An Optimal Scheduling Algorithm for Maximizing Throughput in WiMAX Mesh Networks

Chih-Yung Chang, *Member, IEEE*, Ming-Hsien Li, Wen-Chuan Huang, and Shih-Chieh Lee

***Abstract*—The WiMAX mesh network (WMN) architecture is defined in the IEEE 802.16 standard for increasing network coverage and improving communication performance. In the past few years, several greedy or heuristic algorithms have been proposed to cope with the scheduling problem in WMNs. However, their performance highly depends on the network topology and bandwidth requests, and they do not achieve optimal performance in all cases. This paper proposes an optimal scheduling algorithm called the scheduling algorithm with dynamic programming approach (SADP), which exploits the opportunities of spatial reuse and maximizes the network throughput based on the network topology and the uplink bandwidth requests of each subscriber station. In addition, a heuristic scheduling algorithm (HSA) is proposed to reduce the computing complexity. The performance results were approximate to the optimal results. The simulation study reveals that the proposed SADP provides the WMN with maximal throughput and shortest transmission time, and the proposed HSA likely achieves the optimal results.**

***Index Terms*—Dynamic programming, scheduling, spatial reuse, WiMAX, 802.16 mesh networks.**

I. Introduction

A WIRELESS metropolitan area network provides wireless broadband services for an extensive coverage area. An IEEE 802.16 network consists of one base station (BS) and several subscriber stations (SSs). The BS functions as a gateway that ensures that each of its serving SSs can access external networks, such as the Internet. Each SS works as an access point that wirelessly connects to a dedicated BS and manages data delivery between the BS and mobile or static terminals.

The IEEE 802.16d standard [1] proposes two modes of frameworks: point-to-multipoint (PMP) and mesh. In the PMP framework, each SS must directly communicate with the BS; hence, each SS cannot be located more than the line-of-sight distance away from the BS. However, in a mesh framework, an SS is allowed to transmit data to the BS through other SSs in a multihop manner. The mesh network architecture extends the coverage of the BS and enables the routing path to be dynamically updated or repaired under unpredicted situations, such as path breakdown or inferior radio qualities for some relay SSs. Consequently, the mesh framework supports higher reliability and superior availability compared with the PMP framework.

The IEEE 802.16 mesh framework supports two resource allocation strategies: distributed scheduling and centralized scheduling. In centralized scheduling, BS is in charge of allocation of bandwidth and arranges the transmission time slots for each SS according to its bandwidth requirement. A tree topology rooted by the BS must be established to describe the routing information between the BS and SSs. In centralized scheduling, the BS periodically collects the network configuration and bandwidth requirement of each SS and arranges a schedule to avoid collision, reduce the transmission cycle, and maximize network throughput. In distributed scheduling, each SS determines its own transmission schedule according to local information, including bandwidth requirements and slot allocation within two-hop neighboring SSs. A contention-based algorithm is required to determine the slot allocation. Prior studies [4]–[6] implemented the distributed resource allocation mechanisms to negotiate flow transmission of neighboring SSs in WiMAX mesh networks (WMNs). Compared with distributed scheduling, the centralized scheduling mechanism maintains the global information, including the network topology and the bandwidth requirements from all SSs; hence, it is more efficient and easier to implement.

According to the IEEE 802.16d standard, two types of control messages are used in the centralized scheduling mechanism: mesh centralized scheduling configuration (MSH-CSCF) and mesh centralized scheduling (MSH-CSCH). The BS determines a tree topology of the WMN and notifies all SSs by flooding the entire network with the MSH-CSCF packet, which contains the tree topology information. Each SS establishes a link to the parent node in the tree according to the received MSH-CSCF message. The MSH-CSCH is the other control packet, which is composed of two types of messages: Request and Grant. Each SS notifies the BS of its bandwidth requirement by sending the MSH-CSCH:Request message to its parent node. After receiving the bandwidth requirement, the parent node prepares another MSH-CSCH:Request message that contains all bandwidth requirements of its child nodes and its own requirement and sends the message to its parent node. Finally, all MSH-CSCH:Request messages are delivered to the BS along the path defined in the tree topology. After

Manuscript received June 23, 2012; revised February 10, 2013; accepted February 27, 2013. Date of publication July 22, 2013; date of current version May 22, 2015. This work was supported by the National Science Council of Taiwan under Contracts NSC 100-2632-E-032-001-MY3 and NSC 100-2221-E-032-015-MY3.

C.-Y. Chang, M.-H. Li, and W.-C. Huang are with the Department of Computer Science and Information Engineering, Tamkang University, Taipei 251, Taiwan (e-mail: cychang@mail.tku.edu.tw; minghsienli@gmail.com; 894190122@s94.tku.edu.tw).

S.-C. Lee is with HTC Corporation, New Taipei City 231, Taiwan (e-mail: aaron_lee@htc.com).

Digital Object Identifier 10.1109/JSYST.2013.2271386

receiving all MSH-CSCH:Request messages, the BS allocates available time slots to each requested SS for transmitting data. Subsequently, the BS notifies all the SSs the decisions, including the bandwidth allocation and scheduling, using a MSH-CSCH:Grant message. The bandwidth allocation considers the bandwidth requirements from each SS and the child nodes. If the bandwidth requests of SSs change, the SSs inform the BS of their changes of bandwidth requests through the MSH-CSCH messages during the control subframe. After receiving the MSH-CSCH messages, the BS reschedules all transmissions by applying the scheduling algorithm and subsequently rebroadcasts the schedules to all nodes.

Generally, path planning and scheduling are the crucial factors that determine the throughput of a WMN. However, they are not detailed in IEEE 802.16 standards [1]–[3]. In literature, a number of studies [7]–[9] have proposed tree construction algorithms in a WMN. Wei *et al.* [7] proposed an interference-aware routing protocol that constructs a path from the source node to the destination node hop by hop. A source or forwarding node, e.g., a, would select the next forwarding node, e.g., b, from its neighboring nodes if the number of nodes encountering interference from the link (a, b) is minimal. To cope with the network congestion problem, Chen *et al.* [8] proposed a tree construction protocol for building a routing tree that considers load balance. In [9], a cross-layer tree construction protocol that considers both load balance and QoS parameters is proposed. Although previous studies [7]–[9] focused on constructing an efficient tree topology for increasing the network throughput, they did not consider the scheduling problem, which is another crucial factor that determines the network throughput of a WMN.

An efficient scheduling algorithm must consider four factors: collision, congestion, delay, and spatial reuse. Previous studies [10]–[14] considered metrics of spatial reuse and collision to schedule all transmissions with less delay time. Ramanathan and Lloyd [15] considered the constraint of deadline delay for supporting QoS transmissions. However, these works [10]–[15] used heuristic or greedy strategies; hence, the scheduling algorithms cannot always provide optimal scheduling. Several studies [16]–[19] have conducted QoS scheduling by considering the required bandwidth and delay over 802.16e PMP networks. However, these scheduling mechanisms are proposed to improve the throughput of the one-hop neighbors of the BS and are unsuitable for multihop WMNs.

This paper proposes an optimal scheduling algorithm and a heuristic scheduling algorithm (HSA) that allocates and schedules the available bandwidth for each SS in a mesh network for maximizing the network throughput and minimizing the time required for all transmission requests. Based on the existing mesh network topology, the proposed algorithms allocate each time slot to maximize the network throughput for exploiting the opportunities of spatial reuse. The remainder of this paper is organized as follows: Section II presents the existing scheduling algorithms for WMNs; Section III introduces the proposed optimal scheduling algorithm; Section IV introduces the HSA; Section V provides the performance evaluation of the proposed scheduling algorithm; and finally, Section VI offers a conclusion.

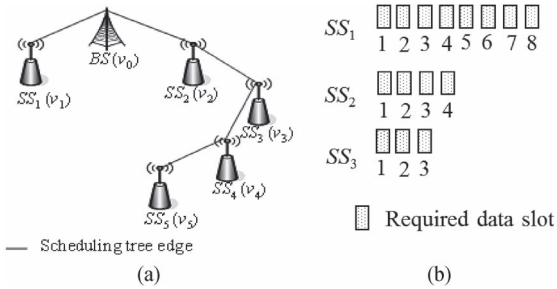


Fig. 1. Typical example of scheduling problem in an IEEE 802.16 mesh network. (a) Example of the scheduling tree topology. (b) SS_1, SS_2, and SS_5 require eight, four, and three time slots, respectively.

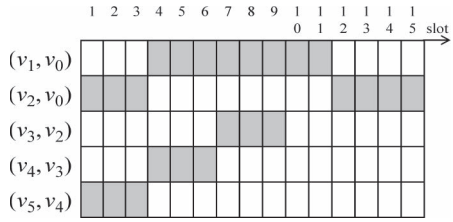


Fig. 2. Optimal scheduling of the example shown in Fig. 1(a).

II. Related Work

This section presents a review of the related scheduling mechanisms proposed in [10]–[14] and provides an example (see Fig. 1) for comparison with the proposed strategy. Let transmission rate denote the number of bits that are conveyed per second (bits/s). For simplicity, the example provided here assumes that all links in the tree topology have the same transmission rate. A general scheduling algorithm that considers variant rates is proposed in the next section. As shown in Fig. 1(b), the uplink transmission requests required from SS_1, SS_2, and SS_5 are eight, four, and three time slots, respectively. These data are routed along the paths defined in the tree topology and finally arrive at the BS.

Based on the tree topology shown in Fig. 1(a), Fig. 2 shows an optimal scheduling that considers both spatial reuse and collision-free factors. Let notation (v_i, v_j) indicate the link from v_i to v_j. First, the BS schedules a parallel transmission for v_2 and v_5 and assigns four and three transmission slots for them, respectively. After receiving data from v_5, v_4 starts to forward data to v_3 at slot 4. The BS switches to receive data from another neighbor, v_1. During the period of transmission from v_1 to the BS, another transmission (v_3, v_2) progresses simultaneously to increase the spatial reuse. Eventually, the transmission requirements for each SS are satisfied within 15 time slots.

In [10], Shetiya and Sharma proposed a scheduling algorithm that applies a dynamic programming scheme to achieve maximal throughput within n time slots. For a tree with various transmission rates on the tree links, the proposed scheduling algorithm starts allocating time slots from leaf nodes and ends at the BS. However, this algorithm does not consider spatial reuse; hence, only one SS is assigned to transmit data at each time slot. Because spatial reuse was not considered, various transmission arrangements have the same result in terms of required time slots. We used the sequence (v_5, v_4), (v_4, v_3),

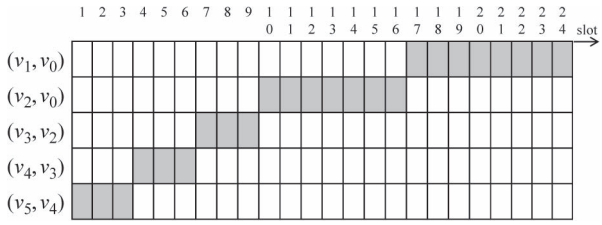


Fig. 3. Transmission schedule by applying the scheduling algorithm proposed in [10].

