\ e=(u,k)}} & \left(\mathbf{W}_{E_1} \mathbb{1}(e \in \mathcal{E}_1) + \mathbf{W}_{E_2} \mathbb{1}(e \in \mathcal{E}_2)\right) \mathbf{h}_e^{\mathcal{E}} \right), \end{aligned} \tag{16}$$

where $\mathbf{F}_k$ is the feature vector of node k, $\mathbf{W}_{E_1}$ is the parameter set for $\mathcal{E}_1$ edges, $\mathbf{W}_{E_2}$ is the parameter set for $\mathcal{E}_2$ edges, $\mathbf{h}_e^{\mathcal{E}}$ takes account of the routing involvement of edge e and σ is node activation function such as the rectified linear unit (ReLU).

For an edge $e \in \mathcal{E}$, the edge embedding is defined as

$$\mathbf{h}_e^{\mathcal{E}} = \sigma_e \left(\mathbf{h}_u^{\mathcal{V}} + \mathbf{h}_k^{\mathcal{V}}\right), \text{ with } (u, k) = e \tag{17}$$

where $\mathbf{h}_u^{\mathcal{V}}$ and $\mathbf{h}_k^{\mathcal{V}}$ are embedded feature vectors of nodes connected by edge e and σ_e is edge activation function.

Accordingly, for a route $R \in \mathcal{R}$, the meta-path embedding is defined as

$$\begin{aligned} \mathbf{h}_R^{\mathcal{R}} = & \sigma_m(\mathbf{h}_{R(1)}^{\mathcal{V}}, \mathbf{h}_{(R(1),R(2))}^{\mathcal{E}} \cdots, \\ & \mathbf{h}_{(R(|R|-1),R(|R|))}^{\mathcal{E}}, \mathbf{h}_{R(|R|)}^{\mathcal{V}}), \end{aligned} \tag{18}$$

where σ_m is meta-path activation function.

V. Primal-Dual Algorithm for Unsupervised Training

The problem (11) is also a combinatorial optimization problem. The primal-dual method is a standard approach in the design of algorithms for combinatorial optimization problems [8], [14]. Accordingly, we apply primal-dual method to solve the problem (11).

We follow work [15] in the primal-dual problem formulation. The primal problem is

$$\mathsf{P} = \max_{\mathbf{W}} \quad \mathbf{x}^{\mathsf{T}} \mathbf{A}_V \Psi(\mathbf{W}) \tag{19}$$

$$\begin{aligned} s.t. \quad & x_i \le \mathbb{E}_{\mathbf{H}}[C_i], \text{ for all } \Psi_i(\mathbf{W}) = 1 \quad (20) \\ & \mathbf{A}_S \mathbf{A}_I \Psi(\mathbf{W}) \le \mathbb{1}_{|\mathcal{S}|} \quad (21) \\ & \Psi_i(\mathbf{W}) \in \{0, 1\},\ 1 \le i \le N \quad (22) \\ & \Psi_i(\mathbf{W}) = 1 \text{ only if } q_i = 1,\ 1 \le i \le N \quad (23) \end{aligned}$$

The Lagrangian form of primal problem (19) is defined by associating Lagrange multipliers $\boldsymbol{\mu}$ with the capacity constraint and $\boldsymbol{\lambda}$ with the interference constraint, i.e., the Lagrangian problem can be written as

$$\begin{aligned} L[\mathbf{W}, \boldsymbol{\Lambda}] = \mathbf{x}^{\mathsf{T}} \mathbf{A}_V \Psi(\mathbf{W}) + \sum_{i=1}^{N} \mu_i \Psi_i(\mathbf{W}) \left(x_i - \mathbb{E}_{\mathbf{H}}[C_i]\right) \\ + \boldsymbol{\lambda}^{\mathsf{T}} (\mathbf{A}_S \mathbf{A}_I \Psi(\mathbf{W}) - \mathbb{1}_{|\mathcal{S}|}) \end{aligned} \tag{24}$$

The dual function is obtained by maximizing the Lagrangian over the primal variables as

$$g[\boldsymbol{\Lambda}] = \max_{\mathbf{W}, \Psi_i(\mathbf{W}) \in \{0,1\}} L[\mathbf{W}, \boldsymbol{\Lambda}]. \tag{25}$$

As a result, the dual problem is defined as

$$\mathsf{D} = \min_{\mathbf{\Lambda} \geq 0} g[\mathbf{\Lambda}]. \tag{26}$$

The primal problem is non-convex [15], which implies that the duality gap can be non-zero, i.e. $\mathsf{D} \geq \mathsf{P}$. Therefore, this is called a duality relaxation on the primal problem.

