



On the performances of IEEE 802.16(d) mesh CDS-mode networks using Single-Switched-Beam Antennas

Chih-Che Lin, Shie-Yuan Wang*, Teng-Wei Hsu

Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan

ARTICLE INFO

Article history:

Received 25 March 2011

Received in revised form 13 October 2011

Accepted 17 December 2011

Available online 29 December 2011

Keywords:

Directional antenna

Switched-beam antenna

Mesh network

Coordinated distributed scheduling mode

IEEE 802.16

ABSTRACT

The IEEE 802.16(d) mesh coordinated distributed scheduling (CDS) mode is a novel technology for future fixed wireless backbone networks and designed for the use of omnidirectional antennas. The use of Single-Switched-Beam Antennas (SSBAs) may have great potential to increase network capacity due to the antenna directivity. However, a network designed for omnidirectional antennas usually cannot operate well or achieve good performance with the presence of antenna directivity.

In this paper, we review the designs of the IEEE 802.16 mesh CDS-mode network, study the issues of this network with the use of Single-Switched-Beam Antennas (SSBAs), and propose a complete solution to solve these issues. The performances of our proposed scheme is evaluated using simulations. The simulation results show that our proposed scheme can effectively solve the issues of using SSBAs in the IEEE 802.16 mesh CDS-mode network and greatly increase its network capacity.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The IEEE 802.16(d) mesh coordinated distributed scheduling (CDS) mode [1] is a wireless technology for next-generation fixed broadband relay networks. The 802.16(d) mesh CDS mode uses a TDMA-based Medium Access Control (MAC) layer and OFDM-based physical layer and mainly operates at a single frequency. The transmissions of control messages and data are separated. The control message transmissions are determined using a distributed pseudo-random algorithm, called the Mesh Election Algorithm (MEA), which employs a unique exponential holdoff design. MEA allows a node to know the possible next control message transmission intervals of the nodes in its two-hop neighborhood. Thus, by using MEA each node can schedule collision-free control message transmissions in its one-hop neighborhood. The data transmission of this network is scheduled in an on-demand manner. A pair of transmitting and receiving nodes should

negotiate a data transmission schedule using a three-way handshake procedure (THP). The handshake information of THP should be transmitted using control messages. Because the transmission timing of a control message is controlled by MEA, how MEA influences the network performance of the IEEE 802.16(d) mesh CDS mode is worth studying.

The performance of MEA has been extensively studied for years (e.g., [2–7]); however, most of the prior works were based on the use of omnidirectional radios. In the literature, rare work studied the performances and challenges of the 802.16(d) mesh CDS-mode network when it uses directional antennas. Since the MEA specified in the standard is assumed to operate over omnidirectional antennas, several operational problems will need to be solved, when it operates over directional antennas.

Recent directional antenna technologies can be categorized into three classes: (1) the switched-beam antenna (SBA); (2) the adaptive array antenna; and (3) the Multiple Input Multiple Output (MIMO) array antennas. SBA can point to several pre-determined directions with pre-defined antenna gain patterns, which can work well

* Corresponding author.

E-mail address: shieyuan@csie.nctu.edu.tw (S.-Y. Wang).

without the presence of the multi-path effect (such as in the free-space environment). Due to lacking the nulling capability, they may not achieve the optimal concurrent transmission schedules in multi-path-prone environments.

The adaptive array antenna can form arbitrary beams and point to arbitrary directions. By forming “nulls” towards directions of unwanted transmitters and receivers, the adaptive array antenna can effectively reduce the signal interference. The MIMO antenna employs multiple antenna elements on both the transmitter and receiver ends, which thus can further increase the capacity of a network. Adaptive array antennas and MIMO antennas can outperform SBAs in achieved network capacity and signal quality; however, SBAs are superior in cost and design complexity [8]. For this reason, in this paper we adopt Single-Switched-Beam Antennas (SSBAs) to construct a cost-effective directional-antenna network using the IEEE 802.16 mesh CDS mode.

In the remainder of this paper, we reviewed the design of this new network, identified the problems of using SSBAs in this network, and proposed a scheme to solve these problems. The proposed scheme can operate using only directional transmissions/receptions. (Most of the previous proposals for directional antenna networks have to use omnidirectional transmissions/receptions in some phases of network operation, which increases the design complexity of the power control.) We conducted proof-of-concept simulations to evaluate the network capacity increased by using SSBAs in this network. In addition, we also evaluated the performances of TCP (Transport-layer Control Protocol) using a real-life TCP implementation in such networks. TCP is a well-known transport-layer protocol widely used in current network applications (such as FTP, HTTP) and sensitive to network congestion and end-to-end packet delay jitters. Due to the unique protocol design of the 802.16 mesh CDS mode, the performance of TCP over this network with SSBAs is interesting and worth studying.

To the best of the authors' knowledge, this paper is the first work that discusses how to enable the IEEE 802.16(d) mesh CDS-mode network to operate with SSBAs and evaluates the performances of this network with SSBAs. Although there have been many prior works studying TDMA networks with directional radios [9–19], they differ from an IEEE 802.16(d) mesh CDS-mode network using SSBAs in either control message scheduling or data scheduling. Thus, the issues and performances of the IEEE 802.16(d) mesh CDS-mode network employing SSBAs is worth studying.

The remainder of this paper is organized as follows. In Section 2, we survey prior works related to this paper and explain the differences between these works and ours. In Section 3, we briefly review the operation of the IEEE 802.16(d) mesh CDS-mode network. In Section 4, we explain the problems of the IEEE 802.16(d) mesh CDS mode when it operates with SSBAs and present the design of our proposed scheme. In Section 5, we evaluate the performances of the proposed scheme using the NCTUns network simulator [20]. Finally, in Section 7 we conclude the paper. In addition, the issues of network initialization with SSBAs for this new network are discussed and solved in appendix.

