A decentralized minislot scheduling protocol (DMSP) in TDMA-based wireless mesh networks

Chi-Tao Chiang a, Hung-Chang Chen b, Wen-Hwa Liao c, Kuei-Ping Shih a,*

a Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan
b Formosa Wireless Communication Corp., Taipei, Taiwan
c Department of Information Management, Tatung University, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 3 August 2012
Received in revised form 22 December 2012
Accepted 23 February 2013
Available online 7 March 2013

Keywords:
IEEE 802.16
Mesh networks
Minislot scheduling problem
Time division multiple access (TDMA)
Worldwide interoperability for microwave access (WiMAX)

A B S T R A C T

Wireless mesh network is a promising network topology that can provide high data rate backhaul network access. For achieving high data rate backhaul network access, a well-designed bandwidth scheduling protocol is necessary for wireless mesh networks. This paper takes minislot scheduling problem for IEEE 802.16 mesh networks as an example and formulates the problem as an integer linear programming model in this paper, where minislot is an atomic bandwidth allocation unit for data transmissions among subscriber stations and base station. Due to the high computational complexity for solving integer linear programming model at subscriber stations and the degradation of bandwidth utilization resulted from data collision problems and minislot insufficient problems, this paper proposes a decentralized minislot scheduling protocol to make subscriber stations, rather than base station, schedule minislot usage for throughput gains in the IEEE 802.16 mesh networks. The decentralized minislot scheduling protocol includes minislot usage constraints and minislot decision strategies to alleviate data collisions and minislot insufficient problems as well as to increase bandwidth utilization. The proposed protocol can not only accommodate to the IEEE 802.16 standard, but also makes subscriber stations schedule minislots with the latest minislot usage information. Besides IEEE 802.16 mesh networks, the proposed protocol also can apply to any wireless mesh networks with less or no modifications. From the simulation results, the performance of the proposed protocol outperforms the other related contributions in terms of the transmission delay, control overhead, minislot utilization and the network throughput.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

A wireless mesh network (WMN) is an emerging and stable topology for next generation wireless networking due to the characteristics such as multi-hop communications, self-forming, self-healing and self-organization, etc. (Akyildiz et al., 2005). WMNs have been supported in worldwide interoperability for microwave access (WiMAX), wireless fidelity (Wi-Fi), sensor networks and cellular networks. WiMAX, based on the IEEE 802.16 standard (Draft IEEE Standard for Local and Metropolitan Area, 2007), aims at providing high data rate backhaul network access by supporting the mesh mode and is considered in this paper.

Figure 1(a) shows an IEEE 802.16 mesh network, which is composed of one base station (BS) and multiple subscriber stations (SSs). Different from the point-to-multipoint (PMP) mode supported in the IEEE 802.16 standard, transmissions occur between SSs and may also be relayed via other SSs in the mesh mode. Consequently, the coverage of BS can be extended efficiently. To ensure that each SS can access the backhaul network through BS, BS maintains a routing tree, rooted by BS and composed of all SSs in the network, as shown in Fig. 1(b). Solid lines in Fig. 1(b) stand for the links of the routing tree. Dash lines in Fig. 1(b) stand for the links existing in the network tropology, but not existing in the routing tree.

In IEEE 802.16 mesh networks, bandwidth is divided into several frames and bandwidth access method is based on the time division multiple access (TDMA). An IEEE 802.16 mesh frame consists of a control subframe and a data subframe, as illustrated in Fig. 2. A control subframe contains transmission opportunities for the coordination of data transmissions among SSs and the BS. A data subframe contains minislots delivering SSs’ or BS’s data. Before data transmissions at the data subframe, SSs request the bandwidth by the mesh centralized scheduling (MSH-CSCH) request messages. Then, BS grants bandwidth in the unit of bps piggybacked in the MSH-CSCH grant message and broadcasts the
message to all SSs. While receiving the MSH-CSCH grant message, each SS shall apply a well-designed minislot scheduling algorithm to schedule the minislots in the data subframe for its transmissions (Draft IEEE Standard for Local and Metropolitan Area, 2007). Notice that BS is responsible for allocating the amount of bandwidth to SSs, rather than scheduling minislots. Although the IEEE 802.16 standard specifies the request and grant mechanism for bandwidth allocations, how to schedule minislots in the data subframe by SS in a decentralized manner is out of the scope of the standard and is defined as Minislot Scheduling Problem (MSP) in this paper.

Much related work (Viswanathan and Mukherjee, 2006; Han et al., 2007; Chen et al., 2006; Wei et al., 2005; Fu et al., 2005; Cao et al., 2006; Jin et al., 2007; Chen et al., 2009; Liu et al., 2009; Wei et al., 2005; Fu et al., 2005; Cao et al., 2006; Jin et al., 2007; Chen et al., 2006; Xiong et al., 2007; Lu and Zhang, 2007) has paid much attention on the centralized scheduling mechanism in IEEE 802.16 mesh networks. However, some related contributions do not consider the minislot reuse or bidirectional transmissions. As a result, the bandwidth cannot be fully utilized. Moreover, all aforementioned work also assumes that the minislot assignment is performed by BS in a centralized manner. Two main disadvantages to schedule minislot usage by BS are described as follows. One is that high computational complexity and high control overhead which are needed for BS to schedule the minislot usage before data transmissions. The other is that BS schedules minislots inappropriately for data transmissions based on the outdated minislot usage information; thereby it will lead to data collision problems and minislot insufficient problems. Minislot utilization is also decreased. Therefore, this paper investigates MSP and proposes a decentralized minislot scheduling protocol (DMSP) to make SSs schedule the minislot usage of the data subframe to achieve efficient data transmissions. The paper proposes minislot usage constraints and minislot decision strategies to cope with data collision problems and minislot insufficient problems to increase the minislot utilization. To our best knowledge, the paper is the first one to make SSs schedule minislots to accommodate to the IEEE 802.16 standard. In addition, DMSP can also apply to other TDMA-based wireless mesh networks with less or no modifications (Xu et al., 2012; Loscri, 2008; Sha et al., 2012; Yu and Chiu, 2010). From simulation results, DMSP achieves better performance than other related contributions in the transmission delay, control overhead, minislot utilization, and network throughput.