For some problems, the duality gap is zero. For routing link scheduling problems, when the channel between each pair of nodes is deterministic, the problem is known to be NP-hard. Introducing fading channels can vanish the duality gap [15], i.e. $\mathsf{P} = \mathsf{D}$. More specifically, work [15] provides the following theorem

Theorem 1. *If the channel cumulative distribution function (cdf) is continuous, then* $\mathsf{P} = \mathsf{D}$.

Assuming the optimal set of dual variables $\mathbf{\Lambda}^*$ is available, the primal updates can be formulated as

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \epsilon_t \nabla_{\mathbf{W}} \Psi(\mathbf{W})(\mathbf{A}_V^\top \mathbf{x} + \mathbf{A}_I^\top \mathbf{A}_S^\top \boldsymbol{\lambda}(t)) \tag{27}$$

and the dual updates can be formulated as

$$\mu_i(t+1) = [\mu_i(t) - \epsilon_t \Psi_i(\mathbf{W}(t))(x_i - \mathbb{E}_{\mathbf{H}}[C_i])]^+ , \tag{28}$$

$$\boldsymbol{\lambda}(t+1) = \left[\boldsymbol{\lambda}(t) - \epsilon_t(\mathbf{A}_S \mathbf{A}_I \Psi(\mathbf{W}(t)) - \mathbb{1}_{|\mathcal{S}|})\right]^+ , \tag{29}$$

where $[\cdot]^+$ denotes componentwise maximum between 0 and the value inside the square brackets, while ϵ_t is a properly selected stepsize.

As we can notice, (28) cannot be computed without explicit knowledge of fading channel distribution $\mathbf{H}$ and data distribution $\mathbf{x}$. We solve this by sampling a realization $\mathbf{H}(t)$ and update according to:

$$\mu_i(t+1) = [\mu_i(t) - \epsilon_t \Psi_i(\mathbf{W}(t))(x_i - C_i(\mathbf{H}(t)))]^+ , \tag{30}$$

which enables us to use the observed capacity $C_i(\mathbf{H}(t))$ directly without the need to know the explicit fading channel condition. The unsupervised HetGNN training process is presented in detail in Algorithm 1.

Algorithm 1 Primal-Dual Unsupervised Training Algorithm

Input 1: Node set $\mathcal{V} = \mathcal{S} + \mathcal{M}$
Input 2: Network topology
Input 3: Route set $\mathcal{R}$
Discover neighbor set for each node
Configure routing link set $\mathcal{L}$
Construct interference links for each routing link
for $t \in \mathbb{Z}$ **do**
  Update node features
  Update edge features
  Embed all the routes (metapaths) and generate current scheduling policy $\Psi(\mathbf{W}(t))$
  Observe total data amount collected by the data centers
  Update primal and dual variables
  $\mathbf{W}(t+1) = \mathbf{W}(t) + \epsilon_t \nabla_{\mathbf{W}} \Psi(\mathbf{W})(\mathbf{A}_V^\top \mathbf{x} + \mathbf{A}_I^\top \mathbf{A}_S^\top \boldsymbol{\lambda}(t))$
  $\mu_i(t+1) = [\mu_i(t) - \epsilon_t \Psi_i(\mathbf{W}(t))(x_i - \mathbb{E}_{\mathbf{H}}[C_i])]^+$
  $\boldsymbol{\lambda}(t+1) = \left[\boldsymbol{\lambda}(t) - \epsilon_t(\mathbf{A}_S \mathbf{A}_I \Psi(\mathbf{W}(t)) - \mathbb{1}_{|\mathcal{S}|})\right]^+$
end for