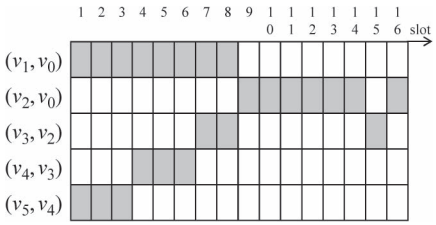


Fig. 4. Scheduling by applying the algorithm proposed in [11].

(v_3, v_2), (v_2, v_0), (v_1, v_0) to evaluate the total number of time slots. Fig. 3 shows the schedule by applying the algorithm proposed in [10]. The number of time slots required to satisfy all transmission requirements is 24, which is larger than that required by the optimal scheduling algorithm.

Cheng *et al.* [11] proposed a combined distributed and centralized (CDC) scheme that combines the distributed scheduling and centralized scheduling mechanisms to increase the time slot utilization. Fig. 4 shows the scheduling by applying the scheduling algorithm proposed in [11]. Because the CDC scheme allocates the time slots in an order based on the index of each SS, the link (v_1, v_0) is first allocated eight time slots. Because (v_5, v_4), (v_4, v_3), and (v_3, v_2) do not interfere with (v_1, v_0), the BS allocates three, three, and two time slots to links (v_5, v_4), (v_4, v_3), and (v_3, v_2), respectively. Subsequently, the BS allocates six time slots to (v_2, v_0). Finally, the BS allocates the bandwidth resource for the data transmission of v_3 and reserves one slot each for (v_3, v_2) and (v_2, v_0). Consequently, the number of time slots is 16, which is not optimal.

Han *et al.* [12] and [13] used the concepts of primary and secondary interferences to present an interference model for a WMN. Four policies for determining the order of allocating bandwidth resource to nodes in a mesh network are proposed based on various metrics, as follows: Random, Minimum Interference, Nearest Node to BS First (NNBF), and Farthest Node to BS First (FNBF). The simulation results showed that the policy of NNBF outperforms the other three policies. Therefore, we applied the policy of NNBF to the network shown in Fig. 1(a) and compared its performance with the result of the optimal scheduling. As demonstrated in [12] and [13], if two SSs have the same hop count to the BS, the SS with the smaller node ID has higher priority for allocation of time slots. The scheduling result by applying [12] and [13] is the same as that shown in Fig. 4. Compared with the optimal scheduling, Han *et al.* [12] and [13] require more time slots.

In [14], a heuristic approach was proposed to cope with the admission control and scheduling problems in a WMN. Each link in the network was assigned a label of even or odd according to the layer to which it belongs. An even link transmits in an even time slot, whereas an odd link transmits in an odd time slot. The algorithm proposed in [14] initially attempts to allocate free time slots and subchannels to each flow and to allocate maximal bandwidth to all links belonging to the flow. If the allocated bandwidth of a link is more than the minimum requirement of a flow, the assigned time slots to the link can be used by other flows if required. Fig. 5 shows the transmission schedule when applying the algorithm proposed in [14]. The algorithm initially allocates (v_1, v_0) with eight odd time slots (slots 1, 3, 5, 7, 9, 11, 13, and 15). To avoid interference with flow (v_1, v_0), flow (v_2, v_0) is allocated two, four, six, and eight slots, which are even slots. Because the links belonging to flows (v_5, v_0) and (v_1, v_0) do not interfere with each other, the BS allocates odd time slots to the links belonging to flow (v_5, v_0). Consequently, transmissions of all flows can be completed at time slot 23, which is more than the time required for optimal scheduling.

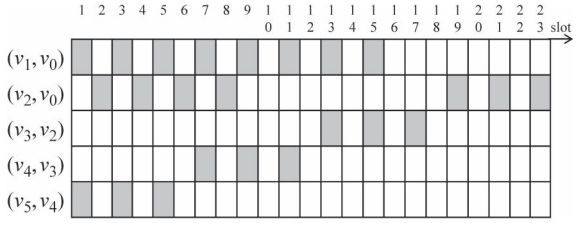


Fig. 5. Schedule by applying the algorithm proposed in [14].

The approaches proposed in [10]–[14] did not completely explore the opportunities of spatial reuse and required more time slots than the optimal scheduling. In the next section, a centralized scheduling algorithm is proposed to arrange an optimal schedule for a WMN. The proposed optimal approach arranges the maximal number of parallel transmissions in each time slot and provides the shortest transmission period for all SSs. Therefore, the proposed optimal approach achieves the shortest delay time and highest throughputs.

III. Optimal Scheduling Algorithm

A. Network Environment and Problem Definition

This paper proposes two scheduling algorithms in a WMN $G = (V, E)$, where V and E denote the set of nodes and the set of links in the mesh network, respectively. The node set $V = \{v_i, 0 \leq i \leq n\}$ comprises $n + 1$ nodes, where v_0 represents the BS, and $\{v_1, \ldots, v_n\}$ represents the n SSs. Let $T = (V, E', R)$ denote the scheduling tree for a given WMN $G = (V, E)$, where $E' \subseteq E$ is the set of links in the scheduling tree, and R is the set of link transmission rates. Let $l_{i,j}$ denote the link between v_i and v_j, where v_j is the parent of v_i in tree T, and $r_{i,j}$ denotes the data transmission rate of $l_{i,j}$. Let d_i denote the bandwidth request of node v_i, which is the amount of data required for transmission per second (bits/s). The set of bandwidth requests maintained by the BS is denoted by $D = \{d_1, d_2, \ldots, d_n\}$. According to a scheduling tree T and a request set D, the BS makes an optimal schedule that assigns the maximal number of transmissions in each slot to

ensure the minimal transmission time required for satisfying the requests of D.

To explore the opportunities of parallel transmissions, we used the primary and secondary interference relations proposed in [12]. The primary interference indicates that a node cannot transmit and receive data simultaneously. This constraint is also referred to as the transmission/reception constraint. The secondary interference indicates that all neighboring nodes, except for the sender, of a receiver cannot transmit data at the time slot when the receiver receives data from the sender. This constraint is also referred to as the interference-free constraint. Based on the concepts of primary and secondary interference constraints, a 2-D collision matrix (CM) is used to specify the interference relations between nodes in the network. The value of entry $CM(i,j)$ is defined by

$$CM(i,j) = \begin{cases} 0, & \text{otherwise} \\ 1, & \text{if } v_i \text{ interferes with } v_j. \end{cases}$$

We assumed that the BS knows the CM information because it has collected mesh network configuration (MSH-NCFG) [1] messages, including the neighbor information of each SS. Consequently, for a given set of SSs, the BS can use CM to determine nodes that can be arranged at the same slot. Mathematically, the problem and objectives of this study were formulated as follows:

In a mesh network, a data flow is defined as a sequence of packets from a source node to a destination. Let $P_{i0} = \{v_i = v_1^{i0}, v_2^{i0}, \ldots, v_k^{i0}, v_0\}$ denote the path of flow from v_i to v_0 (BS). For a given tree T and a request set D, a working set $W_{T,D}$ denotes the set of all SSs that participate in the data transmission for satisfying all requests in D, that is, $W_{T,D} = \bigcup_{1 \le i \le n} P_{i0}$. A scheduling $S_{T,D}$ is valid if a set of time slots T_i is assigned to node v_i to satisfy all requests $d_j \in D$ for all $v_i \in W_{T,D}$. The goal of this study was to find an optimal scheduling S_{opt} to ensure that scheduling S_{opt} is valid and the number of total required time slots is minimized. Let t_i denote the number of slots T_i in $S_{T,D}$. The objective of this study is given as follows:

Objective function

$$\textbf{Minimize } t = \text{Max}_{1 \le i \le n}\, t_i\,.$$

Subject to the collision-free and flow constraints:

1) *Collision-free constraint*

$$T_i \cap T_j = \phi, \qquad \text{if } CM(i,j) = 1$$

where $CM(i,j)$ is the collision relation between nodes v_i and v_j. The value of entry $CM(i,j)$ is one or zero, which indicates whether nodes v_i and v_j have a collision relation. If the value of $CM(i,j)$ is 1, nodes v_i and v_j have collision relation and cannot be assigned a common slot in their schedules. The collision-free constraint guarantees that the schedules of T_i and T_j do not collide.

2) *Flow constraint*: For any two nodes v_j^{i0} and $v_{j'}^{i0} \in P_{i0}$, let t_j^{i0} and $t_{j'}^{i0}$ be the time slots assigned to v_j^{i0} and $v_{j'}^{i0}$ for transmitting the same data, respectively. Hence

$$t_j^{i0} < t_{j'}^{t0}, \qquad \text{if } j < j'\,.$$

The condition $j < j'$ indicates that node v_j is closer to the source node v_i than node $v_{j'}$. Therefore, each packet of a data flow arrives at node v_j earlier than node $v_{j'}$. Therefore, the flow constraint requests that the slot assignment must also follow the order of the packet arrivals.

B. Basic Concepts and Scheduling Rules

This section presents the basic concept and the scheduling rules that are applied in the proposed scheduling algorithms. In a mesh network, it is difficult for the BS to schedule parallel transmissions of multiple flows by considering both collision-free and flow constraints. To guarantee that both constraints can be implemented, the sequential, parallel, and nonparallel relations between SSs are identified. According to the IEEE 802.16d standard, the BS is aware of the network topology. The BS subsequently determines the tree topology based on the network topology. According to the network and tree topologies, the BS can obtain the relations between SSs. The sequential relation is defined for nodes belonging to the same flow, whereas parallel and nonparallel relations are defined for nodes belonging to different flows. Let nodes v_i and v_j belong to the same flow. The sequential relation is defined as follows.

Definition—Sequential relation $(\rightarrow)$*:* Nodes v_i and v_j are said to be sequential and denoted by $v_i \rightarrow v_j$ if v_j is closer to the destination than v_i. □

In another word, if $v_i \rightarrow v_j$, node v_j will help to forward the traffic from v_i. The transmission time allocated to v_j must be later than that allocated to v_i to ensure that the data of v_i is successfully transferred to the BS. For any nodes v_i and v_j belonging to a data flow, the schedule of nodes v_i and v_j must satisfy the relation $v_i \rightarrow v_j$, which indicates that the schedule satisfies the flow constraint.

Let nodes v_i and v_j belong to different flows. The parallel relation between nodes v_i and v_j is defined as follows.

Definition—Parallel relation $(\parallel)$*:* Node v_i is said to be parallel with v_j and denoted by $v_i \parallel v_j$ if they can transmit data at the same time slot without interference at their receiver sides; otherwise, node v_i has a nonparallel relation with node v_j and is denoted by symbol $v_i \nparallel v_j$. □

According to the sequential and parallel relations between SSs, the BS can arrange a valid order of SSs for data transmission in the mesh network.