The rest of the paper is organized as follows. Section 2 briefly reviews and summarizes the contributions of related work. Section 3 describes the details of MSP. Section 4 formulates MSP as an integer linear programming (ILP) model, which is one of the contributions of this paper. Section 5 demonstrates the operations of DMSP, which is the main contribution of this paper. Section 6 analyzes the time complexity and message complexity of DMSP. Finally, the simulation results and conclusions are described in Sections 7 and 8, respectively.

2. Related work

For the distributed TDMA-based scheduling approaches, much related work (Mao et al., 2007; Ergen and Varaiya, 2010; Shi, 2010), especially in wireless sensor networks has been developed to make sensor nodes schedule the slots for prolonging the network life time and decreasing the delay time for data transmissions. In addition to the aforementioned related work, much research (Viswanathan and Mukherjee, 2006; Han et al., 2007; Chen et al., 2006; Wei et al., 2005; Fu et al., 2005; Cao et al., 2006; Jin et al., 2007; Chen et al., 2006; Xiong et al., 2007; Lu and Zhang, 2007) has focused on the MSP in IEEE 802.16 mesh networks. Table 1 summarizes the

<table>
<thead>
<tr>
<th>Related work</th>
<th>Spatial reuse</th>
<th>Uplink/Downlink traffic</th>
<th>Minislot schedule</th>
<th>Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wei et al. (2005)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>No</td>
<td>BS</td>
</tr>
<tr>
<td>Lu and Zhang (2007)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>No</td>
<td>BS</td>
</tr>
<tr>
<td>Cao et al. (2006)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>No</td>
<td>BS</td>
</tr>
<tr>
<td>Chen et al. (2006)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>Yes</td>
<td>BS</td>
</tr>
<tr>
<td>Han et al. (2007)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>Yes</td>
<td>BS</td>
</tr>
<tr>
<td>Chen et al. (2009)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>Yes</td>
<td>BS</td>
</tr>
<tr>
<td>Jin et al. (2007)</td>
<td>Yes</td>
<td>Uplink Only</td>
<td>Yes</td>
<td>BS</td>
</tr>
<tr>
<td>Fu et al. (2005)</td>
<td>Yes</td>
<td>Both</td>
<td>No</td>
<td>BS</td>
</tr>
<tr>
<td>Viswanathan and Mukherjee (2006)</td>
<td>Yes</td>
<td>Both</td>
<td>No</td>
<td>BS</td>
</tr>
<tr>
<td>Xiong et al. (2007)</td>
<td>Yes</td>
<td>Both</td>
<td>No</td>
<td>BS</td>
</tr>
<tr>
<td>Liu et al. (2009)</td>
<td>Yes</td>
<td>Both</td>
<td>Yes</td>
<td>BS</td>
</tr>
<tr>
<td>DMSP</td>
<td>Yes</td>
<td>Both</td>
<td>Yes</td>
<td>SSs</td>
</tr>
</tbody>
</table>

Table 1 Comparisons of existing protocols and DMSP.
considerations of the related work. In Lu and Zhang (2007), the authors proposed a routing tree construction scheme for centralized scheduling. The goal of the algorithm is to increase minislot reuse by adjusting the routing tree structure. Although the algorithm enhances the spatial reuse, additional control overhead is needed for the announcement of the adjusted routing tree structure. In Cao et al. (2006), an ILP model formulating the scheduling problem was proposed to achieve high system throughput and fairness. However, it is not practical for BS to schedule minislots by solving the ILP.

In Wei et al. (2005), a simple heuristic interference-aware centralized scheduling scheme was proposed to achieve high channel utilizations by avoiding interference among stations and increasing the minislots utilization. The authors considered a blocking metric of a route, defined as the number of nodes whose transmissions would be blocked by nodes on the route. BS applying the proposed algorithm schedules the SS with the lowest blocking metric to transmit its data first so that the number of concurrent transmissions can be increased. Chen et al. (2006) proposed an odd–even alternative layer algorithm to schedule minislots for the improvement of the concurrent transmissions. The routing tree is grouped into several layers. When a layer is activated for communications, the neighboring layers are not allowed for transmissions in order to avoid collisions. In Han et al. (2007), a transmission-tree scheduling (TTS) algorithm with the three selection criteria, including minimal interference, nearest to BS, and farthest to BS criteria, were proposed. However, TTS only took the scheduling of unidirectional transmissions into considerations, rather than the scheduling of bidirectional transmissions. Similar to the aforementioned work, Chen et al. (2009), Jin et al. (2007), Cao et al. (2006) also only considered the unidirectional transmissions. Without taking the bidirectional transmissions into account, the minislot utilization is unable to be efficiently improved.

In order to achieve high minislot reuse, a bidirectional concurrent transmission model and its corresponding ILP problem were proposed in Xiong et al. (2007). As mentioned above, solving ILP problem is time-exhausted and not practical in real environments. In addition, this model only maximized the concurrent transmissions, but did not take the whole traffic in the mesh network into considerations.

All the above related work requires the BS to schedule the minislots of each SS in a centralized manner. However, minislots scheduling needs high computational cost at BS, especially in a large scale of mesh networks. Therefore, from SSs’ viewpoint, this paper first formulates the MSP in terms of ILP model and then proposes a decentralized minislot scheduling protocol (DMSP) to make SSs schedule the minislots for data transmissions in an efficient and decentralized manner.

3. Problem statement of minislot scheduling problem (MSP)

According to the spec. (Draft IEEE Standard for Local and Metropolitan Area, 2007), each SS in an IEEE 802.16 mesh networks shall schedule minislots for its transmissions in a distributed manner. However, scheduling minislots distributively must consider data collision problem and minislot insufficient problems to improve minislot utilization. Before introducing MSP, some terminologies are defined first.