2. Related work

In addition to the mesh CDS mode, the IEEE 802.16(d) specification also defines the Point-to-MultiPoint (PMP) mode and the mesh Centralized Scheduling (CS) mode. The issues and performance on using directional antennas in the latter two modes have been well studied. These two modes employ centralized mechanisms to schedule control messages and data. Due to this fundamental difference, the previous studies on these two modes differ from this work.

Regarding the unique holdoff time design of the 802.16(d) mesh CDS mode, several dynamic holdoff time schemes have been proposed to enhance the network performance [2–4,21–25]. The main ideas of these proposals are to dynamically adjust each node's holdoff time value according to several criteria (e.g., its output buffer occupancy, its intention to send data, etc.) to increase network performances without generating network congestion. These proposals assume that the network uses omnidirectional antennas and thus each node in the network only needs to run one instance of MEA to schedule control messages. However, our work considers the network uses SSBAs to additional network capacity. In this network, due to limited coverage of SSBAs, single MEA cannot schedule a control message broadcast to all neighboring nodes at the same time. With this constraint, we propose a scheme where each node runs multiple MEA instances with a dynamic holdoff time design. For these differences, our proposed dynamic holdoff time scheme greatly differs from those proposed in the previous works.

The deployment cost of an IEEE 802.16-based network using directional antennas was discussed in [26]. In [27], Xiong et al. relaxes the data scheduling rules to increase the concurrency of data transmissions with nodes using omnidirectional antennas. We integrated this notion into our developed data scheduler for 802.16(d) mesh CDS-mode omnidirectional antenna networks. However, for 802.16(d) mesh CDS-mode networks using SSBAs, we developed a transmission-domain-aware minislot scheduler that can utilize the advantage of SSBAs to increase the concurrency of data transmissions.

In the literature, collision-free distributed algorithms for time-division networks have been proposed. Most of them, however, [12–16] require nodes to use omnidirectional transmissions or receptions in some phases. In contrast, our work need not use omnidirectional transmissions and receptions to operate. Note that, the antenna gain of an SSBA in the directional mode and that in the omnidirectional mode may greatly vary. Without proper transmission power control, the connectivity among nodes in the directional mode and that in the omnidirectional mode can be inconsistent and hinder network operation. Since the objective of our work is to propose a low-cost low-complexity solution for the IEEE 802.16 mesh CDS-mode network, in our proposal the omnidirectional broadcast of SSBA need not be used, which avoids the use of complicated power control and thus reduces the complexity of MAC-layer design and the efforts of network deployment.

MAC protocols using pure directional transmissions/receptions have been proposed in [17–19]. In these

proposals, control messages can be transmitted at arbitrary time; however, the IEEE 802.16(d) mesh CDS-mode network uses a unique holdoff time design to control its control message transmission timing in a special manner. This fundamental distinction differentiates these works and ours. In addition to the difference of the operations of the studied networks, in this paper we also studied the performances of a real-life TCP implementation (BIC-TCP) over the IEEE 802.16(d) mesh CDS-mode network with SSBA. To the best of the authors' knowledge, this is the first work that studies the performance of a real-life TCP implementation over this new network with SSBA.

3. Introduction to the IEEE 802.16(d) mesh CDS mode

To make this paper self-contained, we briefly present the operation of an IEEE 802.16(d) mesh CDS-mode network here. In this network, link bandwidth is divided into frames on the time axis and managed in a time-division-multiple-access (TDMA) manner. Each frame comprises one control and one data subframes. A control subframe is further divided into transmission opportunities (TxOpps), whereas a data subframe is further divided into minislots. Control messages and data packets are transmitted over TxOpps and minislots, respectively.

The 802.16 mesh CDS mode defines three types of control messages: (1) Mesh Network Entry (MSH-NENT); (2) Mesh Network Configuration (MSH-NCFG); and (3) Mesh Distributed Coordinated Scheduling (MSH-DSCH). The MSH-NENT and MSH-NCFG messages are used for nodes to exchange control information for network initialization whereas the MSH-DSCH message is used for nodes to schedule minislot allocations. A minislot allocation is a set of consecutive minislots across several consecutive data subframes for data packet transfers. The standard categorizes TxOpps into three types, each for transmitting a specific type of control messages. The TxOpps used for transmitting MSH-NENT, MSH-NCFG, and MSH-DSCH messages are called MSH-NENT TxOpps, MSH-NCFG TxOpps, and MSH-DSCH TxOpps, respectively [1].

3.1. Control message scheduling mechanism used in the 802.16(d) mesh CDS mode

In the mesh CDS mode, each node's control message transmission is scheduled by MEA [1], which is hash-based algorithm with an exponential holdoff mechanism. In this mode, each node needs to maintain (i.e., learn) the information about its two-hop neighborhood (which is an input of MEA). The two-hop neighborhood of a node i is defined as follows:

$$nbr(i) = \{i\} \cup nbr_1(i) \cup nbr_2(i), \quad (1)$$

where $nbr_1(i)$ and $nbr_2(i)$ are defined as follows, respectively

$$nbr_1(i) = \{k \mid \text{node } k \in \text{node } i\text{'s one-hop neighboring nodes, when omnidirectional antennas are used}\} \quad (2)$$

$$nbr_2(i) = \{k \mid \text{node } k \in \text{node } i\text{'s two-hop neighboring nodes, when omnidirectional antennas are used}\} \quad (3)$$

The purpose of maintaining the two-hop neighborhood is to avoid the hidden terminal problem when transmitting control messages. By using the same MEA in every node, this algorithm guarantees that in the network only one node in any two-hop neighborhood will win the chance to use a specific TxOpp. Thus, when nodes transmit their control messages, no message collisions will occur. Since the transmission of control messages can be guaranteed collision-free and these messages are used for scheduling a collision-free minislot allocation, the transmission of data packets can also be guaranteed collision-free. To maintain the two-hop neighborhood information, every node in the network must periodically broadcast its MSH-NCFG and MSH-DSCH messages in a collision-free manner. Because the scheduling processes of these two messages are the same, in the following we only explain the latter for brevity.