The following definition extends the relations between two nodes to the relations between two sets.

Definition—Sequential relation $S_a \rightarrow S_b$*:* Two disjoint sets S_a and S_b have a sequential relation and are denoted by $S_a \rightarrow S_b$ if they satisfy $v_i \rightarrow v_j$ for $\forall v_i \in S_a$ and $\forall v_j \in S_h$. □

If two disjoint sets S_a and S_b have a sequential relation, the time slots assigned to $v_i \in S_a$ and $v_j \in S_b$ cannot be overlapped. This occurs because each $v_j \in S_b$ must wait for the data transmitted from each $v_i \in S_a$; hence, the time slots assigned to v_j must be later than the time slots assigned to each

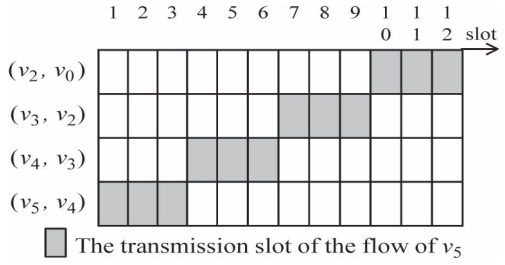


Fig. 6. Valid scheduling that considers the flow from v_5 to BS in Fig. 1(a).

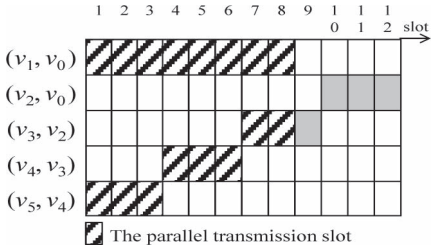


Fig. 7. Parallel scheduling for the example given in Fig. 1(a).

$v_i \in S_a$. Fig. 6 shows an example of scheduling that considers the flow from v_5 to the BS in Fig. 1(a). In this example, both v_4 and v_3 have a sequential relation with v_2, because v_2 is closer to the BS than v_4 and v_3 in the flow. Therefore, the sequential relation $S_a \rightarrow S_b$ occurs, where $S_a = \{v_3, v_4\}$ and $S_b = \{v_2\}$. A valid scheduling occurs when the time slots assigned to v_2 are later than the slots assigned to v_3 and v_4, because v_2 must wait for the data sent from v_3 and v_4.

Assuming that S_a and S_b do not have a sequential relation, a common time slot can be assigned to $v_i \in S_a$ and $v_j \in S_b$ if a parallel relation occurs between v_i and v_j. The parallel relation between two sets is defined as follows.

Definition—Parallel relation $P_a \parallel P_b$: Two disjoint sets P_a and P_b have a parallel relation and are denoted by $P_a \parallel P_b$ if they satisfy $v_i \parallel v_j$ for $\forall v_i \in P_a$ and $\forall v_j \in P_b$. □

Fig. 7 shows a parallel schedule for nodes v_1, v_5, v_4, and v_3 in Fig. 1(a). There exists a parallel relation $P_a \parallel P_b$, where $P_a = \{v_1\}$ and $P_b = \{v_3, v_4, v_5\}$.

In addition, as shown in Fig. 1(a), a nonparallel relation $N_a \nparallel N_b$ occurs, where $N_a = \{v_1\}$ and $N_b = \{v_2\}$. Therefore, the BS cannot assign a common slot to both v_1 and v_2 for satisfying the interference constraint between v_1 and v_2. Two valid schedules that consider the nonparallel relation between v_1 and v_2 are shown in Fig. 8(a) and (b). No overlapping occurs between the slots assigned to v_1 and v_2.

These relations may change because of a new node entry or node failure. If any of these situations occur, the SSs notify the BS of the neighbor change information through the MSH-NCFG [1] messages. The BS can subsequently determine the new tree topology and rebroadcast it to all SSs. After receiving the new bandwidth requests, the BS reschedules the transmission of all nodes by considering the new relations, which are calculated based on the new network and tree topologies. Based on the sequential, parallel, and nonparallel relations, the proposed algorithm creates an optimal schedule for maximizing the bandwidth utilization and minimizing the total transmission time.

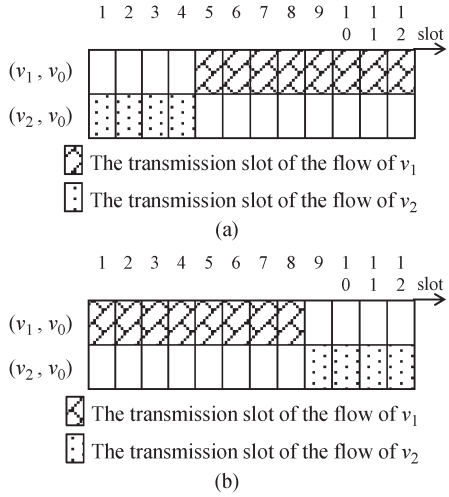


Fig. 8. Scheduling cases consider nonparallel relation for the example given in Fig. 1(a). (a) Schedule case 1. (b) Schedule case 2.

C. SADP

This section introduces the proposed scheduling algorithm with dynamic programming approach (SADP). The proposed algorithm initially schedules each data flow from the source node to the BS by considering the sequential relation. Subsequently, based on the dynamic programming approach, the SADP algorithm merges two different optimal schedules to derive a larger optimal schedule by arranging the transmission order. A large number of possible transmission sequences may occur during the derivation of the larger optimal schedule. The proposed SADP retains the optimal schedule with minimal transmission time. It is worthy to notice that the merging process must satisfy three relations: sequential, nonparallel, and parallel. The SADP repeatedly executes the same merging process until the optimal schedule that contains all data flow is obtained. Because the SADP must record the optimal schedule of each merging process, a structure, which is denoted by U, represents the schedule.

For a given mesh network, the proposed SADP initially schedules each SS to satisfy its bandwidth requirements and guarantee the shortest transmission period. Let d_i denote the bandwidth request of node v_i and $U(\{v_i\})$ denote the optimal schedule for the data transmission along the path from v_i to v_0 (or BS). Consider an optimal schedule $U(\{v_i\})$ for the path P_{i0} from v_i to v_0, where path $P_{i0} = \{v_i = v_1^{i0}, v_2^{i0}, \ldots, v_k^{i0}, v_0\}$ and any two nodes on the path satisfy the sequential relation. Let t_j denote the required data relaying time for $v_j^{i0} \in P_{i0}$. Let notation $\{v_j\}^{t_j}$ denote the schedule of node v_j. Let t_{k+1} denote the total transmission time required for transmitting data from v_i to v_0. We obtain $t_{k+1} = t_1 + t_2 + \cdots + t_k$. The optimal schedule $U(\{v_i\})$ of node v_i can be represented by the structure, as follows:

$$U(\{v_i\}) = \left\{\{v_1^{i0}\}^{t_1} \rightarrow \{v_2^{i0}\}^{t_2} \rightarrow \ldots \rightarrow \{v_k^{i0}\}^{t_k}\right\}^{t_{k+1}} \quad 1 \le i \le n,\ v_1^{i0}, \ldots, v_k^{i0} \in P_{i0} - \{v_0\}. \quad (1)$$

Fig. 9 shows an example to further illustrate the sequential relation and optimal schedule $U(\{v_i\})$. In this example, assume that the amount of data of node v_c is eight units. The data

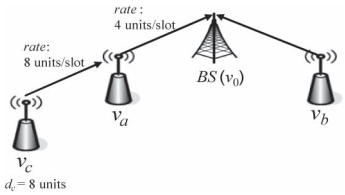


Fig. 9. Tree topology example.

of node v_c would be transmitted to v_0 along the path $P_{c0} = \{v_c, v_a, v_0\}$. Assume that the rates $r_{c,a}$ and $r_{a,0}$ are eight and four units/slot, respectively. Let t_1 and t_2 denote the required transmission times of v_c and v_a, respectively. Therefore, t_1 and t_2 require 1(8/8) and 2(8/4) slots, respectively. Because the time slots allocated to v_c and v_a cannot be overlapped, the total transmission time is $t_k = t_1 + t_2 = 3$. Notation $\{v_c\} \rightarrow \{v_a\}$ indicates that the transmissions for nodes v_c and v_a have a sequential relation. Hence, the optimal schedule of node v_c can be represented by $U(\{v_c\}) = \{\{v_c^{c0}\}^1 \rightarrow \{v_a^{c0}\}^2\}^3$. After the node v_c has been scheduled, it is called a scheduled node.

More than one data flow may be requested for transmitting data to the BS. However, scheduling two or more data flows is more complex than scheduling only one data flow. This section describes the manner in which the two data flows are scheduled and represented by considering the three relations. A data structure of a scheduled node group (SNG) is used to maintain the set of all the scheduled nodes. The elements in an SNG can be extended from a single source node to multiple source nodes. Let $U(SNG)$ denote the optimal schedule of all nodes in the SNG. When the number of elements in an SNG is more than one, the nodes in the SNG may have nonparallel, parallel, and sequential relations. For simplicity, we discussed the nonparallel and parallel relations and ignored the sequential relation because the sequential relation has been discussed in (1). It is worthy to note that a sequential relation occurs if the distance between the source node and the BS is more than one hop. We first discuss the nonparallel relation. Consider two paths, i.e., P_{i0} and P_{j0}. Let the xth node v_x^{i0} on path P_{i0} and the yth node v_v^{j0} on path P_{j0} have a nonparallel relation, and their required transmission times are t_x and t_y, respectively. The optimal schedule of these two nodes is represented by

$$\left\{\left\{v_x^{i0}\right\}^{t_x} \nparallel \left\{v_y^{j0}\right\}^{t_y}\right\}^{t_k} \tag{2}$$

where the total required transmission time $t_k = t_x + t_y$. If v_x^{i0} and v_y^{j0} have a parallel relation, the time slots allocated to v_x^{i0} and v_y^{j0} can be overlapped for $t_{k'}$ slots, where $t_k = \min(t_x, t_y)$. Therefore, the optimal schedule of nodes v_x^{i0} and v_y^{j0} can be represented by

$$\left\{\left\{\left\{v_x^{i0}\right\}^{t_x} \parallel \left\{v_y^{j0}\right\}^{t_y}\right\}^{t_k} \nparallel \{v\}^{t_{k'}}\right\}^{t_{k''}} \tag{3}$$

where v is the node that requires longer transmission time in the parallel relation, and $t_{k'}$ is the extra required transmission time for v after the parallel transmission.