3.1. Notations

Let \( m \) be the number of minislots in a data subframe and \( s_k \) be the \( k \)-th minislot in the data subframe for downlink/uplink data transmissions, where \( 1 \leq k \leq m \). Suppose that one BS, denoted as \( v_0 \), and \( n \) SSs, termed \( v_i, i = 1, 2, 3, \ldots, n \), form an IEEE 802.16 mesh network. Initially, when a new SS, say \( v_i \), joins a mesh network, it synchronizes with the network and selects one sponsor node, say \( v_j \), for backhaul network access based on the receiving signal strength of \( v_j \) (Draft IEEE Standard for Local and Metropolitan Area, 2007). The detailed selection policies of a sponsor node are discussed in Shetiya and Sharma (2005), Shetiya and Sharma (2006), Lu and Zhang (2007) and are beyond the scope of this paper. After selecting sponsor node, \( v_j \) forwards the selection result to the BS. BS periodically broadcasts a mesh centralized scheduling configuration (MSCS-CF) message, including the tree structure information to all SSs. Let \( T \) denote the routing tree, which is rooted by \( v_0 \) and is composed of \( v_1, \ldots, v_n \) nodes. The detailed notations are listed and defined as follows:

- \( V = \{ v_i, i = 0, 1, 2, \ldots, n \} \): the set of nodes on the routing tree. \( v_0 \) represents BS and each \( v_i \in V - \{ v_0 \} \) represents the SSs.
- \( t_{i,j} \): a boolean coefficient, indicating whether \( v_i \) is the sponsor node of \( v_j \). \( t_{i,j} = 1 \), if \( v_i \) is the sponsor node of \( v_j \); otherwise, \( t_{i,j} = 0 \), \( \forall i \neq j \).
- \( N(i) = \{ v_j \in V | d(v_i, v_j) \leq R, v_i \neq v_j \} \): the set of \( v_i \)'s neighbors, where \( d(v_i, v_j) \) is the Euclidean distance between \( v_i \) and \( v_j \). \( N(0) \) is the set of BS’s neighbors, if \( i = 0 \).
- \( p_{i,j} \): a transmission pair from \( v_i \) to \( v_j \), where either \( t_{i,j} = 1 \) or \( t_{j,i} = 1 \), \( \forall v_i, v_j \in V, i \neq j \).
- \( f_{i,j} \): the data flow from \( v_i \) to \( v_j \), \( \forall v_i, v_j \in V, i \neq j \).
- \( M = \{ s_k, k = 1, 2, \ldots, m \} \): the set of minislots.
- \( M_i \): the set of minislots used by \( v_i \) for data transmissions, where \( M_i \subseteq M, \forall v_i \in V \).
- \( M_f \): the set of minislots used by \( v_i \) for data receptions, where \( M_f \subseteq M, \forall v_i \in V \).

3.2. Minislot scheduling problems (MSP)

MSP is to schedule stations’ transmissions with minimal number of minislots to improve the network throughput. Because minimizing the number of minislots needed for the satisfactions of stations’ requirements implies that more requirements can be accepted, the network throughput can be increased. More specifically, the formal definition of MSP is as follows.

Definition 1 (Minislot scheduling problem (MSP)). Given a set of data requirements of stations in the mesh network, MSP is to schedule minislots so that each data requirement can be met using minimal number of minislots.

In order to solve MSP efficiently, data collision problem and minislot insufficient problems should be taken into consideration. Therefore, data collision problem and minislot insufficient problems are described as follows.

3.2.1. Data collision problem

The data collision problem is a well-known problem in wireless communications and decreases the bandwidth utilization as well as the network throughput. Assume that one node, say \( v_i \), is receiving data at minislot \( s_k \). The data collision problem will occur when at least one neighbor of \( v_i \) issues a signal at \( s_k \) simultaneously. In this situation, \( v_i \) will fail to receive data at \( s_k \). The definition of the data collision problem is formulated as follows.

---

1 For simplicity, SSs and BS are assumed with identical transmission range.
Definition 2 (Data collision problem). For any \( v_i \in V \) and \( v_j \in N(i) \), the data collision problem occurs if and only if \( M_i \cap M_j \neq \emptyset \).

Obviously, the data collision problem incurs data collision, wastes bandwidth utilization, and degrades system throughput. As a result, when designing a minislot scheduling protocol, the data collision problem should be carefully considered.

3.2.2. Minislot insufficient problems

When an SS schedules minislots for transmissions but the number of available minislots is not enough to schedule, this problem is termed minislot insufficient problem. Minislot insufficient problems can be divided into two types based on inappropriate minislot selections. One is caused by the inappropriate minislot selections on the links of the self-flow and the other on the links of the neighboring flows, which are termed Intra-flow minislot insufficient problem and Inter-flow minislot insufficient problem, respectively. The detailed descriptions of the two types of minislot insufficient problems are given as follows.

Intra-flow minislot insufficient problem: Suppose that there is one flow \( f_i \) and \( v_i \) is one of the internal nodes of \( f_i \). If \( v_i \) suffers minislot insufficient problems caused by the inappropriate minislot selections for \( f_i \)'s data delivery on the previous links, this type of minislot insufficient problem is called intra-flow minislot insufficient problem.

Considering the routing tree shown in Fig. 1(b), Fig. 3 is an example to illustrate intra-flow minislot insufficient problem. Assume that \( m = 10 \) and \( M = \{s_{1}, s_{2}, \ldots, s_{10}\} \). A minislot in this figure marked as "R" represents that the minislot is scheduled for that SS or BS to receive data and a minislot marked as "S" represents that the minislot is scheduled for that SS or BS to transmit data. There are two flows, \( f_{0.5} \) and \( f_{0.1} \). One flow \( f_{0.5} \) goes through \( p_{0.1} - p_{2.1} \) and \( p_{2.1} - p_{3.1} \). The other one flow \( f_{0.1} \) goes through \( p_{0.1} - p_{1.1} \). The minislot usage of \( p_{0.1} - p_{2.1} \) and \( p_{2.1} - p_{3.1} \) are \( s_{2} \), \( s_{3} \) and \( s_{5} \), respectively. Suppose that another flow \( f_{12.0} \) goes through \( p_{12.0} - p_{10.0} \), \( p_{10.0} - p_{9.0} \), \( p_{9.0} - p_{8.0} \), and \( p_{8.0} - p_{7.0} \). All transmission pairs \( f_{12.0} \) goes through require two minislots. Considering the minislot usage as the left side of Fig. 3, only one minislot \( s_{8} \) is available for \( S_{8} \) and the number of available minislots is not enough to satisfy the requirements of \( p_{2.1} \). As a result, \( S_{8} \) must transmit its remaining data at next frame and the delay time will increase. However, if the minislot usage of \( f_{12.0} \) as right side of Fig. 3, the intra-flow minislot insufficient problem is avoided.