Each node should perform the MEA to determine the TxOpp on which to broadcast its next MSH-DSCH message. The MEA takes a given TxOpp number and an eligible node list (a list of nodes eligible to contend for the given TxOpp) as input. It then iteratively computes a hash value for each node in the given eligible node list on the given TxOpp. Finally, it outputs the ID of the winning node whose computed value is the largest among all the nodes in the eligible node list. The detailed hash operations used in MEA are defined in [1]. Since nodes within two hops use the same MEA and consistent eligible node lists, every node knows whether it will win a given TxOpp in its two-hop neighborhood.

To achieve the consistency of neighboring nodes' eligible node lists for each TxOpp, each node should periodically broadcast the next MSH-DSCH TxOpps used by itself and its one-hop neighboring nodes. If a neighboring node j 's next MSH-DSCH TxOpp is unknown, node i will conservatively consider that node j potentially contends for every TxOpp and put node j into the eligible node list for all following TxOpps until receiving the information of node j 's next MSH-DSCH TxOpp. Using this design, no collision will occur on any TxOpp. If a node cannot win a given TxOpp, it repeats the above process with the next TxOpp as input until eventually winning one TxOpp.

The eligibility of a node for contending for a TxOpp is determined by the holdoff time mechanism [1]. The holdoff time mechanism first defines the control message transmission cycle of a node as the time interval between the node's two consecutive control message transmissions. As shown in Fig. 1, the transmission cycle of a node comprises (1) the holdoff time and (2) the contention time. The former is defined as the number of consecutive TxOpps during which a node must suspend its contention for TxOpps after winning a TxOpp. The latter is defined as the number of consecutive TxOpps for which a node may contend to win a TxOpp. Conceptually, by obtaining the holdoff times of the nodes in its two-hop neighborhood, a node can know for which TxOpps these neighboring nodes will and will not contend. Based on such information, it can construct an eligible node list of its two-hop neighborhood

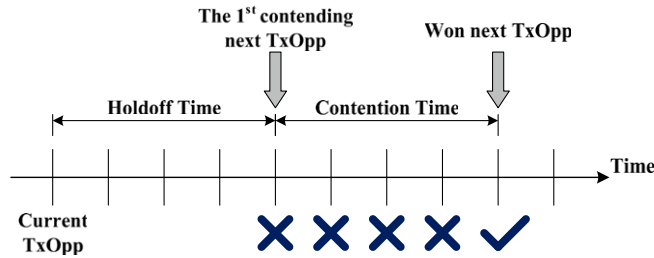


Fig. 1. Transmission cycle of a node.

for each TxOpp. In the standard [1], the holdoff time of each node is fixed and defined as follows:

$$\text{Holdoff Time} = 2^{\text{exponent} + \text{base}}, \quad (4)$$

where the base value is fixed to 4 and the range of the exponent value is between 0 and 7. In contrast, the contention time may vary depending on which and how many nodes are eligible to contend for TxOpps.

Similar to the next MSH-DSCH TxOpp, each node should put the holdoff times of its own and its one-hop neighboring nodes in MSH-DSCH messages. By this design, every node knows such information about every other node in its two-hop neighborhood, for any given TxOpp; thus, the output of every node's MEA in any two-hop neighborhood is consistent. The consistency of the MEA output means that, each node only knows if it wins a TxOpp or not. If it wins the TxOpp, no other nodes in its two-hop neighborhood will win the same TxOpp. If it did not win the TxOpp, it would think that some node *a* wins it while other node may think that node *b* wins it. (This is because, in reality, each node's two-hop neighborhoods may not be the same but partially overlap.) However, except the winning node, all the other nodes in the same two-hop neighborhood will not win the same TxOpp.

Another advantage of the holdoff time design is that it prevents nodes from contending for TxOpps after they win one until the holdoff time has elapsed. This design ensures that nodes other than the winning node will have a chance to win subsequent TxOpps and all nodes can fairly share TxOpps in the long run.

3.2. Data scheduling mechanism used in the 802.16(d) mesh CDS mode

The 802.16(d) mesh CDS mode schedules data transmissions of nodes in a distributed and on-demand manner. A three-way handshake procedure (THP) is used for a pair of nodes to negotiate a minislot allocation agreed by both nodes. A minislot allocation is a set of consecutive minislots on which the transmitting node can transmit data packets and the receiving node can receive data packets. As shown in Fig. 2, the THP uses a “request-grant-confirm” control sequence for two peer nodes to negotiate a minislot allocation. In a THP, the requesting node first transmits a request IE (Information Element) and an availability IE to the granting node using an MSH-DSCH message. The request IE specifies the number of minislots that the requesting node needs to transmit data packets and the

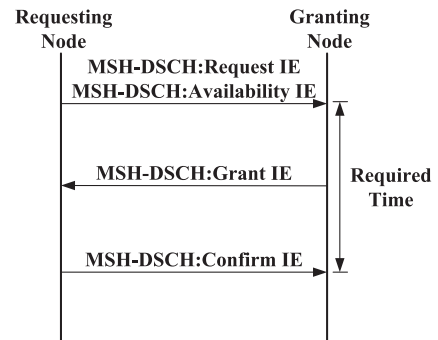


Fig. 2. The THP defined in the IEEE 802.16(d) mesh CDS mode.

availability IE specifies a set of consecutive minislots on which the requesting node are available for data transmission.

On receiving these two IEs, the granting node first determines whether it can receive data packets from the requesting node within the minislot set specified by the received availability IE. If not, the granting node can simply ignore the received request IE. Otherwise, it allocates a minislot allocation within the specified minislot set and then transmits a grant IE to the requesting node as an acknowledgment using its MSH-DSCH message. The grant IE specifies a subset of the minislots specified by the received availability IE on which the granting node is willing to receive data packets from the requesting node. Upon receiving the grant IE, the requesting node then broadcasts a confirm IE using its MSH-DSCH message to complete this THP. The confirm IE is simply a copy of the received grant IE and used to notify the requesting node's one-hop neighboring nodes of the duration on which this minislot allocation will take place.