Because the relations among the nodes in the SNG can be complex, the following section discusses the derivation of an optimal scheduling $U(SNG)$ in a systematic manner. In the first step, we discussed the sequential relation and classified all nodes in the SNG into k sets $S_1, S_2, \ldots, S_k$, where any two sets in $\{S_1, S_2, \ldots, S_k\}$ satisfy the sequential relation. Based on the k classified sets, the optimal schedule $U(SNG)$ can be represented by the sequential transmission of the k sets, as shown in (4), where opt(S_i) denotes the optimal transmission schedule of nodes in set S_i. Hence

$$U(SNG) = \left\{\text{opt}\left(S_1{}^{t_1}\right) \rightarrow \text{opt}\left(S_2{}^{t_2}\right) \rightarrow \ldots \rightarrow \text{opt}\left(S_k{}^{t_k}\right)\right\}^{t_{k+1}}. \tag{4}$$

In the following, we further discuss the nonparallel relation between nodes in a set S_i. In the optimal schedule of S_i, the transmission schedules in S_i can be further divided into l sets $N_1, N_2, \ldots, N_l$, where any two sets in $\{N_1, N_2, \ldots, N_l\}$ satisfy the nonparallel relation, as shown in the following equation:

$$\text{opt}(S_i) = \{N_1^{t_1} \nparallel N_2^{t_2} \nparallel \cdots \nparallel N_l^{t_l}\}^{t_{l+1}}, \qquad i \in k. \tag{5}$$

In the optimal transmission schedule of N_j, assume that m nodes are allowed for transmitting data simultaneously. The optimal schedule for each N_j can be represented as shown in the following equation, where $t_{m+1} = \min(t_1, t_2, \ldots, t_m)$. Hence

$$N_j = \{P_1^{t_1} \parallel P_2^{t_2} \parallel \cdots \parallel P_m^{t_m}\}^{t_{m+1}}, \qquad j \in l. \tag{6}$$

The following recursive equations implement the dynamic programming algorithm for calculating the optimal uplink schedule:

$$\begin{aligned} S_{\text{opt}} &= U(SNG) \\ &= \arg \min_{\substack{SNG' \in 2^{SNG} - \Phi \\ SNG' \subseteq SNG.}} U\left(SNG' \cup (SNG - SNG')\right) \end{aligned} \tag{7}$$

Let 2^{SNG} denote the power set of the SNG. The recursive equation (7) derives the optimal schedule of the SNG from the solutions of two smaller sets, i.e., SNG' and $SNG - SNG'$. Consider the example shown in Fig. 9. Let a scheduled group SNG be $\{v_a, v_b, v_c\}$. The optimal solution $U(SNG)$ can be derived based on the following three schedules: $U(\{v_a\} \cup \{v_b, v_c\})$, $U(\{v_b\} \cup \{v_a, v_c\})$, and $U(\{v_c\} \cup \{v_a, v_b\})$. The recursive relation of (7) can be reapplied to derive the optimal solution of $U(\{v_b, v_c\})$, $U(\{v_a, v_c\})$, and $U(\{v_a, v_b\})$ from the optimal solutions of $U(\{v_a\})$, $U(\{v_b\})$, and $U(\{v_c\})$, which are optimal schedules for the transmission from the single node to the BS.

Let symbol $\oplus$ denote the basic operation for deriving the optimal schedule. The following equation copes with the problem of using two optimal schedules, i.e., $U(SNG_i)$ and $U(SNG_j)$, to derive an optimal schedule for node set $SNG_i \cup SNG_j$. Hence

$$U(SNG_i \cup SNG_j) = U(SNG_i) \oplus U(SNG_j). \tag{8}$$

After the deriving process, the SADP can combine the optimal transmission schedules of two smaller node sets, i.e., SNG_i and SNG_j, and derive the optimal schedule for a

larger set, i.e., $SNG_i \cup SNG_j$. The following further discusses the details of operation $\oplus$. Consider the following optimal schedules:

$$U(SNG_i) = \left\{\text{opt}\left(S_1^i\right) \to \text{opt}\left(S_2^i\right) \to \ldots \to \text{opt}\left(S_m^i\right)\right\}$$
$$U(SNG_j) = \left\{\text{opt}\left(S_1^j\right) \to \text{opt}\left(S_2^j\right) \to \ldots \to \text{opt}\left(S_n^j\right)\right\}.$$

To construct the optimal solution of $U(SNG_i \cup SNG_j)$, the parallelization and sequencing tasks must be executed on sets SNG_i and SNG_j. The parallelization task aims to extremely exploit the parallel transmissions between sets SNG_i and SNG_j, whereas the sequencing task aims to guarantee that the original sequential relations in SNG_i and SNG_j are also valid in the optimal schedule of $U(SNG_i \cup SNG_j)$. When executing the parallelization task, transmission schedule $N_i \parallel N_j$ must be extremely exploited for any $N_i \in S_x^i$ and $N_j \in S_y^j$ under the following criteria:

$$CM(P_k, P_{k'}) = 0, \qquad \forall P_k \in N_i \text{ and } \forall P_{k'} \in N_j\,.$$

Let $N_i(t)$ denote the transmission time of N_i. An exploitation of parallel transmission $N_i \| N_j$ can reduce the total transmission time from $N_i(t) + N_j(t)$ to $\min(N_i(t), N_j(t)) + |N_i(t) - N_j(t)|$.

In addition to the parallelization task, the sequencing task must be applied on sets SNG_i and SNG_j to guarantee that the optimal schedule $U(SNG_i \cup SNG_j)$ can maintain the sequential relations in SNG_i and SNG_j. Therefore, for any schedules $N_j \in S_x^i$ and $N_{j'} \in S_{x'}^i$, if $x < x'$, N_j and $N_{j'}$ have a sequential relation; the slots allocated to N_j must be arranged before the slots allocated to $N_{j'}$.

Based on the tree shown in Fig. 1, an example is provided to illustrate the merging procedure of individual schedules of nodes v_1 and v_5. Based on the tree topology, the quality of each link, and the bandwidth request of each node, as shown in Fig. 1, the BS can obtain the optimal schedule of each node of v_1 and v_5. Hence

$$U(\{v_1\}) = \{v_1^{10}\}^8$$
$$U(\{v_5\}) = \left\{\{v_5^{50}\}^3 \to \{v_4^{50}\}^3 \to \{v_3^{50}\}^3 \to \{v_2^{50}\}^3\right\}^{12}.$$

The optimal schedule of $U(\{v_1\})$ contains one subschedule $\{v_1^{10}\}^8$, whereas the optimal schedule of $U(\{v_5\})$ comprises four subschedules, i.e., $\{v_5^{50}\}^3$, $\{v_4^{50}\}^3$, $\{v_3^{50}\}^3$, and $\{v_2^{50}\}^3$, which have the sequential relation. Notice that the sequential relation must be maintained in a merged schedule. To exploit the opportunities of a parallel relation, the BS examines whether the subschedule $\{v_1^{10}\}^8$ of $U(\{v_1\})$ and each of the subschedules of $U(\{v_5\})$ can be allocated at the same time slots. Because subschedules $\{v_1^{10}\}^8$ and $\{v_5^{50}\}^3$ satisfy the parallel relation, the merged schedule can be recorded as $\{\{v_1^{10}\}^8 \| \{v_5^{50}\}^3\}^8$, and the total transmission time of which is eight slots. This result is shown in Fig. 10.

Because the transmission time of subschedule $\{v_1^{10}\}^8$ is larger than that of $\{v_5^{50}\}^3$, $\{v_1^{10}\}^8$ can be further examined to merge more subschedules with a parallel relation. The five null slots shown in Fig. 10 represent the capacity for merging more subschedules.

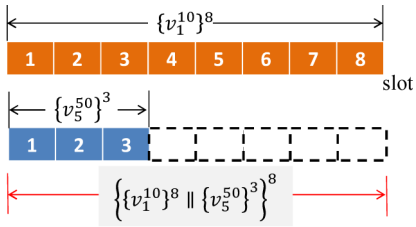


Fig. 10. Example of the merged schedule $\{\{v_1^{10}\}^8 \| \{v_5^{50}\}^3\}^8$.

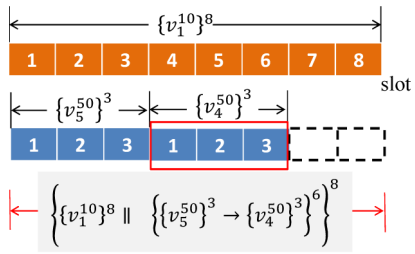


Fig. 11. Example of the merged schedule $\{\{v_1^{10}\}^8 \| \{\{v_5^{50}\}^3 \to \{v_4^{50}\}^3\}^6\}^8$.

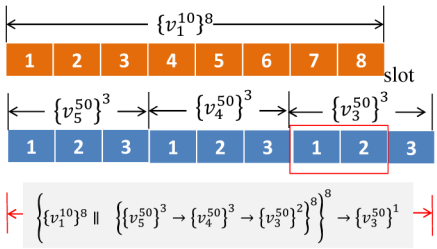


Fig. 12. Two null slots can be allocated to the subschedule $\{v_3^{50}\}^3$ because node v_3 has parallel relation with v_1.

As shown in Fig. 11, three of the five null slots can be allocated to subschedule $\{v_4^{50}\}^3$ because it has a parallel relation with $\{\{v_1^{10}\} \| \{v_5^{50}\}^3\}^8$. As shown in Fig. 11, the new merged schedule is $\{\{v_1^{10}\}^8 \| \{\{v_5^{50}\}^3 \to \{v_4^{50}\}^3\}^6\}^8$.

Fig. 11 shows two null slots that can be assigned for the subschedule $\{v_3^{50}\}^3$ because it has a parallel relation with $\{v_1^{10}\}^8$. Fig. 12 shows the overall merging result, which can be represented by

$$\left\{\{v_1^{10}\}^8 \parallel \left\{\{v_5^{50}\}^3 \to \{v_4^{50}\}^3 \to \{v_5^{50}\}^2\right\}^8\right\}^8 \to \{v_3^{50}\}^1.$$

Because the subschedule $\{v_2^{50}\}^3$ has a sequential relation with $\{v_3^{50}\}^3$, it is arranged to be executed sequentially after the schedule $\{v_3^{50}\}^3$, as shown in Fig. 12. The merging result is shown in Fig. 13. Finally, the optimal schedule $U(\{v_1, v_5\})$ can be obtained by merging $U(\{v_1\})$ and $U(\{v_5\})$. Hence

$$U(\{v_1, v_5\}) = \left\{\left\{\{v_1^{10}\}^8 \parallel \left\{\{v_5^{50}\}^3 \to \{v_4^{50}\}^3 \to \{v_3^{50}\}^2\right\}^8\right\}^8 \to \{v_3^{50}\}^1 \to \{v_2^{50}\}^3\right\}^{12}.$$

The total transmission time is 12 slots.