Inter-flow minislot insufficient problem: Given \( f_{i} \) and \( j \), \( v_k \) is one of the internal nodes of \( f_j \). If \( v_k \) suffers minislot insufficient problems due to the inappropriate minislot selections on the links for \( f_j \)'s data delivery, this type of minislot insufficient problem is termed inter-flow minislot insufficient problem.

Suppose that there is one flow \( f_{0.11} \) with the bandwidth requirement of two minislots and \( f_{0.11} \) goes through \( p_{0.3} - p_{1.7} \) and \( p_{1.7} - p_{11.7} \). The minislot usages of \( p_{0.3} - p_{1.7} \) and \( p_{1.7} - p_{11.7} \) are \( \{s_{1}, s_{5}\} \), \( \{s_{6}, s_{7}\} \) and \( \{s_{8}, s_{9}\} \), respectively, which are as shown in Fig. 4(a).

Considering the same flow \( f_{12.0} \) and the minislot usage of \( f_{12.0} \) without intra-flow minislot insufficient problem, which are illustrated in Fig. 4(b), the bandwidth requirement of \( p_{3} - p_{7} \) cannot be satisfied. It is because that \( p_{3} - p_{7} \) will be interfered by \( p_{3} - p_{7} \) on \( S_{8} \), which is not available for \( p_{3} - p_{7} \). Consequently, the minislot usage of \( f_{12.0} \) shown in Fig. 4(b) needs to be adjusted. If the minislot usage of \( f_{12.0} \) is illustrated as Fig. 4(c), both \( f_{12.0} \)'s data can be delivered with sufficient amount of minislots. Therefore, inter-flow minislot insufficient problem can be avoided.

4. Integer linear programming (ILP) model for MSP

Based on the above descriptions of MSP, this paper formulates MSP as an integer linear programming (ILP) model by taking the minislot reuse and interference relationships among stations into accounts. This ILP model is one of the contributions of this paper and the details of the ILP model are described as follows.

**Given:**

- \( r_{i}^{U} \): \( S_{i} \)'s uplink bandwidth requirement in the unit of bits per second (bps).
- \( r_{i}^{D} \): \( S_{i} \)'s downlink bandwidth requirement in the unit of bps.
- \( R^{U} = \{ v_i ^{U}, v_i ^{V} \in V -\{v_0\} \} \): the set of \( S_{i} \)'s uplink bandwidth requirements.
- \( R^{D} = \{ v_i ^{V}, v_i ^{V} \in V -\{v_0\} \} \): the set of \( S_{i} \)'s downlink bandwidth requirements.
- \( R^{P} \): the data in bytes transmitted from \( v_i \) to \( v_j \).
- \( \tau \): the length of one minislot in the unit of seconds.

**Variables:**

- \( \alpha_{i,j,t} \): a Boolean variable indicating whether \( v_i \) transmits data to \( v_j \) at \( s_k \) or not. If \( v_i \) sends data to \( v_j \) at \( s_k \), \( \alpha_{i,j,t} = 1 \); otherwise, \( \alpha_{i,j,t} = 0 \).
objective function and constraints of ILP are described as follows. Transmissions of stations at one minislot, minimizing the number of collisions in the transmission set is minimized, as shown in Eq.(1). Given a network of stations operated in a centralized manner, DMSP is proposed to achieve the best outcome in a given mathematical model for collision-free transmission.

**Objective:**

\[
\text{Minimize} \sum_{k=1}^{m} c_k \tag{1}
\]

Subject to:

\[
\begin{align*}
& (C1) \quad d_{i,j}^{-1} \times (t_{i,j} + t_{f,j}) + \sum_{v \in N(i) \cup N(j)} d_{v,i} \leq c_k, \forall k \\
& (C2) \quad \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right) - \sum_{v \in N(i) \cup N(j)} d_{v,i} \times t_{i,j} = \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right), \forall k, \forall l \\
& (C3) \quad \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right) = \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right), \forall k, \forall l \\
& (C4) \quad \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right) = \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right), \forall k, \forall l \\
& (C5) \quad \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right) = \sum_{s} \left( d_{i,j}^{-1} \times t_{i,s} + d_{j,s} \right), \forall k, \forall l
\end{align*}
\]

Note:

- **(C1)** is used to determine whether the transmission set is a collision-free transmission set. If \( A^i \) is a collision-free transmission set, \( c_k = 1 \) for all \( k \) such that \( A^i \) is free for transmission.
- **(C2)** is used to ensure that all downlink transmissions from SSSs or BS can be forwarded by other SSSs. That is, the total amount of data sent by \( v_i \) to its neighbors should be equal to the sum of the data from its sponsor node of \( v_i \) and itself.
- **(C3)** ensures that all uplink transmissions from SSSs can be forwarded by the other SSSs. Similarly, the total amount of data sent by \( v_i \) to its sponsor node should be equal to the sum of the data from its neighbors and itself.
- **(C4)** is used to make sure that all downlink data requirements of SSSs will be met. That is, each SSS’s downlink data \( r_{i,j}^d \) will be sent by BS over the total used minislots.
- **(C5)** is to ensure that all uplink data requirements of SSSs will be met. That is, BS can receive all uplink data from all SSSs over the total used minislots.

Although ILP is a mathematical method for determining a way to achieve the best outcome in a given mathematical model for some lists of requirements represented as linear equations, it is well-known that solving ILP is an NP-complete problem and is operated in a centralized manner. Therefore, DMSP is proposed as a heuristic scheme for SSSs to solve the MSP in a distributed manner. The ILP will be solved by CPLEX solver and the results will be compared in Section 7 to verify the efficiency of the proposed DMSP.