VI. Numerical Experiments

This Section presents the performance evaluation of the proposed HetGNN scheme. In the simulation, we randomly deployed nodes in a $[0, 100\text{m}]^2$ square with 1 data center (red square), a set of single-link nodes (red circles) and a set of multi-link nodes (green circles). An example of node deployment is shown in Fig. 2 with 30 single-link data nodes and 5 multi-link data nodes. We calculate the channel capacity matrix by channel gains with the shadowing model

$$\log(X_s) \sim \text{Gaussian}(0, \sigma^2) \tag{31}$$

and dual-slope path-loss model [16]

$$PL(d) = \begin{cases} K_0 d^{\alpha_1} & \text{if } d \leq d_b \\ K_0 \frac{d^{\alpha_2}}{d_b^{\alpha_2 - \alpha_1}} & \text{if } d > d_b \end{cases} \tag{32}$$

with $K_0 = 39\text{dB}$ $\alpha_1 = 2$ $\alpha_2 = 4$ $d_b = 60\text{m}$. The channel capacity is computed as

$$C = BW \log_2(1 + \mathsf{SNR}), \tag{33}$$

where bandwidth (BW) for $\mathcal{E}_1$ links is 2.4GHz and for $\mathcal{E}_2$ links is 5GHz, noise power spectral density is -174dBm and power is 1mW.

To determine single-link neighbors and $\mathcal{E}_1$ interference links, we define a communication distance threshold D_I, which is also an interference threshold. For each pair of nodes i and j, we calculate distance d_{ij}. For a single-link node i, a node $j \in \mathcal{N}_i$ if $d_{ij} < D_I$, where node j can be either single-link or multi-link. In the simulation, the threshold D_I is set to 25m. The multi-link nodes can directly communicate with the data center.

We discover routes for randomly deployed data nodes by modifying Dijkstra's shortest path algorithm to fit the heterogeneous networks. More specifically, we modified the algorithm to take node type and edge type into account. One route is discovered for each data node. If the route lengths are the same for two candidate routes, a route with more $\mathcal{E}_2$ links is selected.

We used three metrics including data throughput, link capacity violation and interference violation to evaluate the performance, where link capacity constraint is violated if the scheduled data exceeds the link capacity and interference constraint is violated if the interfering $\mathcal{E}_1$ links are scheduled simultaneously. The CNN model and HomGNN model are used as benchmarks. We used PyTorch for simulation implementation.

For each node deployment scenario, data packets are generated with a random expectation and a fixed variance.

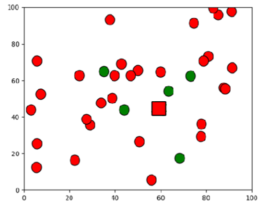

Figure 2: Node Deployment for Data Packet Generation

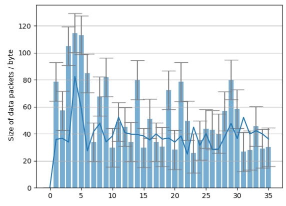

Figure 3: Random Data Packet Size Distribution

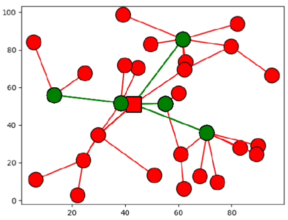

Figure 4: Node Deployment

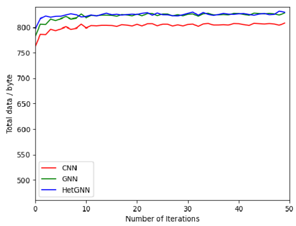

Figure 5: Data Throughput

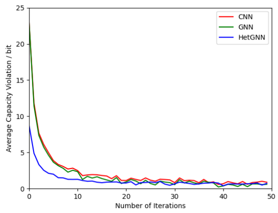

Figure 6: Routing Link Capacity Violation

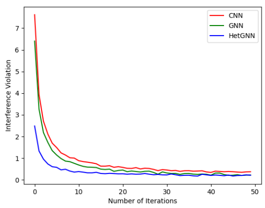

Figure 7: Routing Link Interference Violation

The average amount of data generated has the same order as the channel capacity. The multi-link data nodes are expected to have larger average data packets. Fig. 3 shows a distribution of data packet size for the node deployment shown in Fig. 2, where packet size unit is byte.