4. Using SSBA in the 802.16(d) mesh CDS mode

The definition of the two-hop neighborhood in [1] is based on the use of omnidirectional antennas and thus has an important property: *if node A is in node B's two-hop neighborhood, then node B is also in node A's two-hop neighborhood*. It is this property ensuring that the MEA used in each node generates collision-free TxOpp scheduling because nodes A and B cannot both win the same TxOpp in their respective two-hop neighborhoods. However, when radio coverage is purely directional, the above

property no longer holds. This makes receiving control messages from other nodes non-trivial and network operation encounters several issues that need to be solved. These issues and our proposed solutions are explained below.

4.1. Problem 1: imprecise representation for TxOpps in control messages

In the standard, to reduce bandwidth consumption, the next TxOpp number of a node carried in MSH-DSCH and MSH-NCFG messages is represented by a 5-bit Mx field and a 3-bit Exp field [1], rather than a single long field. Using this representation scheme, a TxOpp number is represented using the following formula:

$$2^{exp} * Mx < \text{TxOpp number} \leq 2^{exp} * (Mx + 1), \quad (5)$$

where $0 \leq Mx \leq 30$, $0 \leq exp \leq 7$. The interval between $(2^{exp} * Mx, 2^{exp} * (Mx + 1)]$ is called the next transmission interval of a control message in the standard.

It is known that using MEA no two nodes in the same two-hop neighborhood will use the same TxOpp to transmit messages. However, the transmission intervals of two nodes in the same two-hop neighborhood may overlap with each other. For example, consider two nodes A and B are in node C's two-hop neighborhood. Suppose that the current TxOpp is 0 and nodes A and B choose TxOpps 33 and 36 as their next MSH-DSCH TxOpps, respectively. In this condition, both nodes A and B may use $(2^5 * 1, 2^5 * 2]$ (base = 4, exp = 1, $Mx = 1$) as their next transmission intervals and notify node C of these settings.

This overlapping problem does not hinder the operation of this network, when omnidirectional radios are used, because each node can listen incoming messages omnidirectionally, when it need not transmit control messages. In this condition, a node will not miss any control messages broadcast from neighboring nodes as long as it is not in the transmitting state. However, using pure directional reception, a node cannot determine to which direction its antenna should point because this imprecise representation cannot provide sufficient information for a node to know which node will transmit a message on a specific TxOpp among nodes with overlapped next transmission intervals.

Another problem is that this imprecise representation scheme reduces the flexibility of control message scheduling. Consider two neighboring nodes A and B. Using this TxOpp representation, on receiving an MSH-DSCH message from node B, node A cannot know the exact next TxOpp number won by node B. Instead, it can only derive an interval of 2^{exp} TxOpps in length during which node B will broadcast its next MSH-DSCH message. Since node A cannot know the exact next TxOpp that node B wins, to successfully receive the message transmitted from node B, it has to point its antenna towards node B during the whole interval. In contrast, if the exact TxOpp that node B will use can be known, node A can exchange control messages with other neighboring nodes in this long interval, reducing the latency of control message exchange.

From this observation, to enhance the performance of the 802.16(d) mesh CDS-mode network using SSBA, a control message scheduling scheme has to control nodes' antenna directions in a per-TxOpp manner. To this end, our proposed scheme introduces a new *offset* field into the MSH-NCFG and MSH-DSCH message formats. With the help of the *offset* field, a node can use the following expression to precisely derive the TxOpp numbers won by each of its neighboring nodes and thus know to which direction it should point the antenna on each TxOpp

$$\text{TxOpp number} = 2^{exp} * Mx + \text{offset}. \quad (6)$$

4.2. Problem 2: control message scheduling using SSBA

The original MEA defined in [1] schedules when to broadcast control messages for nodes using omnidirectional radios, which cannot be directly applied to networks using SSBA. The reason is that a node using the original MEA cannot know to which direction it should point its antenna for data transmission and reception on a given TxOpp. To solve this problem, we propose a distributed control message scheduling scheme called the Multiple Transmission Domain (MTD) Scheme to coordinate nodes' antenna pointings on each TxOpp. The MTD scheme uses a Transmission-domain-aware Mesh Election Algorithm (TMEA) and a Transmission-domain-aware Minislot Scheduling Algorithm (TMSA). The former is designed for a node to properly control when and in which direction it should transmit a control message using a SSBA, while the latter is designed for a node to exploit the spatial-reuse advantage of SSBA to increase data transmission concurrency. The notion of a transmission domain (TD) is explained below.

4.2.1. Transmission domain and its two-hop neighborhood

In this section, we define a transmission domain (TD) and its two-hop neighborhood for the use of SSBA. A TD of a node is defined as the set of nodes that are located in the coverage of a single switched beam and can simultaneously receive a message directionally transmitted by the node in that coverage. According to this definition, a node using an omnidirectional antenna has a single TD that includes all of its one-hop neighboring nodes in its 360-degree radio coverage. In contrast, a node using a SSBA has several TDs, each of which includes only the nodes in a specific beam coverage.

A node using a SSBA has $\frac{2\pi}{B}$ disjoint TDs to form a 360-degree radio coverage, where B denotes the antenna beamwidth in radians. Each TD is assigned a unique identifier i called the "Transmission Domain Index (TDI)." A TD i is composed of the nodes in the sector area between $B \cdot (i - \frac{1}{2}) \bmod 2\pi$ and $B \cdot (i + \frac{1}{2}) \bmod 2\pi$ in polar coordinates, $\forall i \in N$, $0 \leq i \leq \frac{2\pi}{B}$. In this paper, B is set to $\frac{\pi}{2}$ radians and it can be changed to another value to better suit a given network topology. Therefore, each node has four disjoint TDs each comprising nodes in the areas of $\frac{\pi}{2} \cdot (i \pm \frac{1}{2}) \bmod 2\pi$ in polar coordinates, where $i \in N$, $0 \leq i \leq 3$.