5. Decentralized minislot scheduling protocol (DMSP)

In order to alleviate the data collision problem and minislot insufficient problems with low computational complexity, a heuristic scheme, termed decentralized minislot scheduling protocol (DMSP), is proposed to increase minislot utilization and network throughput. In DMSP, each SS schedules minislots for data transmissions in a decentralized manner so that SSSs can adaptively arrange the minislots based on the latest minislot usage information and the share computation load of BS. Basically, three operations included in DMSP are listed as follows:

- **Determination of the validity of minislots based on minislot usage constraints.**
- **Collection of k-hop results of minislot usage constraints.**
- **Determination of minislot usage for data transmissions based on minislot decision strategies.**

Initially, each SS determines which minislots are available for data transmissions or receptions. For data collision problem alleviations, minislot usage constraints are proposed for SSSs to validate the minislots for data transmissions and receptions based on the minislot usage of its neighbors and itself. However, minislot insufficient problems cannot be efficiently avoided when the minislot usage information of one hop neighbors is considered. Therefore, SSSs need to collect the minislot usage information of k-hop neighbors for the minislot selection. With the information, SSSs apply minislot decision strategies to find appropriate minislots for data transmissions. Moreover, the minislots allowed concurrent transmissions without data collisions will be selected first and thereby the minislot utilization is further improved. The details of the above-listed operations are described as follows.

5.1. Determination of the validity of minislots based on minislot usage constraints

To avoid the data collision problem, minislot usage constraints are proposed to validate the usage of the available minislots for data transmissions and receptions. Initially, each node, \( v_i \), periodically collects the minislot usage information of \( v_j \), \( \forall v_j \in N(i) \). With \( M_i^d \) and \( M_i^u \), \( v_i \) can validate the minislot usage of the transmission pair \( p_{i,j} \).

Firstly, the available minislots for the sender and the receiver of \( p_{i,j} \) are validated. Let \( M_{i,j}^d \) be a set of minislots which are the filtered results based on the first one of minislot usage constraints. For each \( s_k \in M_{i,j}^d \), \( s_k \) should be free for \( v_i \) and \( v_j \). The formal definition of \( M_{i,j}^d \) is

\[
M_{i,j}^d = \left\{ s_k | s_h \notin M_i \cup M_j \cup M_{i,j}^d | s_k \notin M \right\} \tag{2}
\]

Take Fig. 5(a) as an example. SSS10 evaluates the minislot usages of SSS10 and SSS6 based on \( M_i^d \) and \( M_j^d \). Since \( M_i^d = M_j^d = \emptyset \), the \( (10,6) \) can be derived as \( S_{10} S_{10} S_{6} S_{6} S_{0} S_{0} S_{0} S_{0} S_{0} S_{0} \) based on Eq. (2).

In order to avoid the data collision problem caused by \( p_{i,j} \), the minislot usage of the \( v_i \)’s neighbors should also be taken into account. The consideration aims to ensure that the transmission of \( p_{i,j} \) will not interfere \( v_i \)’s reception, where \( v_j, v_i \in N(i) \) and \( i \neq j \). Let \( M_{i,j}^{(2)} \) be the minislot set of \( p_{i,j} \) containing the results of the second constraint. Given \( M_i, v_i \in N(i)-(v_j) \), for each \( s_k \in M_{i,j}^{(2)}, \)
\( v_i \) can transmit its data to \( v_j \) at minislot \( s_k \) and does not interfere \( v_i \)'s receptions. Eq. (3) can be formulated as follows:

\[
F_{ij}^{(2)} = \{s_k | s_k \notin M_i^j, s_k \in M, \forall v_i \in N(i) \setminus \{v_j\}\}
\]

(3)

Take the same example shown in Fig. 5(a) to demonstrate how the second one of minislot usage constraints is applied. \( SS_{10} \) observes the minislot usage of other neighbors except for \( SS_6 \), including \( M_2 = \{s_3, s_6\}, M_3 = \{s_4\} \) and \( M_4 = \emptyset \). Therefore, \( F_{10-6}^{(2)} = \{s_3 | s_3 \notin M_2 \cup M_3 \cup M_4 = \{s_3, s_5, s_6, s_7, s_8, s_9, s_{10}\}\}

After applying the above two minislot usage constraints, \( v_i \) knows the candidate minislots for its data transmissions without interference. Nevertheless, not all candidate minislots are suited for \( v_i \)'s receptions. Therefore, \( F_{ij}^{(1)} \) and \( F_{ij}^{(2)} \) will be forwarded to \( v_i \) through the MSH-CSCH request message. \( v_i \) then has to determine the suitable minislots for the data reception based on the received information.

While receiving the MSH-CSCH request message, \( v_i \) will validate \( s_k \) based on the third minislot usage constraint. Let \( F_{ij}^{(3)} \) be the minislot set of the results of the third minislot usage constraint. The main concept of the third minislot usage constraint is that, for each \( s_k \in F_{ij}^{(3)} \), \( s_k \) is unable to use \( v_i \) if \( \forall v_j \in N(i) \setminus \{v_i\} \). That is, \( s_k \notin M_i^j \). This constraint is to ensure that the data receptions of \( v_i \) are not interfered by \( v_j \). Eq. (3) can be formulated as

\[
F_{ij}^{(3)} = \{s_k | s_k \notin M_i^j, \forall v_j \in N(i) \setminus \{v_i\}\}
\]

(4)

In Fig. 5(a), \( v_i \) knows the minislot usage of its neighbors, including \( M_2 = \{s_3, s_6\}, M_3 = \{s_4\} \) and \( M_4 = \emptyset \). Therefore, \( F_{10-6}^{(3)} = \{s_3 | s_3 \notin M_2 \cup M_3 \cup M_4 = \{s_3, s_5, s_6, s_7, s_8, s_9, s_{10}\}\}