We conducted extensive simulations. Due to space limitation, we show the results of two node deployment scenarios with 24 single-link data nodes, 5 multi-link data nodes and 1 data center. It is interesting to observe that locations of multi-link nodes play important role in network performance.

Fig. 4 shows one of two deployments, for which Fig. 5 shows data throughput, CNN model delivers 808.27 bytes of data per scheduling step, HomGNN model delivers 828.27 bytes of data per scheduling step, and the proposed HetGNN model delivers 829.57 bytes of data per scheduling step. Accordingly, HetGNN improves improves CNN throughput by 2.6% and has similar throughput as HomGNN model.

Fig. 6 shows average link capacity violation, CNN model over schedules 0.88 bit of data per scheduling step, HomGNN model over schedules 0.7 bit of data per scheduling step and our HetGNN model over schedules 0.73 bit of data per scheduling step. As a result, HetGNN improves CNN

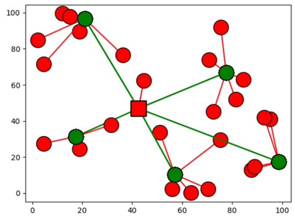

Figure 8: Node Deployment

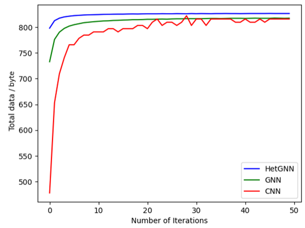

Figure 9: Data Throughput

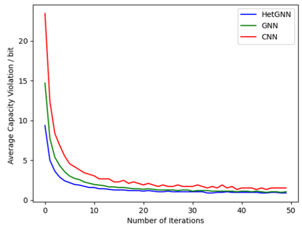

Figure 10: Routing Link Capacity Violation

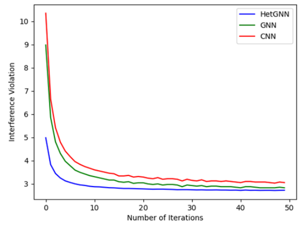

Figure 11: Routing Link Interference Violation

performance by 17% and has a similar performance as the HomGNN model.

Fig. 7 shows average link interference violation, CNN model violates interference constraint 0.38 time per scheduling step, HomGNN model violates interference constraint 0.22 time per scheduling step and HetGNN model violates 0.21 per scheduling step. Therefore, HetGNN reduces interference constraint violation over the CNN model by 45% and has a similar performance as the HomGNN model.

In summary, for the node deployment shown in Fig. 4, the proposed HetGNN model outperforms the CNN model and, however, performs similarly as the HomGNN model. This is because the multi-link nodes are close to the data center, and therefore, the advantage of multi-link nodes is limited.

The second node deployment is shown in Fig. 8, for which Fig. 9 shows data throughput, the CNN model delivers 807.57 bytes of data per scheduling step, HomGNN model delivers 808.17 bytes of data per scheduling step, and the HetGNN model delivers 827.7 bytes of data per scheduling step. Accordingly, HetGNN improves CNN throughput by 2.5% and improves HomGNN throughput by 2.4%.

Fig. 10 shows average link capacity violation, CNN model over schedules 1.55 bits of data per scheduling step, HomGNN model over schedules 1.02 bits of data per scheduling step and our HetGNN model over schedules 0.82 bit of data per scheduling step. As a result, HetGNN improves CNN performance by 47% and improves HomGNN performance by 20%.

Fig. 11 shows average link interference violation, CNN model violates interference constraint 3.08 times per scheduing stepe, HomGNN model violates interference constraint 2.83 times per scheduling step and HetGNN model violates

2.75 times per scheduling step. Therefore, HetGNN reduces interference constraint violation over the CNN model by 11% and reduces interference constraint violation over the HomGNN model by 3%.

In summary, for the node deployment shown in Fig. 8, the proposed HetGNN model outperforms both CNN and HomGNN models. This is because the multi-link nodes are away from the data center, and therefore, the advantage of multi-link nodes is explored by HetGNN model.