A node i using an SSBA maintains a two-hop neighborhood for each of its TDs. $nbr(i_j)$ denotes the set of nodes in node i 's two-hop neighborhood associated with TD j and its definition is given as follows:

$$\begin{aligned} nbr(i_j) = & \{i\} \cup \{k | k \text{ is in the radio coverage of TD } j\} \\ & \cup \{nbr_1(k) | k \text{ is in the radio coverage of TD } j\}, \end{aligned} \quad (7)$$

where $nbr_1(k)$ is defined by Eq. (2). The reason why we use $nbr_1(k)$ in the last term is that, a node k in TD j may point its SSBA to any of its one-hop neighboring nodes to transmit/receive messages when node i intends to transmit a message to it. Thus, from the perspective of TxOpp scheduling, the message schedulings of nodes in $nbr_1(k)$ may conflict with those of node i to node k . For this reason, $nbr_1(\cdot)$ should be used to take such conflicts into consideration.

The details of the MTD scheme are explained below in Section 4.2.2. For clarity, in the remainder of this paper, the 802.16(d) mesh CDS mode that operates over an omnidirectional-antenna network is referred to as the single-transmission-domain (STD) scheme.

4.2.2. Transmission-domain-aware Mesh Election Algorithm (TMEA)

One intuitive idea for the 802.16(d) mesh-CDS network with SSBA is to run multiple MEAs in each TD of a node. However, this intuitive approach cannot work and achieve good performance without a proper revision. In the following, we first present a basic version of TMEA that uses a Static holdoff time design (called TMEA-S) and discuss its drawbacks. We then present an advanced version of TMEA that uses a dynamic holdoff time design (TMEA-D) to boost the performance of this network.

Consider an 802.16(d) mesh CDS-mode network using SSBA, where each node has four TDs. Following the design in [1], each node should execute four MEA Instances (denoted as MEAIs), each maintaining a two-hop neighborhood and scheduling next message transmission for a specific TD. For brevity, the MEAI for a TD k is denoted as TD- k MEAI. The standard [1] does not define how multiple MEAIs of the same node operate and coordinate with each other; thus, these MEAIs calculate their next TxOpps independently. Without coordination, the MEAIs on a node may choose the same TxOpp because they do not know whether the TxOpp that they choose has already been chosen by other MEAIs. In this case, some of neighboring nodes will inevitably miss important control messages broadcast from this node. Such a scheduling conflict is called an “intra-node scheduling conflict” in this paper as it takes place among the TDs on the same node.

A basic solution that we proposed to address the intra-node scheduling conflict is TMEA-S, which only uses the static holdoff time design defined by the standard. The basic idea of TMEA-S is to let all MEAIs on the same node share their next TxOpp information with each other. Following this design, if an MEAI finds a scheduling conflict, then it will yield the chosen TxOpp and start next iteration to find new TxOpp that it can win.

Fig. 3 illustrates an example operation of TMEA-S. In this example, a TD-1 MEAI is scheduling its next control message transmission. At the first two iterations, the TD-1 MEAI chooses TxOpps that have been chosen by the TD-3 MEAI and TD-2 MEAI, respectively. Thus, it yields these TxOpps and start next iteration and finds a TxOpp that will not result in intra-node scheduling conflicts.

Note that in TMEA-S, a TD- j MEAI should use an eligible node list derived from the holdoff times of nodes in $\cup nbr(i_m), \forall TD m \in \text{node } i$, rather than an eligible node list derived from the holdoff times of nodes in $nbr(i_j)$ only. Consider the example network shown in Fig. 4, where nodes A and B are located in the other's TD 2 and TD 0, respectively. Assume that the TD-0, TD-1, and TD-3 MEAIs of node A have chosen TxOpps 5, 2, and 3, respectively, and the TD-2 MEAI of node B has chosen TxOpp 8.

Later on, when the TD-2 MEAI of node A schedules its next TxOpp, if the TD-2 MEAI of node B is not included in its two-hop neighborhood (which is included only in $nbr(A_0)$), it may also choose TxOpp 8 as its next TxOpp, resulting in a scheduling conflict on TxOpp 8. When node B sends a control message to node A on TxOpp 8, node A will not be able to receive the message because node A will point its antenna to its TD 2 on TxOpp 8. Because such a scheduling conflict is caused by the MEAIs on different nodes, it is called an “inter-node scheduling conflict” in this paper. To avoid them, each node using TMEA-S has to notify its one-hop neighboring nodes of the next TxOpp numbers of all its own MEAIs and those of all its one-hop neighboring nodes' MEAIs (learned from received MSH-DSCH messages transmitted by its one-hop neighboring nodes) using MSH-DSCH messages.

With the complete next TxOpp information of all neighboring MEAIs, TMEA-S can build an eligible node list considering the next TxOpps of all nodes' MEAIs in $\cup nbr(i_m), \forall TD m \in \text{node } i$ to avoid inter-node scheduling conflicts. For instance, for the example shown in Fig. 4, using this design the TD-2 MEAI of node A will include node B in its eligible node list for TxOpp 8 according to the next TxOpp

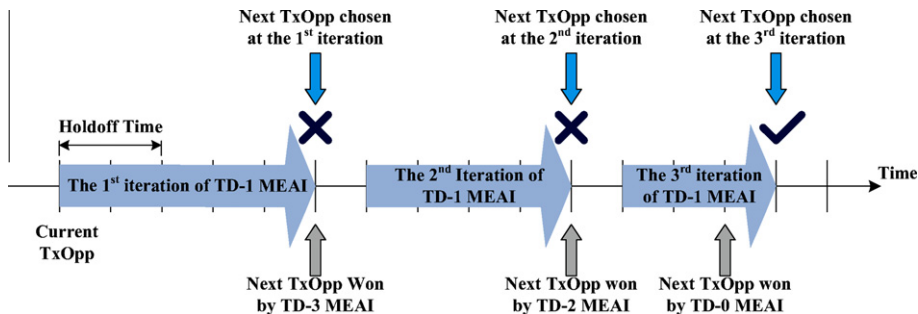


Fig. 3. Example case of the operation of TMEA-S associated with TD 1.