Based on the results from the above three minislot usage constraints, \( v_i \) is able to determine easily which minislots are available for \( p_{i,j} \) without collisions. Let \( F_{ij}^{(4)} \) be the filtered results based on \( F_{ij}^{(1)} \) and \( F_{ij}^{(2)} \). Therefore, \( F_{ij}^{(4)} \) is formally represented as

\[
F_{ij}^{(4)} = F_{ij}^{(1)} \cap F_{ij}^{(2)} \cap F_{ij}^{(3)}
\]

(5)

As a result, \( v_i \) has \( F_{10-6}^{(4)} \) based on the three minislot usage constraints, respectively. Because \( F_{10-6}^{(3)} = \{s_2, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\} \), \( F_{10-6}^{(2)} = \{s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\} \), \( F_{10-6}^{(1)} = \{s_5, s_7, s_8, s_9, s_{10}\} \), and \( F_{10-6}^{(4)} = \{s_5, s_7, s_8, s_9, s_{10}\} \) as indicated in Fig. 5(a).

5.2. Collection of \( \kappa \)-hops results of minislot usage constraints

Recall that if the node schedules the minislot only based on the minislot usage of one-hop neighbors, the amount of minislots may not be enough for the transmission pairs of a data flow. Consequently, a \( \kappa \)-hop backward decision strategy is applied for SSs to determine which SS on the flow should perform the minislot scheduling. The detail of the \( \kappa \)-hop backward decision strategy is described as follows.

Giving one flow \( f_{x,y} \) from \( v_x \) to \( v_y \), let \( v_i, v_j, v_k \) be the nodes on \( f_{x,y} \), where \( v_j \) is \( v_i \)'s one-hop neighbor as well as \( v_k \) is \( v_i \)'s \( \kappa \)-hop neighbor. The basic idea of the \( \kappa \)-hop backward decision strategy is that \( v_i \) should arrange the minislot usage of \( p_{i,j} \) after receiving the \( \kappa \)-hops results filtered by minislot usage constraints from \( v_i \) and \( v_k \). Take \( \kappa = 3 \) for the example shown in Fig. 5(b). Assume that nodes SS_10, SS_5, and SS_2 are on flow \( f_{12-0} \). Since SS_5 has collected 3-hops results filtered by minislot usage constraints, including \( F_{12-10}^{(3)} = \{s_4\} \) and \( F_{12-2}^{(3)} = \{s_4\} \). SS_2 needs to arrange the minislot usage for \( p_{12-10} \) based on \( \kappa \)-hop backward decision strategy.

Actually, the best way to find an optimal minislot scheduling for all nodes on the routing tree is to collect all minislot usages of all \( v_i \) in the service flow. However, it is time-consuming and has been proved as an NP-hard problem. In addition, the larger the value of \( \kappa \) is, the more the control overhead is. Therefore, it is a tradeoff between the control overhead and the efficiency of minislot scheduling. In the paper, the relationship between the throughput and the value of \( \kappa \) is investigated in the Section 7.

5.3. Determination of minislot usage for data transmissions based on minislot decision strategies

After the previous two operations, a node, say \( v_i \), which has collected filtered results of \( \kappa \)-hops schedules the minislot usage of \( p_{i,j} \) based on minislot decision strategies, where \( v_i \) is \( v_i \)'s one-hop neighbor and \( v_j \) is \( v_i \)'s \( \kappa -1 \)-hop neighbor. Minislot decision strategies are used by SSs to decide which minislots are appropriate for data transmissions to alleviate minislot insufficient problems as well as increase minislot reuse. Minislot decision strategies are composed of two heuristic strategies, least available minislot first strategy (LAMFS) and most reuse minislot first strategy (MRMFS). Both the strategies are proposed in this paper and are the contributions of this work. The LAMFS and MRMFS are described as follows.

5.3.1. Least available minislot first strategy (LAMFS)

In order to alleviate the intra-flow minislot insufficient problem, the available minislots of the following transmission pairs on the flow should be taken into considerations because the number of available minislots may be enough to satisfy the requirements of the following transmission pairs when certain minislots are only valid for the transmission pairs and these minislots are chosen by other transmission pairs. In this paper, the minislot available rate is introduced first and then LAMFS is proposed to alleviate the intra-flow minislot insufficient problem.
Let $D_{1-i}$ be a set of minislot available rates for $p_{1-i}$. For each minislot $s_k$ in $F_{1-i}$, the minislot available rate, termed $d_k$, is defined as the number of collected minislot sets $F_{1-i}, \ldots, F_{k-1-i}$, containing $s_k$, where $v_i$ is $v_i$'s $k$-th neighbor. Take Fig. 5(b) as an example and assume that $k = 3$. SS2 has collected $k$ filtered results based on minislot usage constraints, $F_{12-i}$ and $F_{10-i}$, which are shown as follows:

- $F_{12-i} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$
- $F_{10-i} = \{s_2, s_3, s_4, s_5, s_{10}\}$
- $F_{6-i} = \{s_5, s_6, s_7, s_8, s_9, s_{10}\}$

Observing the above-mentioned three minislot sets, the set of minislot available rates for $p_{12-i}$, termed $D_{12-i}$ is $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_{10}\} = \{1, 1, 1, 2, 3, 3, 3, 3\}$ because only $F_{12-i}$ contains $s_1, s_2$ and $s_4$; $s_8$ is one of the elements of $F_{12-i}$ and $F_{6-i}$; the other minislots in $F_{12-i}$ are belong to $F_{12-i}$ and $F_{6-i}$. Based on the above-mentioned rules, the set of minislot available rates for $p_{10-i}$ and $p_{6-i}$, termed $D_{10-i}$ and $D_{6-i}$ are shown as follows:

- $D_{10-i} = \{d_2, d_3, d_4, d_8, d_{10}\} = \{1, 3, 3, 3, 3\}$
- $D_{6-i} = \{d_3, d_4, d_5, d_6, d_{10}\} = \{3, 2, 3, 3, 3\}$

The principle of LAMFS is that the minislots with the least minislot available rate among these $k$ results filtered by minislot usage constraints of transmission pairs are preferable to be selected first, which is formulated as follows, where $\Omega_{1-i}$ is a minislot set scheduled for $p_{1-i}$ to deliver the data of each flow according to $D_{1-i}$.

$$\Omega_{1-i} = \left\{ s_k \mid s_k \in \arg \min_{s_i} d_i \right\} \tag{6}$$

As an example in Fig. 5(b), SS2 will select $s_1, s_3$, and $s_4$ for $p_{12-i}$ based on the above Eq. (6). Therefore, $\Omega_{12-i} = \{s_1, s_3, s_4\}$.