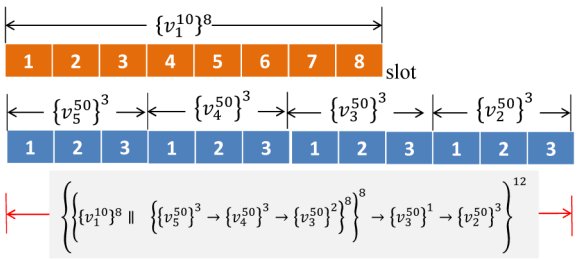


Fig. 13. Example of optimal schedule $U(\{v_1, v_5\})$, which can be obtained by merging $U(\{v_1\})$ and $U(\{v_5\})$.

Input : A graph $G=(V, E)$, $V=\{v_1,\ldots, v_n\}$, $D=\{d_1,\ldots, d_n\}$, a scheduling tree $T=(V, E')$ and a matrix CM

Output : An optimum scheduling $U(SNG)$

1. $i=0, j=0, k=0$;

2.$D[1\ldots n]$[subset of SNG] = $\varnothing$

3.**for** ($i=1$; $i<=n$ i++)

$D[1][\{ v_i \}] = \left\{\{v_1^{i0}\}^{t_1} \rightarrow \{v_2^{i0}\}^{t_2} \rightarrow \ldots \rightarrow \{v_k^{i0}\}^{t_k}\right\}^{t_{k+1}}$

// v_k^{i0} is the parent of v_{k-1}^{i0} on P_{i0}.

4.**end for**

5. **for** ($k=2$; $k<= n$; k++)

6. **for** ($j=1$; $j<=k$; j++)

7. **for** (each subset $A \subseteq SNG'$ and $|A| = j$, where SNG' contains k nodes)

8. $D[k][SNG'] = \arg\min(D[j][A] \oplus D[k-j][SNG'-A])$

9. **end for**

10. **end for**

11. **end for**

12. **return** $D[n][SNG]$

Fig. 14. Proposed SADP.

Fig. 14 shows the detailed steps of the proposed SADP algorithm. In step 3, the optimal schedule for the data transmission from a single node v_i to the BS along the path embedded in the scheduling tree is constructed and stored in the array $D[1][\{v_i\}]$. Based on (8) and the information stored in $D[1][\{v_i\}]$, the optimal transmission schedule of any two nodes v_i and v_j can be calculated and stored in $D[2][\{v_i, v_j\}]$. The operation for constructing the optimal transmission schedule for a larger set can be calculated based on the optimal schedules of smaller sets that are previously calculated. Consequently, the optimal schedule of SNG' containing k nodes can be obtained using $A \oplus SNG' - A$, where A is each subset of SNG', and the calculation results are stored in $D[k][SNG']$, as shown in step 8. When the value of k reaches n, the final optimal schedule of the mesh network can be obtained and stored in $D[n][SNG]$, as shown in step 8.

IV. HSA

The SADP algorithm applies the dynamic programming approach and can attain an optimal transmission schedule. However, the computational complexity of SADP is high. This section introduces the HSA, which has lower complexity and can achieve similar performance to the SADP algorithm.

Several prior studies considered only the parallel transmission to enhance the network throughput. However, a large number of packets transmitted to the SS neighboring the BS is buffered, because the BS can only receive data from one of its neighbors at a time. This situation blocks the data transmission in SSs that neighbor the BS and increases the end-to-end delay. To avoid the traffic congestion problem at these SSs, the proposed HSA always schedules these SSs first. The HSA considers the traffic congestion problem and the parallel transmission. When the BS assigns a slot to one link, the HSA simultaneously assigns the same time slot to as many links as possible to exploit the parallel transmission opportunities.

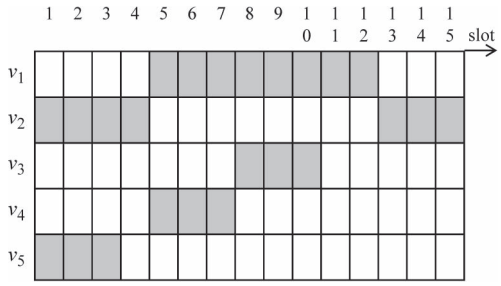


Fig. 15. Schedule by applying the HSA on the tree given in Fig. 1.

The proposed HSA schedules the transmission of SSs slot by slot. The uplinks directly connected to the BS are scheduled first because these links encounter congestion easily. In general, the required transmission time of a scheduling tree increases with the length of the longest path. To reduce the total transmission time, the proposed HSA initially considers the flow with maximal length. Let $P_{\max}$ denote the flow path with maximal length in the tree. The BS first schedules the closest link of $P_{\max}$. If no data arrives or is generated from the node of this link, the link will not be assigned a slot because of a lack of a transmission request. In this case, the BS randomly chooses one link from other links closest to the BS. Afterward, the BS subsequently schedules the farthest link of $P_{\max}$ to start the flow of path $P_{\max}$ as soon as possible. This policy aims to reduce the congestion near the BS and minimize the maximal transmission time. The other links belonging to $P_{\max}$ are subsequently scheduled in an order from the nearest SS to the farthest SS. In the scheduling of each slot, the maximal number of possible links not belonging to $P_{\max}$ must be assigned to maximize the number of parallel transmissions. To reduce the congestion phenomenon, the link with the least distance to the BS is scheduled first.

Based on the tree topology shown in Fig. 1(a), Fig. 15 shows an example to illustrate the basic concept of HSA. Because the flow of v_5 has the longest path $P_{5,0} = \{v_5, v_4, v_3, v_2, v_0(\text{BS})\}$ in the tree, the link (v_2, BS) closest to the BS is scheduled first. As shown in Fig. 15, slot 1 is first allocated to link (v_2, BS). The HSA subsequently allocates time slot 1 to link (v_5, v_4), which is the last link of the longest path $P_{5,0}$. The next step is to schedule links that can be simultaneously transmitted with the scheduled links, i.e., (v_2, BS) and (v_5, v_4). In this step, other links cannot be scheduled because they interfere with the previously scheduled links. After scheduling slot 1, the proposed HSA schedules the links of $P_{\max}$ again. Because node v_5 has remaining data and path $P_{5,0}$ is the longest path compared with the paths of other flows, the links on the path $P_{5,0}$ are rescheduled based on the concept of the proposed HSA mechanism. Consequently, slot 2 is assigned to links (v_2, BS) and (v_5, v_4).

Input : Tree T
Output : A bandwidth scheduling S
Initializa tion :
$t = 1$
Ω is the set of all $U(v_i)$
Scheduling :
1. **while** Ω is not empty
2. Unmark all nodes
3. Compare the path length of all flows and set the longest path of $U(v_i)$ as P_{max}
4. Consider the node v_j nearest to BS on P_{max}, $U(v_i)$ and each child node v_k of v_j // v_j is the $1-$hop node
5. **if** $check\_S-relation(v_j, U(v_i))$ or $check\_S-relation(v_j, U(v_k))$
// Check whether or not node v_j has data to transmit
6. Allocate time slot t to v_j;
7. Update the transmissi on time of $U(v_i)$ or $U(v_k)$ and mark v_j;
8. **else**
9. Consider other nodes nearest to BS and select one node v_l and each child node v_k of v_l
// other $1-$hop nodes
10. **if** $check\_S-relation(v_l, U(v_k))$ // Check whether or not node v_l has data to transmit
11. Allocate time slot t to v_l
12. Update the transmissi on time of $U(v_k)$ and mark v_l;
13. **end if**
14. **end if**
15. **if** $check\_P-relation(v_l)$ and $check\_S-relation(v_l, U(v_l))$
// Check whether or not node v_l can transmit without collission // v_l *is the farthest node on* P_{max}
16. Allocate time slot t to node v_l
17. Update the transmissi on time of $U(v_l)$ and mark v_l;
18. **end if**
19. **for** each unmarked node $v_m \in P_{\max}$
20. Select node v_m from highest $L(v_m)$ to lowest $L(v_m)$ and consider each child node v_n of v_m
21. **if** $check\_P-relation(v_m)$ and $check\_S-relation(v_m, U(v_n))$
// Check whether or not node v_m has data to transmit without collission
22. Allcate time slot t to node v_m
23. Update the transmissi on time of $U(v_n)$ and mark v_m;
24. **else**
25. Mark v_m;
26. **end if**
27. **end for**
28. **for** each unmarked node $v_o \notin P_{\max}$
29. Select a node v_o from lowest $L(v_o)$ to highest $L(v_o)$ and consider each child node v_p of v_o
30. **if** $check\_P\_relation(v_o)$ and $check\_S-relation(v_o, U(v_p))$
31. Allcate time slot t to node v_m
32. Update the transmissi on time of $U(v_p)$ and mark v_o;
33. **else**
34. Mark v_o;
35. **end if**
36. **end for**
37. **while** there exists a flow of node v_j which the transission time of $U(v_j)$ euqal to 0
38. // Remove each well – scheduled flow
39. Remove $U(v_j)$ from Ω;
40. **end while**
41. $t++$;
42. **end while**

Fig. 16. Pseudocode of the proposed HSA.

Following this schedule, node v_5 transmits its entire data to node v_5 after slot 3. Hence, the longest path changes from $P_{5,0}$ to $P_{4,0}$. For path $P_{4,0}$, link (v_2, BS) is the link nearest to the BS. Therefore, the proposed HSA allocates slot 4 to link (v_2, BS). All other links cannot be scheduled because they interfere with link (v_2, BS). For slot 5, links on path $P_{4,0}$ are rescheduled in advance. Although link (v_2, BS) has the highest priority, no traffic can be transmitted through this link at slot 5. Therefore, the BS chooses one link from the links closest to the BS. Slot 5 is assigned to link (v_1, BS). Subsequently, the BS allocates slot 5 to link (v_4, v_3), which is the farthest link of the longest path. The similar procedure repeats until all requirements are scheduled.

Fig. 16 shows the pseudocode of the proposed HSA. Let $L(v_x)$ denote the level of v_x in the path. Steps 3–14 schedule the link closest to the BS, whereas steps 15 and 18 schedule the last link belonging to the longest path $P_{\max}$. Steps 19–27 schedule the links belonging to $P_{\max}$ that can be simultaneously transmitted with the scheduled links. Steps 28–36 schedule the remaining links that can be simultaneously transmitted with the scheduled links. Steps 37–42 remove the well-scheduled flows and start another loop to repeatedly schedule the remaining links. The $Check\_S\_relation(v_y, U(v_x))$ aims to examine whether node v_y can be scheduled for forwarding the flow of $U(v_x)$. The $Check\_P\_relation(v_y)$ checks whether node v_y can be scheduled for parallel transmissions.