VII. Conclusions

This paper studies the routing link scheduling problem in heterogeneous IoT networks consisting of heterogeneous data nodes forming different types of links. The routing link scheduling problem is formulated as an optimization problem with link interference and channel capacity as constraints. Instead of applying a conventional optimization-based routing scheduling approach, we apply machine learning techniques. Accordingly, the optimization problem is parameterized. Due to the NP-Hard complexity of the optimization problem, the parameterized problem is further solved by applying a primal-dual approach. The formulated primal-dual problems are proved to have zero duality gap. We train primal-dual problems using heterogeneous graph neural network (HetGNN) techniques to learn unknown environments and link conditions for routing link scheduling policy. Compared with CNN and HomGNN models, the proposed HetGNN model can improve data throughput by up to 2.6%, reduce link capacity violation by up to 47% and lower link interference violation by up to 45%.

References

[1] F. A. Kuipers and P. F. A. V. Mieghem, "Conditions That Impact the Complexity of QoS Routing," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 717–730, 2005.

[2] S. Waharte, A. Golynski, and R. Boutaba, "On the Complexity of Routing in Wireless Multihop Network," in *8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2012.

[3] Y. Lu, Z. Zuo, H. He, and R. Li, "Further Complexity Results for Routing Schedule Problems of Networks," *IEEE Networking Letters*, vol. 1, no. 4, pp. 164–167, 2019.

[4] J. Chen, Y. Xiao, G. Lin, G. He, F. Liu, W. Zhou, and J. Liu, "Deep reinforcement learning based dynamic routing optimization for delay-sensitive applications," in *IEEE Global Communications Conference (GLOBECOM)*, 2023.

[5] J. Guo, T. Sumi, Y. Kawashima, K. Parsons, Y. Nagai, and P. Orlik, "Minimizing Route Overlap for Priority Data Delivery in Next Generation IoT Networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2023.

[6] Z. Zhao, G. Verma, C. Rao, and A. Swami, "Link Scheduling Using Graph Neural Networks," *IEEE Transaction on Wireless Communications*, vol. 22, no. 6, pp. 3997–4012, 2023.

[7] M. Lee, G. Yu, and G. Y. Li, "Graph Embedding-Based Wireless Link Scheduling With Few Training Samples," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2282–2294, 2021.

[8] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *IEEE INFOCOM*, 2003.

[9] L. Yang, Y. Wei, F. R. Yu, and Z. Han, "Joint Routing and Scheduling Optimization in Time-Sensitive Networks Using Graph-Convolutional-Network-Based Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 981–23 994, 2022.

[10] V. Cholvia, P. Garncarekb, T. Jurdzinski, and D. R. Kowalski, "Stable routing scheduling algorithms in multi-hop wireless networks," *Theoretical Computer Science*, vol. 921, pp. 20–35, 2022.

[11] D. Burghal, K. Kim, J. Guo, P. Orlik, T. Hori, T. Sumi, and Y. Nagai, "Multi-channel delay sensitive scheduling for convergecast network," in *IEEEWirelessCommunicationsandNetworkingConference(WCNC)*, 2020.

[12] T. Chondrogiannis, P. Bouros, J. Gamper, and U. Leser, "Alternative Routing: K-Shortest Paths with Limited Overlap," in *The 23rd SIGSPATIAL International Conference on Advances in Geographic Information*, 2015.

[13] X. Jiang, R. Zhu, P. Ji, and S. Li, "Co-embedding of Nodes and Edges with Graph Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7075–7086, 2023.

[14] M. X. Goemans and D. P. Williamson, *Approximation algorithms for NP-hard problems*. 20 Park Plaza Boston, MA, United States: PWS Publishing Co., 1996, ch. The primal-dual method for approximation algorithms and its application to network design problems, pp. 144–191.

[15] A. Ribeiro and G. B. Giannakis, "Separation Principles in Wireless Networking," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4488–4505, 2010.

[16] J. Andrews, X. Zhang, G. Durgin, and A. Gupta, "Are we approaching the fundamental limits of wireless network densification?" *IEEE Communications Magazine*, vol. 54, no. 10, pp. 184–190, 2016.