LAMFS is used to alleviate intra-flow minislot insufficient problems. If the minislots, which are also available in the next $k-1$ continuous lines, are selected, the available minislots in the next $k-1$ lines may be insufficient for data transmissions. Thus, minislot insufficient problems will occur in the next $k-1$ links. Therefore, the minislots with the least minislot available rate among these $k$ results filtered by minislot usage constraints of transmission pairs are preferable to be selected first.

### 5.3.2. Most reuse minislot first strategy (MRMFS)

MRMFS is used when the number of available minislots after filtered by LAMFS is still more than SSs’ bandwidth requirements. In order to alleviate inter-flow minislot insufficient problems, the minislot usage of neighboring flows filtered based on minislot usage constraints should be taken into considerations. In the meanwhile, minislot reuse can also be improved. Consequently, MRMFS takes neighboring filtered results of the minislot usage constraints into considerations and computes the minislot utilization rate of each minislot to make minislot selections. Because each node has minislot usage information of one-hop neighbors, it is not difficult to obtain the minislot utilization rates.

Let $U_{1-i}$ be a set of minislot utilization rates for $p_{1-i}$. For each minislot $s_k \in \Omega_{1-i}$, the minislot utilization rate, termed $u_k$, is defined as the number of $v_i$'s and $v_i$'s neighbors, which will use $s_k$ for data transmissions or receptions. Consider $p_{12-i}$ in Fig. 5(b). Because $s_1$ is used by $SS1$ and $SS2$, $u_1 = 2$. Since $s_3$ and $s_4$ are only used by $SS2$, $u_3 = 1$ and $u_4 = 1$. Consequently, $U_{12-i} = \{u_1, u_3, u_4\} = \{2, 1, 1\}$.

The principle of MRMFS is that the minislots with the most minislot utilization rate of transmission pairs are preferable to be selected first, which is formulated as follows, where $\Omega_{1-i}$ is a minislot set scheduled for $p_{1-i}$ to deliver the data of each flow.

$$\Omega_{1-i} = \left\{ s_k \mid s_k \in \arg \max_{s_i} u_i \right\} \tag{7}$$

Therefore, based on MRMFS, $S1$ is scheduled for $p_{12-i}$. If the values of the existing $u_k$ are the same, for example, $\{s_3, s_4\}$, the minislots with small ID will be scheduled first for less delay. Therefore, $s_3$ will be added into $\Omega_{12-i}$.

### 6. Analysis of the proposed protocol

This section analyzes the time and message complexities of DMSP. First, the time complexity and message complexity are defined. Followed by the definitions, the detailed derivations are provided in the following context of this section.

#### 6.1. Time complexity of DMSP

The time complexity of DMSP quantifies the amount of time taken by the operations of DMSP and the time complexity of operations are analyzed as follows.

In the first operation of DMSP, each node, $v_i$, needs to check minislot usage of neighbors and validates the minislots based on minislot usage constraints. For each node, $v_i \in N(i)$, $v_i$ needs to check and validate at most $m$ minislots and the minislot usage of all neighbors should be checked one time. Therefore, the time complexity of first operation is bounded by $mn$, where $n$ is the number of nodes in networks.

After the first operation, $v_i$, forward filtered results to the next-hop node, $v_j$. When receiving results, $v_j$ checks the number of collected information. If $v_j$ has collected $k$ filterd results, $v_j$ will make minislot selections for the previous $k$-th transmission pair based on $k$-hop backward decision strategy. The check procedure takes $O(1)$.

In the last operation, LAMFS and MRMFS are used for the minislot selections. First of all, $v_j$ needs to check the $k$ results and finds the minislots with the least minislot available rate among the $k$ results. Therefore, performing LAMFS takes $O(km)$. After performing LAMFS, MRMFS is used to filter the unnecessary minislots. Different from LAMFS, $v_j$ needs to check the minislot usage of neighbors and selects the minislots with the largest minislot utilization rate among filtered results of neighbors. Consequently, performing MRMFS takes $O(mn)$. Since these two strategies are performed independently, the time complexity of the final operation is $O(km + mn) = O(mn)$.

Based on the above mentioned, the time complexity of DMSP is $O(mn)$.

#### 6.2. Message complexity of DMSP

Message complexity is defined as the number of control messages needed to make minislot selections. The message complexity of DMSP is analyzed as follows.

For the first operation of DMSP, each node $v_i$ collects the minislot usage of its neighbors before using the first and the second minislot usage constraints to validate minislots. $v_i$ needs to broadcast its minislot usage to its neighbors once and each node $v_j \in N(i)$ returns the results to $v_i$. Therefore, $|N(i)| + 1$ control messages need to be exchanged before validating minislots based on the first and the second minislot usage constraints, where $|N(i)|$ is the cardinality of $N(i)$. After that, $v_i$ broadcasts the $F_{1-i}$ to
nodes. Suppose that nodes. Consequently, most control messages are generated by nodes. It is because these control messages cannot be aggregated by aggregating the MSH-CSCH request messages from its children.

7. Simulation results

This study conducts extensive simulations to evaluate the performance of DMSP. Table 2 shows the simulation parameters and these values are specified in the IEEE 802.16 standard (Draft IEEE Standard for Local and Metropolitan Area, 2007). In this simulation, BS and SSs are deployed in a random manner. The maximal height of the routing tree is 6. The frame duration is set to 5 ms. Each frame is composed of 256 minislots. All requests for SSs are generated randomly within the maximal traffic arrival rate and the minimal traffic arrival rate. The total simulation time is 1000 s and the simulation results are averaged over 30 runs. The detailed simulation results are illustrated and described as follows. Before demonstrating the simulation results, the performance metrics are introduced first.