TABLE I
COMPARISON OF RELATED STUDIES

Features	Spatial Reuse	Various Link Rate	Delay Optimum	Time complexity
H. Shetiya [10]	N	Y	N	$n\times\xi^2$
S. M. Cheng [11]	Y	N	N	n^3
B. Han [12][13]	Y	Y	N	n^2
D. Ghosh [14]	Y	N	N	$n^2\times\xi$
Our proposed algorithm- SADP	Y	Y	Y	$n^2\times2^n$
Our proposed algorithm- HSA	Y	Y	N	$\xi\times n$

Table I shows a comparison of the existing scheduling algorithms [10]–[14] with the proposed SADP and HSA algorithms in terms of spatial reuse, various link rate, delay optimum, and time complexity. The proposed optimal approach considers spatial reuse, link rate, and delay simultaneously, which result in a superior schedule to that of existing algorithms. The proposed algorithm (SADP) shown in Fig. 14 mainly consists of three loops. The first and second loops are executed at most n times, where n is the number of SSs in the network. Therefore, the time complexity is $O(n^2)$. For the third loop, the SADP calculates the transmission schedule of each SNG', which contains k nodes. Because the number of possible SNG' is C_k^n, the complexity of loop 3 is 2^n. Consequently, the total time complexity of the proposed SADP is $O(n^{2*}2^n)$. The HSA algorithm achieves similar performance to the SADP; however, it has lower computational complexity. As shown in Fig. 16, the proposed HSA contains two loops. Let notation ξ denote the number of slots in a scheduling period T. The first loop is repeatedly executed until the requirements of all nodes are satisfied or all time slots run out. The time complexity of the first loop is $O(\xi)$. The inferior case of the second loop occurs when n nodes are required to be scheduled. Its time complexity is $O(n)$. Consequently, the total time complexity of the proposed HSA is $O(\xi \times n)$. The comparison of the existing studies and the proposed algorithms in terms of time complexity is shown in Table I.

V. PERFORMANCE STUDY

This section presents the performance evaluation of the proposed SADP and HSA algorithms with the existing approaches [13], which are referred to as NNBF, FNBF, and Node with Max Rate First (NMRF). The NNBF and FNBF first schedule the link whose station is closer and farther to the BS, respectively. The NMRF scheme first schedules the link with a higher data rate. Table II gives the modulation and coding scheme [1] applied in the simulation.

TABLE II
PARAMETERS OF BURST PROFILE

Burst profile No.	Modulation	Uncoded bytes per OFDM symbol	Uncoded bytes per minislot	Overall coding rate
1	BPSK	12	24	1/2
2	QPSK	24	48	1/2
3	QPSK	36	72	3/4
4	16-QAM	48	96	1/2
5	16-QAM	72	144	3/4
6	64-QAM	96	192	2/3
7	64-QAM	108	216	3/4

The region size of the considered environment was set to 500 $\times$ 500 units. One BS was deployed at the center of the service region. The simulation applied two distributions to model the placement of SSs in the considered environment. The uniform distribution, which uniformly deploys SSs in the considered environment, was applied in the first model. To observe the interference effect on the performance of each compared scheduling algorithm, the congregating distribution, which constrains all SSs, was deployed within the central area of 400 $\times$ 400 units in size in the second model. The number of SSs in the mesh network varied from 5 to 25. The BS and SSs were clock synchronized. The bandwidth request of each SS varied from 1 to 3 Mb/s. In the simulation, the system performed the request update every 100 frames. All SSs had a common transmission range. In the physical layer, each node accessed the same channel and used the orthogonal frequency-division multiplexing (OFDM) burst profiles, as shown in Table II.

Fig. 17(a) shows a scenario of the considered environment with uniform distribution. The x-axis and y-axis, ranging from 0 to 500 units, represent the coordinate system of the service region. Each unit represents 100 m. The location of the BS was set at the center of the region and marked by a black rectangle symbol. Twenty-five SSs were uniformly deployed in the area. The z-axis denotes the traffic requirement of each SS. As shown in Fig. 17(a), each SS applies a proper modulation based on the transmission distance and signal-to-noise ratio value. Fig. 17(b) shows 25 SSs placed according to the congregating distribution. Compared with Fig. 17(a), the distances between two SSs and the SSs and BS reduced. Hence, SSs can apply superior modulation, as shown in Fig. 17(b).

Fig. 18 shows a comparison of the proposed SADP and HSA with the other three algorithms in terms of the average throughput. As shown in Fig. 18, the SSs are deployed with uniform distribution. The network throughput generally increases with the number of nodes when the network capacity can support the required bandwidth requests. When the bandwidth requests approach the upper bound of the network capacity, the network throughput increases slowly. The existing algorithms, including NNBF, FNBF, and NMRF may arrange for all children of the BS to receive data at the same time slot, resulting in a situation in which the BS is unable to receive any data from its children at that slot. The proposed SADP applies dynamic programming and can avoid this situation. It arranges for one child to send data to the BS and a few other children to receive data at the same time slot. This reduces the number of time slots required by the SADP, which increases the network throughput.

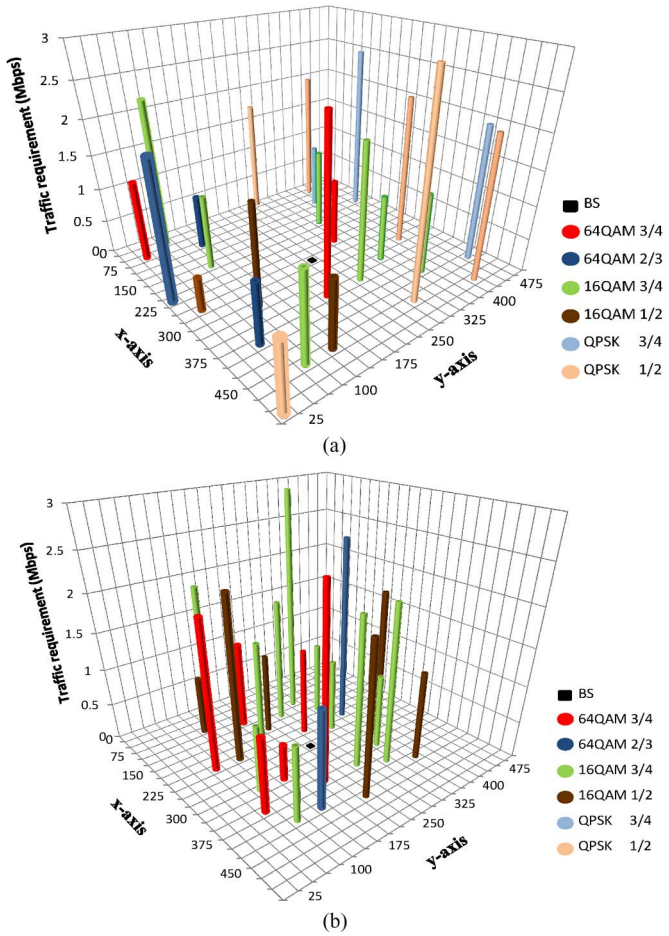


Fig. 17. Two distributions applied in the considered environment. (a) SSs are uniformly deployed in the serving area. (b) SSs are placed by applying the congregating distribution, which constrains that all SSs are located within the central area with size 400 $\times$ 400 units.

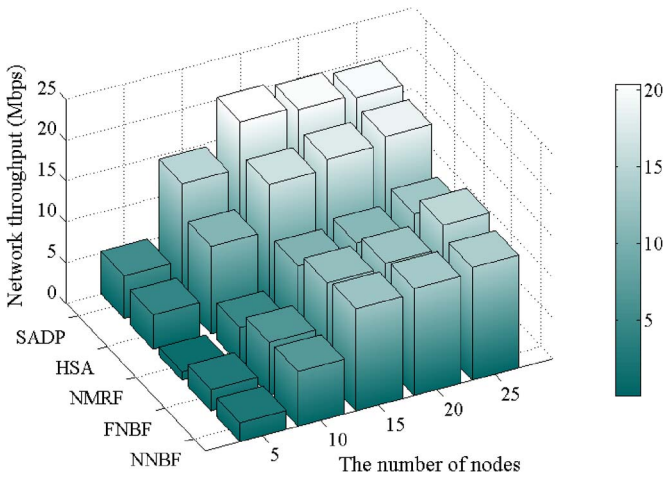


Fig. 18. Comparison of the network throughput of the five compared algorithms with uniform distribution of SSs.

Consequently, the proposed SADP outperformed the other four algorithms, with regard to network throughput, in all cases. In addition, the performance result of the proposed HSA is closer to that of the proposed SADP. This occurred because the proposed HSA exploits more opportunities for parallel transmissions, which reduces the transmission delay for flows

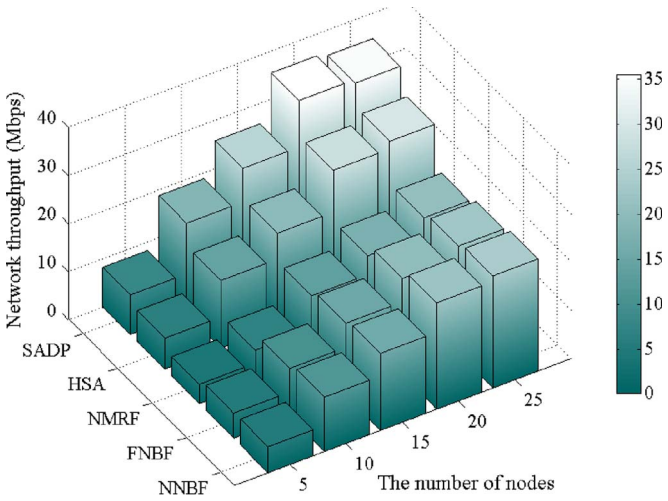


Fig. 19. Comparison of the proposed SADP and HSA algorithms and the other three existing algorithms in terms of the network throughput under congregating distribution.

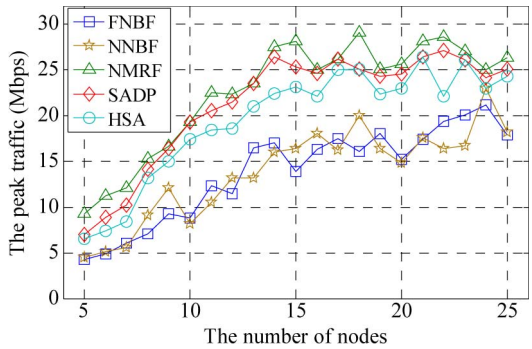


Fig. 20. Comparison of the five algorithms in terms of the peak traffic by applying uniform distribution to place SSs.

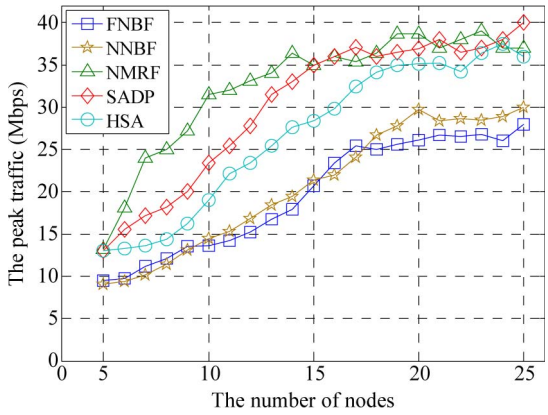


Fig. 21. Comparison of the five algorithms in terms of the peak traffic by applying congregating distribution as the deployment policy.