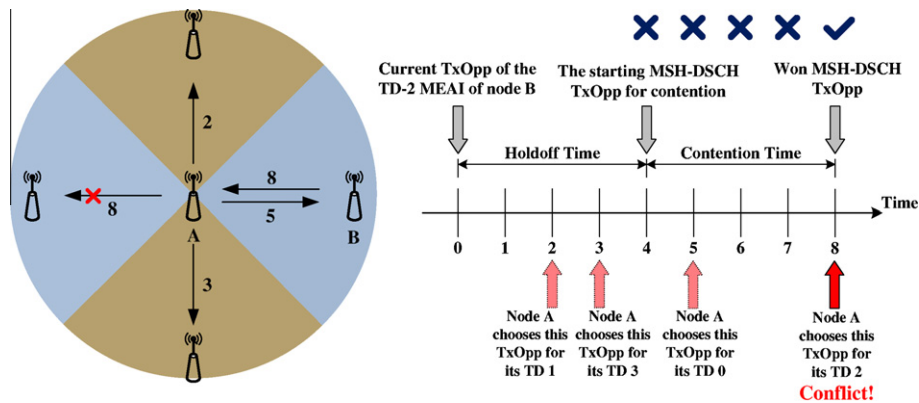


Fig. 4. Example case of the inter-node scheduling conflict problem.

information of node B's last MSH-DSCH message. Because there can be only one winner in any two-hop neighborhood, the TD-2 MEAI of node A will not choose the same TxOpp 8 as the TD-2 MEAI of node B did, resolving a potential inter-node scheduling conflict.

Recall that, when the next TxOpp of a neighboring node in node i 's two-hop neighborhood is unknown, node i should include this node into its own eligible node lists for subsequent TxOpps, until receiving the next TxOpp information of this neighboring node. We expand this node-based conservative eligibility determination rule into a TD-based form: "When the next TxOpp of a neighboring node's MEAI in node i 's two-hop neighborhood is unknown, each MEAI of node i should include this MEAI into its own eligible node lists for subsequent TxOpps, until receiving the next TxOpp information of this MEAI." Using this conservative rule avoids another type of the inter-node scheduling conflicts resulting from unknown and obsolete next TxOpp information of neighboring nodes.

For example, for node i , when the next MSH-DSCH message of node j that are two-hop away from it cannot be received before node i schedules its own next MSH-DSCH TxOpp, using this rule each MEAI of node i will include node j into its own eligible node list and conservatively consider that an MEAI of node j will contend for each subsequent TxOpp. Thus, when an MEAI of node i wins a TxOpp, it will ensure that all node j 's MEAIs will not win the same TxOpp, preventing inter-node scheduling conflicts among two-hop nodes from occurring.

4.3. TMEA-D

To solve the inter-node scheduling conflicts, each TMEA-S of the same node's TD has to use an eligible node list derived from all $nbr(i_m)$, $\forall m$ on the same node, and the TD-based conservative eligibility determination rule. However, these two designs of TMEA-S may make the MEAIs of the same node derive nearly the same eligible node list. In this condition, the diversity of the inputs of MEA, which chooses a winning node of a given TxOpp using a pseudo-random hash function, is very limited. Thus, the winning TxOpp of each MEAI on the same node is likely to be the same at the first k iterations (which continuously re-

sults in intra-node scheduling conflicts), where k is the number of TDs on a node.

This is because in this condition each MEAI's eligible node list for the same TxOpp may have many common nodes (worse yet, their eligible node lists may be the same for many TxOpps due to the conservative TD-based eligibility determination rule), MEAIs on the same node, therefore, may find that MEA always chooses the same node, generating many intra-node scheduling conflicts. We call this problem the "continuous intra-node scheduling conflict" problem, which results in three drawbacks. First, it significantly increases the interval of each MEAI's transmission cycle and therefore increase the required time to negotiate data transmission; second, for the first reason, the utilization of the link bandwidth will be decreased; third, the MAC control unit will waste the time and computing power for the first k iterative executions of MEA, which is very inefficient to the implementation of MAC-layer control chips.

To address this problem, we propose TMEA-D to increase the diversity of the inputs fed into MEA. TMEA-D has three advantages. First, it provides more scheduling flexibility for MEAIs to avoid continuous intra-node scheduling conflicts by allowing each MEAI on the same node uses distinct holdoff time values. Second, TMEA-D can generate fair TxOpp scheduling for MEAIs on the same node. Finally, TMEA-D can assign active MEAIs (those that have data to send) smaller holdoff time exponent values and idle MEAIs larger holdoff time exponent values. Thus, using TMEA-D the time required for a transmitting node to handshake a minislot allocation can be reduced, which allows nodes to more efficiently utilize link bandwidth.

The operation of TMEA-D is shown in Algorithm 1. In the initialization phase, an MEAI first sets its `smallest_tmp_txopp` and `reference_start_txopp` to the current TxOpp and constructs its L_{ij} . It then empties the set S_{calc} , which is used to store the TxOpp numbers that are chosen by this MEAI when executing Algorithm 1 but have been used by other MEAIs. If the node has data to send in the TD, the MEAI of this TD randomly chooses a holdoff time exponent value exp from S_{active} . S_{active} is composed of smaller exp values from 0 to the number of TDs where this node

has data to send. We call such a TD an active TD and denote the number of active TDs as Num_of_ActTDs in Algorithm 1. The chosen *exp* value is stored in the tmp_exp variable and used to derive the contention_start_txopp, which is the reference_start_txopp plus 2^{tmp_exp} . Then, an MEAI calculates the next TxOpp that it can win (stored in tmp_txopp) using MEA with the chosen contention_start_txopp value and L_{ij} . If the chosen next TxOpp is not used by other MEALs on the same node, it first calculates proper *exp* and *offset* values for this chosen TxOpp number and then return the 3-tuple (tmp_txopp, proper_exp, proper_offset) as its output. Otherwise, it adds the chosen next TxOpp number into S_{calc} , removes the used tmp_exp value from S_{active} , and repeats the above process until it wins a TxOpp that has not been used by other MEALs on the same node. (Note that, for brevity, the calculation for the proper_exp proper_offset values shown in

this paper does not consider the sequence number wrapping problem of tmp_txopp, which has been properly solved in our implementation.)