7.1. Simulation I: impact of $\kappa$

In this simulation, the impact of $\kappa$ and the network density to the network throughput and control overhead of DMSP are evaluated as shown in Fig. 6(a) and (b), respectively. The network throughput is defined as the number of bytes successfully arriving at their destinations (BS or SSs) and the control overhead is defined as the ratio of the number of control packets to the number of data packets. Observed from these two figures, the simulation results show that the control overhead and the throughput increase with the increase of the value of $\kappa$, especially in the control overhead. However, the throughput increases slowly while the control overhead still increases linearly when $\kappa > 3$. Therefore, the simulation results indicate that the value of $\kappa$ should be set to three for better network throughput with less control overhead. That is to say that each SS only records at most 3-hops minislot usage information to have high network throughput.

7.2. Simulation II: comparisons with other protocols

In this simulation, DMSP is compared to the TTS with the selection criterion of the nearest BS first (Han et al., 2007), CLSA (Liu et al., 2009) and the optimal solution retrieved from ILP model in terms of the network throughput, data delay, minislot utilization, and the control overhead.

Figure 7(a) illustrates the network throughput of DMSP, TTS, ILP, and CLSA for different number of SSs in the networks. As can be seen, when the network density increases, the throughput

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Simulation parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Values</td>
</tr>
<tr>
<td>Number of BSs</td>
<td>1</td>
</tr>
<tr>
<td>Number of SSs</td>
<td>5–30</td>
</tr>
<tr>
<td>Topology</td>
<td>Random</td>
</tr>
<tr>
<td>Maximal height of routing tree</td>
<td>6</td>
</tr>
<tr>
<td>Frame duration</td>
<td>5 ms</td>
</tr>
<tr>
<td>Number of minislots per frame</td>
<td>256</td>
</tr>
<tr>
<td>Maximal traffic arrival rate</td>
<td>20 Mbps</td>
</tr>
<tr>
<td>Minimal traffic arrival rate</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>Traffic arrival pattern</td>
<td>Random</td>
</tr>
<tr>
<td>Simulation time</td>
<td>1000 s</td>
</tr>
<tr>
<td>Simulation runs</td>
<td>30</td>
</tr>
</tbody>
</table>

Fig. 6. Impacts of the network density and $\kappa$ on (a) the network throughput and (b) the control overhead.
of DMSP, TTS, ILP and CLSA ascends. It is worthwhile that the throughput of DMSP outperforms other related work from 0.1 Mbps to 0.3 Mbps. The reason is that DMSP considers the minislots insufficient problems but the other related work does not. Although the throughput of the ILP also performs better than that of DMSP, the difference is not significant. Compared to the related work, the throughput of DMSP is close to the optimal solution from ILP.

Figure 7(b) shows the impact of the number of SSs on the control overhead. It is observed that DMSP, TTS and CLSA have similar inclinations that the control overhead increases with the increase of the number of MSs. Although DMSP needs some extra information to be piggybacked in the MSH-CSCH messages, the ratio of the number of control packets to the number of data packets does not significantly increase. That is, DMSP can efficiently improve the network throughput, compared to the increase of the control overhead, by using minislot usage constraints and minislot decision strategies.

Figure 8(a) shows the impact of the network density on the average delay, defined as the average number of minislots for the delivery of the data from the sender to the receiver. Similarly, the curves of DMSP, TTS and CLSA increase when the network density increases. Obviously, the average delay of ILP is 3.08 minislots, which is the best one since ILP takes all conditions into considerations. In addition, the performance of DMSP is 3.31 minislots, which is the closet one to ILP since the bidirectional transmissions, intra-flow and inter-flow minislot insufficient problems are taken into considerations. On the other hand, the average delay time of TTS and CLSA increase rapidly. That is because the intra-flow and inter-flow minislot insufficient problems are not considered by TTS and CLSA, which not only degrade the network throughput, but also waste minislots. In addition, TTS only considers the unidirectional transmissions. Consequently, the performance of TTS is the worst one because the minislots cannot be fully reused.

Figure 8(b) shows the impact of the number of SSs on the minislot utilization. The minislot utilization is defined as the ratio of the used minislots to the available minislots of BS and SSs. Generally, DMSP can efficiently use the medium resource in the networks with different number of SSs. Many SSs can transmit their data concurrently. When the network density is low, CLSA and DMSP are also able to select the minislots with low collision probability and high minislot reuse easily. Without considering the bidirectional transmissions, TTS has low minislot utilization in all cases.

In summary, the simulation results indicate that DMSP performs better than the others in terms of the network throughput, data delay, the channel utilization and control overhead using minislot usage constraints and minislot decision strategies.

8. Conclusions

A wireless mesh network (WMN) is a promising wireless topology for providing the wireless backhaul network access. How to achieve the high data rate backhaul network access in WMNs has been gained significant attentions in the recent years. This paper takes the IEEE 802.16 mesh networks as an example and
emphasizes on the scheduling of the minislots in the data subframe for SSs’ transmissions. The scheduling problem is formally defined as the Minislot Scheduling Problem, called MSP. According to MSP, several constraints for the integer linear programming model are proposed. In addition, a Decentralized Minislot Scheduling Protocol (DMSP) is proposed in IEEE 802.16 wireless mesh networks to avoid the data collision and increase the minislot utilization. DMSP uses Minislot Usage Constraints to avoid the data collision problems and applies Minislot Decision Strategies to prevent minislot insufficient problems. Moreover, the paper is the first one to let SSs select the minislots in a distributed manner, which conforms to the IEEE 802.16 standard. Simulation results also show that DMSP performs better in terms of the network throughput, average delay time, minislot utilization, and the control overhead. Beside this, DMSP also can apply to other TDMA-based wireless mesh networks with less or no modifications.

Acknowledgment

The work was partially supported by the National Science Council of the Republic of China under grants NSC 101-2628-E-032-001-MY3 and NSC 100-2632-E-032-001-MY3.

References