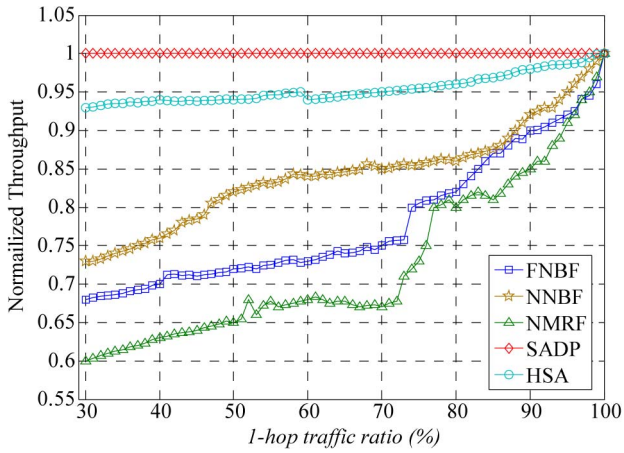


Fig. 22. Comparison of the proposed SADP and HSA and the other three schemes in terms of the normalized throughput by varying the one-hop traffic ratio under uniform distribution.

that have a larger hop count to the BS. In general, the proposed SADP and HSA outperformed the other three algorithms in terms of network throughput.

Instead of applying uniform distribution, Fig. 19 shows the application of the congregating distribution to determine the location of each SS. As shown in Fig. 17(b), the SSs are deployed closer to the BS compared with the deployment of SSs in Fig. 17(a). This also indicates that the SSs in Fig. 19 have superior modulation and a higher transmission rate. Consequently, Fig. 19 shows superior performance to Fig. 18 in terms of network throughput. In addition, the proposed SADP exhibited superior performance to the other four schemes when the number of nodes increased to 25. Generally, the proposed SADP and HSA outperformed the other three algorithms under congregating distribution.

Fig. 20 shows a comparison of the proposed SADP and HSA algorithms with the existing three algorithms in terms of the peak throughput. In the simulation, the number of nodes ranged from 5 to 25. The traffic of each node was randomly selected as 1, 2, or 3 Mb/s. As shown in Fig. 20, all curves generally grow when the number of nodes increases. The NMRF algorithm outperforms the other four scheduling algorithms because it always schedules the link with a higher data rate in advance. Although the NMRF algorithm achieves maximal peak throughput at a few slots, the links with lower data rates may interfere with each other and restrict the degree of parallel transmissions in the remaining time slots. The proposed SADP exploits the opportunities of parallel transmissions and arranges more possible transmissions simultaneously. Consequently, the peak traffic of SADP approaches that of FNBF. The proposed HSA exhibited superior performance to the NNBF and FNBF algorithms in terms of the peak throughput. This occurs because the proposed HSA schedules the links with the smallest and largest hop counts in advance. This alleviates the congestion phenomenon that occurs near the BS. Fig 21 shows the application of the congregating distribution to the node placement. The proposed SADP and HSA outperformed the NNBF and FNBF with an improvement of 15%–20% in terms of the peak traffic.

Fig. 22 shows the varying one-hop traffic ratio, which refers to the ratio of traffic initiated from the one-hop neighbors of the BS to the traffic initiated from all nodes. The proposed SADP and HSA algorithms were compared with the three existing algorithms in terms of normalized network throughput. The performance results were normalized to ensure that the proposed SADP algorithm maintains a constant value of 1. The one-hop traffic ratio ranged from 30% to 100%. The number of nodes was 25. As shown is Fig. 22, the network throughput generally increases with the one-hop traffic ratio.

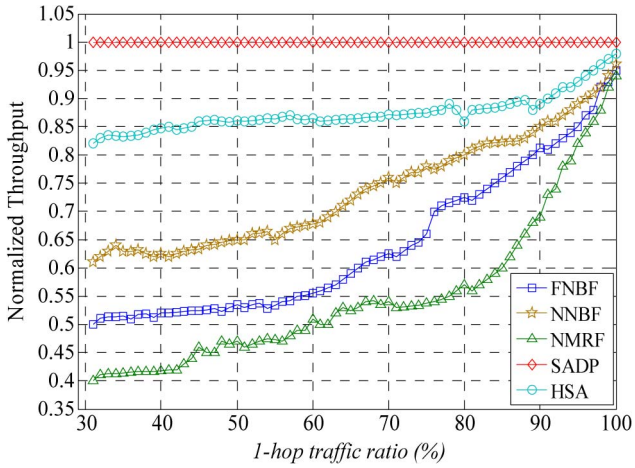


Fig. 23. Normalized throughput versus one-hop traffic ratio by applying the congregating distribution as the deployment policy.

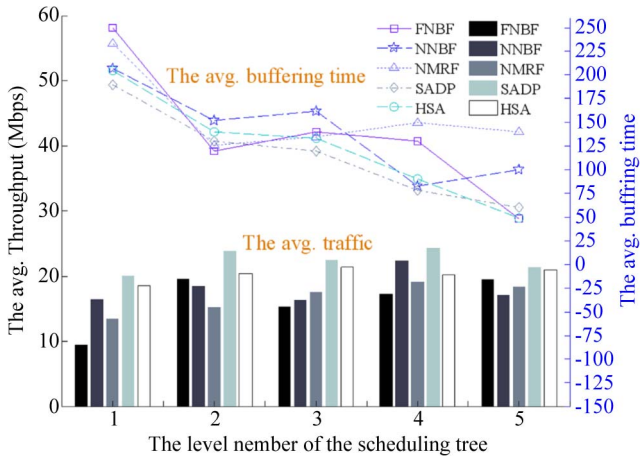


Fig. 24. Comparison of the five algorithms in terms of the average throughput achieved on each level and the average buffering time by applying uniform distribution as the deployment policy.

This occurs because the nodes nearest to the BS can directly transmit data to the BS; thus, the network throughput can be easily increased without data forwarding. Consequently, the BS can continuously receive data from its neighbors when the one-hop traffic ratio increases. In general, the proposed SADP outperforms all heuristic algorithms in terms of network throughput. This occurs because the proposed SADP can obtain the optimal scheduling even if the unbalanced traffic happened in the network. In addition, the performance of the proposed HSA outperforms the other three algorithms because the HSA schedules the stations closest to the BS in advance, which reduces the occurrence of the congestion problem. In addition, the throughputs of the FNBF and NMRF algorithms increase rapidly when the one-hop traffic ratio varied from 70 to 100. This occurred because of the considerable interference that often occurs at the two-hop neighbors of the transmitting node. Hence, the higher one-hop traffic ratio can reduce numerous forwarding transmissions to avoid congestion.

Fig. 23 shows a comparison of five scheduling algorithms by applying congregating distribution as the policy of node placement. Because the distances between SSs are reduced, the interference has a greater effect on the performance of network throughput. The proposed SADP exhibited superior performance to the other scheduling algorithms in the interference-rich environment. This occurred because the proposed SADP applies dynamic programming to obtain the optimal solution. Because the performance of the compared five algorithms were normalized based on the performance of the SADP, the normalized throughput of the proposed HSA and other existing algorithms, including FNBF, NNBF, and NMRF, slowly increased.

Fig. 24 shows the performance of the five compared scheduling algorithms in terms of the average buffering time of arrived packets and the average throughput achieved by each level in the scheduling tree. The transmission requests of each node ranged from 1 to 3 Mb/s. Let the packet generation rate denote the number of packets generated by each node per second. The packet generation rate of each node ranged from 10 000 to 30 000 packets per second (pps). The packet size was 100 bytes. A packet was dropped to avoid the buffer overflow problem when its buffering time was more than 250 ms. Let ζ_i^{pass} denote the amount of data forwarded by node v_i. Let ζ_i^{gen} denote the amount of data generated from node v_i. Let ζ_i denote the total amount of data in the buffer of node v_i. The following expression evaluates the value of ζ_i:

$$\zeta_i = \zeta_i{}^{\text{pass}} + \zeta_i{}^{\text{gen}}. \tag{9}$$

Let a Boolean variable $\sigma_{i,k}^{\text{out}}$ indicate whether the kth bit in the buffer of node v_i is successfully transmitted, as shown in (10) at the bottom of the page.

Let Boolean variable $\sigma_{i,k}^{\text{drop}}$ indicate whether the kth bit in the buffer of node v_i is dropped, as shown in (11) at the bottom of the page.

Let $t_{i,k}^{\text{in}}$ and $t_{i,k}^{\text{out}}$ denote the arrival time and departure time of the kth bit of the node v_i, respectively. Let $t_{i,k}^{\text{drop}}$ denote the dropping time of the kth bit of the node v_i. Let T_{height} and L_h

$$\sigma_{i,k}^{\text{out}} = \begin{cases} 1, & \text{if the } k\text{th bit in the buffer of node } i \text{ is transmitted successfully} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$$\sigma_{i,k}^{\text{drop}} = \begin{cases} 1, & \text{if the } k\text{th bit in the buffer of node } i \text{ is dropped} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

denote the height of the scheduling tree and the set of nodes on the hth level of the scheduling tree, respectively. Let ω_h denote the average buffering time of level h. The following equation shows the derivation of ω_h:

$$\omega_h = \frac{\sum_{i \in L_h} \left(\sum_{k=1}^{\zeta_i} \left(t_{i,k}^{\text{out}} - t_{i,k}^{\text{in}} \right) \times \sigma_{i,k}^{\text{out}} + \sum_{k=1}^{\zeta_i} \left(t_{i,k}^{\text{drop}} - t_{i,k}^{\text{in}} \right) \times \sigma_{i,k}^{\text{drop}} \right)}{\sum_{i \in L_h} \zeta_i}$$
$$1 \le h \le T_{\text{height}}. \tag{12}$$

As shown in Fig. 24, the value of T_{height} was set at 5. In general, the buffering time ω_1 is larger than ω_i, i.e., $2 \le i \le 5$. This occurred because the BS is the common destination of all data flows; however, the BS cannot receive multiple data from various nodes at the same slot. Therefore, the data are buffered for a long time at the nodes in L_1. Fig. 24 shows that the proposed SADP outperforms the other four scheduling algorithms in terms of average buffering time. This occurs because the proposed SADP arranges each node to transmit data in suitable time slots and requires the least transmission time to complete all traffic requirements. In addition, the average buffering times ω_1, ω_3, and ω_5 of the proposed HSA are closer to the proposed SADP algorithm than those of the NNBF and NMRF algorithms. This occurs because the proposed HSA schedules the links with the smallest and largest hop counts in advance to alleviate the traffic congestion that occurs near the BS and reduce the transmission delay. Consequently, the proposed HSA has smaller buffer delay at the nodes in L_5 than the other algorithms. The FNBF schedules the node with the largest path length in advance, which reduces the average buffer delay of the nodes in L_5. However, the transmissions of the nodes in L_1 are blocked, resulting in a large value of ω_1.