In case that S_{active} becomes empty, it means that this MEAI cannot find a tmp_txopp value that has not been won by other MEALs on the same node using the contention_start_txopp values derived from the current smallest_tmp_txopp value and the tmp_exp values in S_{active} . To address this problem, the MEAI advances its smallest_tmp_txopp to the minimum TxOpp stored in the S_{calc} (which is the smallest TxOpp that it wins in the iterations of the while loop) and then re-performs the above process. By doing so, this MEAI can have a chance to win a TxOpp that has not been won by other MEALs belonging to the same node. Such an iterative process repeats until the MEAI finally wins an unused TxOpp that has not been chosen by other MEALs on the same node.

Algorithm 1. TMEA-D

```

1:  Num_of_ActTDs := the number of TDs on the same node where there is data to send
2:   $L_{ij} := \{(k, L_{ij}(k)) | \forall \text{TxOpp } k, L_{ij}(k) \text{ is the eligible node list of node } i\}$ 's TD  $j$ ,
3:  constructed based on the eligibility of all nodes in  $\cup nbr(i_m), \forall \text{TD } m \in \text{node } i$ 
4:  smallest_tmp_txopp := current TxOpp
5:  reference_start_txopp := current TxOpp
6:   $S_{calc} := \emptyset$ 
7:   $S_{active} := \{0, 1, 2, 3, \dots, \min(\text{Num\_of\_ActTDs} - 1, \text{max\_exp})\}$ 
8:   $S_{calc} \leftarrow \emptyset$ 
9:  if there is data to send in my TD then
10:     found_flag := false
11:     while  $S_{active} \neq \emptyset$  do
12:         tmp_exp := random( $S_{active}$ )
13:         contention_start_txopp := smallest_tmp_txopp +  $2^{tmp\_exp}$ 
14:         tmp_txopp  $\leftarrow$  MEA(contention_start_txopp,  $L_{ij}$ )
15:         if tmp_txopp has been used by other MEALs on the same node then
16:              $S_{calc} \leftarrow S_{calc} \cup \{\text{tmp\_txopp}\}$ 
17:              $S_{active} \leftarrow S_{active} - \{\text{tmp\_exp}\}$ 
18:         else
19:             found_flag  $\leftarrow$  true
20:             break
21:         end if
22:     end while
23:     if found_flag = false then
24:         smallest_tmp_txopp  $\leftarrow$  min ( $S_{calc}$ )
25:         goto line 7
26:     end if
27: else
28:     found_flag := false
29:     if min (Num_of_ActTDs, max_exp)  $\leq$  4 then
30:         starting_exp := 4
31:     else
32:         starting_exp := min (Num_of_ActTDs, max_exp)
33:     end if
34:     for tmp_exp  $\leftarrow$  starting_exp to max_exp do
35:         contention_start_txopp := smallest_tmp_txopp +  $2^{tmp\_exp}$ 
36:         tmp_txopp  $\leftarrow$  MEA(contention_start_txopp,  $L_{ij}$ )
37:         if tmp_txopp has been used by other MEALs on the same node then
38:              $S_{calc} \leftarrow S_{calc} \cup \{\text{tmp\_txopp}\}$ 

```

(continued on next page)


```

39:     contention_start_txopp ← contention_start_txopp + 1
40:     else
41:         found_flag ← true
42:         break
43:     end if
44: end for
45: if found_flag = false then
46:     smallest_tmp_txopp ← min( $S_{calc}$ )
47:     goto line 8
48: end if
49: end if
50: proper_exp := floor( $\log_2(\text{tmp\_txopp} - \text{reference\_start\_txopp})$ )
51: proper_offset := (tmp_txopp - reference_start_txopp) -  $2^{\text{proper\_exp}}$ 
52: return (tmp_txopp, proper_exp, proper_offset)

```

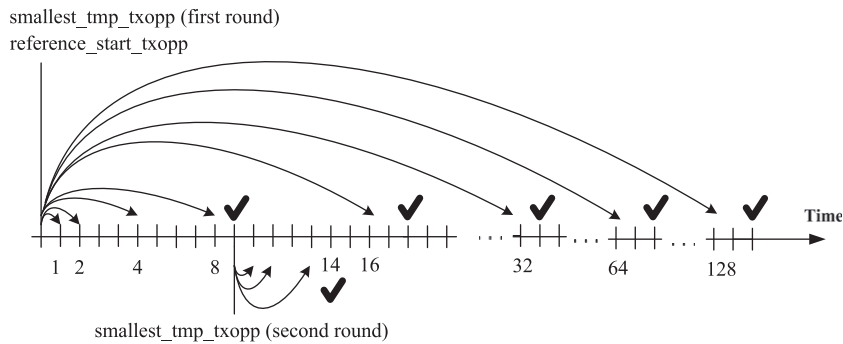


Fig. 5. An example of the iterations of TMEA-D.

Fig. 5 illustrates an example of how TMEA-D solves the above problem. Suppose that a node i has six MEALs and its TD-1 MEAL is calculating the next MSH-DSCH TxOpp. First, the TD-1 MEAL sets its `smallest_tmp_txopp` to 0 and then performs MEA to find its next TxOpp number. In the first round, it finds that the TxOpp numbers that node i can win have been used by other MEALs on node i (TxOpps 9, 18, 33, 66, and 130). As a result, it advances its `smallest_tmp_txopp` to 9 and restarts the next TxOpp finding process. In this round, it successfully wins TxOpp 14 that has not been won by other MEALs of node i .