Fig. 24 also shows a comparison of five algorithms in terms of the average throughput achieved by each tree level. The average throughput achieved by each level was measured every second. As shown in Fig. 24, all data can be transmitted rapidly by applying the proposed SADP. Furthermore, because the proposed HSA first allocates the time slot for the nodes on the first and fifth levels, the first and fifth levels have superior throughput than the other levels. The large amount of throughput achieved on the first level indicates that the data can be successfully transmitted to the BS.

It is worth to notice that the traffic congestion occurs on level i when the amount of data of level $i+1$ is larger than that of level i. As shown in Fig. 24, the FNBF algorithm results in traffic congestion on the first level, thereby reducing the amount of data received by the BS. The NNBF encounters the same problem on the third level. In general, the proposed HSA exhibits superior performance than the other algorithms in terms of average throughput achieved on each level.

Let MTT delay denote the maximal tolerable transmission delay for a data flow from a source node to the BS. Fig. 25 shows the effect of MTT delay on the packet dropping ratio. The MTT delay ranged from 1 to 500 ms. The packet dropping ratio generally decreases with the MTT delay. The proposed SADP initially schedules the packet transmission flow by flow. Based on each flow schedule, the SADP exploits maximal

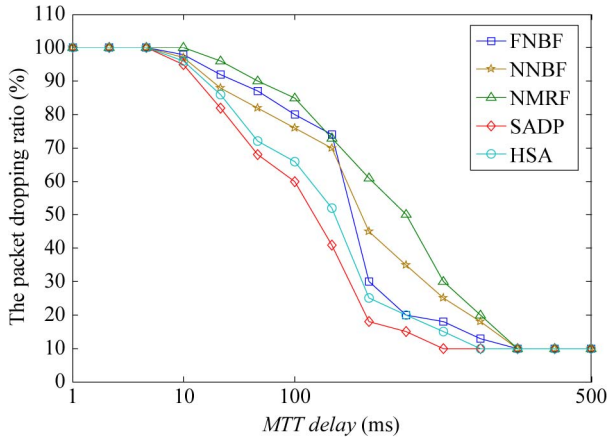


Fig. 25. Packet dropping ratio by varying MTT delay.

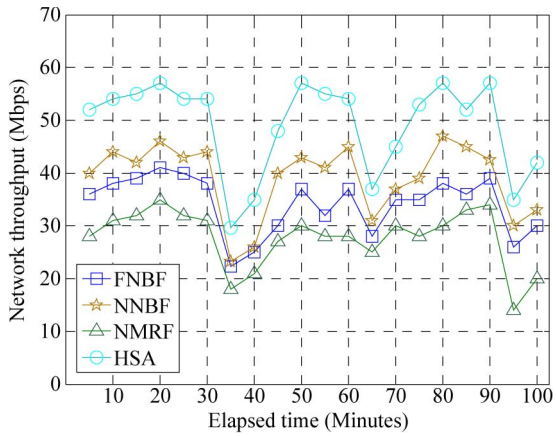


Fig. 26. Comparison of the four algorithms in terms of the network throughput. The transmission rate of each link is varied because the signal is corrupted by adding Gaussian noise for 10 min at 30, 60, and 90 min.

parallelism and avoids interference between neighboring nodes. Therefore, all flows can be scheduled with minimal uplink transmission time. Consequently, the proposed SADP algorithm outperforms the other four scheduling algorithms. In addition, the proposed HSA algorithm exhibits superior performance than the FNBF and NMRF algorithms because it can reduce transmission delay.

Fig. 26 shows the effect of a nonuniform transmission rate on the network throughput. In the simulation, 100 nodes were uniformly deployed within an area with 500 $\times$ 500 units. The bandwidth request of each node was randomly generated, and the value of the bandwidth request ranged from 0 to 2 Mb/s. The experiment was conducted for 2 h. To vary the transmission rate of each link, the signal was corrupted by adding Gaussian noise for 10 min at 30, 60, and 90 min. As shown in Fig. 26, all curves were generally stable at the beginning. However, the performance of all scheduling algorithms substantially reduced after 30 min. This occurred because considerable background noise reduces the transmission success ratio. Therefore, SSs change to lower coding rates, resulting in inferior throughputs. In general, the proposed HSA outperforms NMRF, NNBF, and FNBF, with regard to network throughput. This occurs because the proposed HSA algorithm arranges more possible transmissions simultaneously and alleviates the traffic congestion phenomenon.

VI. Conclusion

Several heuristic approaches have been proposed to manage the scheduling problem in WMNs. However, their performances highly depend on the network topology and the bandwidth request of each node, and they do not achieve optimal performance in all cases. This paper has proposed the SADP algorithm for minimizing the number of time slots required for a given set of bandwidth requests in WMNs. By considering sequential, nonparallel, and parallel relations, a valid and optimal schedule was constructed using a dynamic programming strategy, which avoids redundant computations. This paper also proposes a heuristic algorithm, called HSA, to reduce the computing complexity. The proposed HSA achieves similar performance to the SADP algorithm. The simulation results indicated that the proposed SADP and HSA algorithms outperform the existing algorithms in terms of average network throughput, peak traffic, and packet dropping ratio.

References

[1] *IEEE 802.16 Working Group, Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std. 802.16-2001, 2004.

[2] *IEEE 802.16 Working Group, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE Std. 802.16-2004, 2005.

[3] *IEEE 802.16 Working Group, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems—Multihop Relay Specification*, IEEE Std. 802.16-2006, 2007.

[4] Y. Li1, X. Zhang, H. Zhuang, and X. You, "An end-to-end QoS assurance method in IEEE 802.16 mesh networks," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–6.

[5] C. Cicconetti, I. F. Akyildiz, and L. Lenzini, "FEBA: A bandwidth allocation algorithm for service differentiation in IEEE 802.16 mesh networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 884–897, Jun. 2009.

[6] G. Vejarano and J. McNair, "Reservation-based distributed scheduling in wireless networks," in *Proc. IEEE Int. Symp. WoWMoM Netw.*, 2010, pp. 1–9.

[7] H. Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, "Interference-aware IEEE 802.16 WiMax mesh networks," in *Proc. IEEE VTC-Spring*, 2005, pp. 3102–3106.

[8] L. W. Chen, Y. C. Tseng, D. W. Wang, and J. J. Wu, "Exploiting spectral reuse in resource allocation, scheduling, and routing for IEEE 802.16 mesh networks," in *Proc. IEEE VTC Fall*, 2007, pp. 1608–1612.

[9] Y. Cao, Z. Liu, and Y. Yang, "A centralized scheduling algorithm based on multi-path routing in WiMax mesh network," in *Proc. Int. Conf. WiCOM Netw. Mobile Comput.*, 2006, pp. 1–4.

[10] H. Shetiya and V. Sharma, "Algorithms for routing and centralized scheduling in IEEE 802.16 mesh network," in *Proc. IEEE WCNC*, 2006, pp. 147–152.

[11] S. M. Cheng, P. Lin, D. W. Huang, and S. R. Yang, "A study on distributed/centralized scheduling for wireless mesh network," in *Proc. IEEE IWCMC*, 2006, pp. 599–604.

[12] B. Han, F. -P. Tso, L. Lin, and W. Jia, "Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks," in *Proc. IEEE Int. Conf. MASS*, 2006, pp. 789–794.

[13] B. Han, L. Lin, and W. Jia, "Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks," *Comput. Commun.*, vol. 30, no. 4, pp. 782–792, Feb. 2007.

[14] D. Ghosh, A. Gupta, and P. Mohapatra, "Admission control and interference-aware scheduling in multi-hop WiMAX networks," in *Proc. IEEE Int. Conf. MASS*, 2007, pp. 1–9.

[15] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 2, pp. 166–177, Apr. 1993.

[16] A. Esmailpour and N. Nasser, "A novel scheme for packet scheduling and bandwidth allocation in WiMAX networks," in *Proc. IEEE ICC*, 2011, pp. 1–6.

[17] J. M. Liang, J. J. Chen, Y. C. Wang, and Y. C. Tseng, "A cross-layer framework for overhead reduction, traffic scheduling, and burst allocation in IEEE 802.16 OFDMA networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1740–1755, May 2011.

[18] P. H. Wu and J. N. Hwang, "Cross-layer channel-quality-fair scheduling for video uplink of camera networks over WiMAX," in *Proc. IEEE ICC*, 2011, pp. 1–5.

[19] A. Esmailpour and N. Nasser, "Dynamic QoS-based bandwidth allocation framework for broadband wireless networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2690–2700, Jul. 2011.

Chih-Yung Chang (M'09) received the Ph.D. degree in computer science and information engineering from the National Central University, Taoyuan, Taiwan, in 1995.

He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan. His current research interests include Internet of Things, wireless sensor networks, ad hoc wireless networks, vehicular ad-hoc networks, and WiMAX broadband technologies.

Dr. Chang is a member of the Association for Computing Machinery. He served as an Associate Guest Editor for many Science Citation Index Journals, including *International Journal of Ad Hoc and Ubiquitous Computing* in 2011 and 2012, *International Journal of Distributed Sensor Networks* in 2012, *IET Communications* in 2011, *Telecommunication Systems* in 2010, *Journal of Information Science and Engineering* in 2008, and *Journal of Internet Technology* in 2004 and 2008.

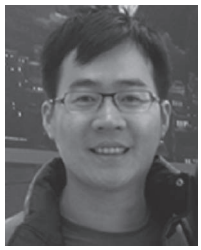
Ming-Hsien Li received the B.S. degree in computer science and information engineering in 2006 from Aletheia University, New Taipei City, Taiwan, and the M.S. degree in computer science and information engineering in 2008 from Tamkang University, New Taipei City, where he is currently working toward the Ph.D. degree in the Department of Computer Science and Information Engineering.

His current research interests include Internet of Things, wireless sensor networks, ad hoc wireless networks, and WiMAX broadband technologies.

Mr. Li was a recipient of several scholarship grants in Taiwan and has participated in many wireless sensor network projects.

Wen-Chuan Huang is working toward the Ph.D. degree in the Department of Computer Science and Information Engineering, Tamkang University, New Taipei City, Taiwan.

His current research interests include wireless sensor networks, ad hoc wireless networks, and WiMAX broadband technologies.



Shih-Chieh Lee received the Ph.D. degree in computer science and information engineering from Tamkang University, New Taipei City, Taiwan, in 2006.

He is currently with HTC Corporation, Taiwan. His research interests include wireless networks, bluetooth radio systems, sensor networks, mobile computing, WiMAX, and network protocol design.