The reason why TxOpp 14 can only be found in the second round is explained here. TxOpp 14 is between TxOpp 9 and TxOpp 18. The former is chosen when the exponent value is advanced to 3 (i.e., the holdoff time becomes $2^3 = 8$) and the latter is chosen when the exponent value is advanced to 4 (i.e., the holdoff time becomes $2^4 = 16$). According to the standard, when MEA wins TxOpp 9, it immediately returns that TxOpp as the output. Later on, when MEA finds that TxOpp 9 has been won by another MEAL and a new one should be found, it advances the exponent value from 3 to 4 and starts the searching from TxOpp $16 = 0 + 16$ (where 0 is the value of `smallest_tmp_txopp` and 16 is the current holdoff time 2^4). However, this exponential holdoff time expansion causes TxOpp 14 to be skipped in the search during the first round. At the second round, because `smallest_tmp_txopp` is moved to TxOpp 9 and the

exponent value starts over from 0 again, the search can start from TxOpp $10 = 9 + 2^0$ and eventually find TxOpp 14.

For the case where there is no data to send in the TD, the MEAL need not use a small next TxOpp to transmit its next control message. Thus, it first determines the starting holdoff time exponent value (stored as `starting_exp`) using the minimum between `Num_of_ActTDs` and `max_exp`.¹ If this value is below 4, then MEAL adjusts it to 4. The rationale behind this design is that the number of active TDs of a node may dynamically fluctuate. Because the node has no data to send in this TD, the control message dissemination of this TD is not time-critical. Thus, TMEA-D prevents the MEAL of an idle TD from using small holdoff time exponent values, which are more valuable for MEALs of active TDs to reduce the time required for negotiating minislot allocations.

After determining the `starting_exp` value, the MEAL of an idle TD iteratively finds its next TxOpp using holdoff time exponent values from `start_exp` to `max_exp`. The calculation in this iterative process is similar to that used by an MEAL of an active TD. The main difference is that an MEAL of an active TD chooses a smaller holdoff time exponent value from S_{active} in a random manner while an MEAL of an idle TD chooses a larger holdoff time exponent value

¹ The maximum holdoff time exponent value is defined as 7 in the standard.

from $[\max(4, \min(\text{Num_of_ActTDs}, \max_exp)), \max_exp]$ in an iterative manner. For brevity, the same explanation for these calculations is not repeated here.

The MEAs of active TDs on the same node share the useable smaller holdoff time values in a random manner, which prevents some of them from monopolizing the smallest holdoff time values and thus ensures a fair sharing of these valuable small holdoff time values among them in a long term. Thus, for MEAs of active TDs on the same node, the average times required by them to negotiate minislot allocations (to transmit data) can be the same.

4.4. Problem 3: Transmission-domain-aware Minislot Scheduling

The 802.16(d) mesh CDS-mode network uses a THP to negotiate minislots for data transmission [1]. The scheduling algorithm for allocating minislots has not been standardized. Thus, a minislot scheduling algorithm that can exploit the spatial-reuse property of SSBAs is desired for our work. In this section, we propose an easy-to-implement Transmission-domain-aware Minislot Scheduling Algorithm (TMSA) which need not modify the control messages used in the original THP. We explain its operation below.

TMSA defines four types of minislot allocations for a node x . The first one is the “local node transmission” type (denoted as “LOCAL_XMIT”), indicating that a minislot allocation is used by node x to transmit data packets to its neighboring node; the second one is the “local node reception” type (denoted as “LOCAL_RECV”), indicating that a minislot allocation is used by node x to receive data packets from its neighboring node; the third one is the “neighboring node transmission” type (denoted as “NBR_XMIT”), which indicates that a minislot allocation is used by a node x 's neighboring node to transmit data packets to another neighboring node (other than node x); and the final one is the “neighboring node reception” type (denoted as “NBR_RECV”), which indicates that a minislot allocation is used by a node x 's neighboring node to receive data from another neighboring node (other than node x). For brevity, we use the notation ($x \rightarrow y$) to represent a minislot allocation

that is used by node x to transmit data packets to node y .

In TMSA, the description of a minislot allocation is represented by a 7-tuple pair (TDI, NID, type, SFN, validity, SMN, MR), where the TDI field denotes the transmission domain index. The type field denotes the type of this minislot allocation; SFN denotes the starting frame number of this minislot allocation, validity denotes the number of frames that this minislot allocation lasts; SMN denotes the starting minislot number of this minislot allocation within a frame; and MR (Mini-slot Range) denotes the number of minislots occupied by this minislot allocation within a frame. The interpretation of the NID field depends on the value of the type field. The NID field represents the ID of the receiving node when the following type field is “LOCAL_XMIT.” Otherwise, it represents the ID of the transmitting node. The allocations of “NBR_XMIT” and “NBR_RECV” types are learned from received MSH-DSCH messages.

We use the example shown in Fig. 6 to illustrate the operation of TMSA. In this example, node D has scheduled a minislot allocation ($D \rightarrow C$) that occupies minislots ranging from 5 to 29 during the frames from 20 to 147 and node F has scheduled a minislot allocation ($F \rightarrow E$) that occupies the same duration. From the perspective of node A, the former minislot allocation is represented as two 7-tuple pairs (0, D, NBR_XMIT, 20, 128, 5, 20) and (1, C, NBR_RECV, 20, 128, 5, 20), and the latter minislot allocation is represented as another two 7-tuple pairs (3, F, NBR_XMIT, 20, 128, 5, 20) and (0, E, NBR_RECV, 20, 128, 5, 20).

Suppose that node A wants to schedule its minislot allocation ($A \rightarrow B$) in the same duration. Using TMSA, node A first checks its minislot allocation list to see whether there is any minislot allocation with the LOCAL_XMIT or LOCAL_RECV type in that duration. If yes, it cannot schedule any minislot allocation in this duration because it cannot simultaneously transmit or receive packets in two different TDs. In this condition, node A should try to schedule ($A \rightarrow B$) within a different duration.

In case no such a minislot allocation exists in the list, node A then checks whether the list contains any learned

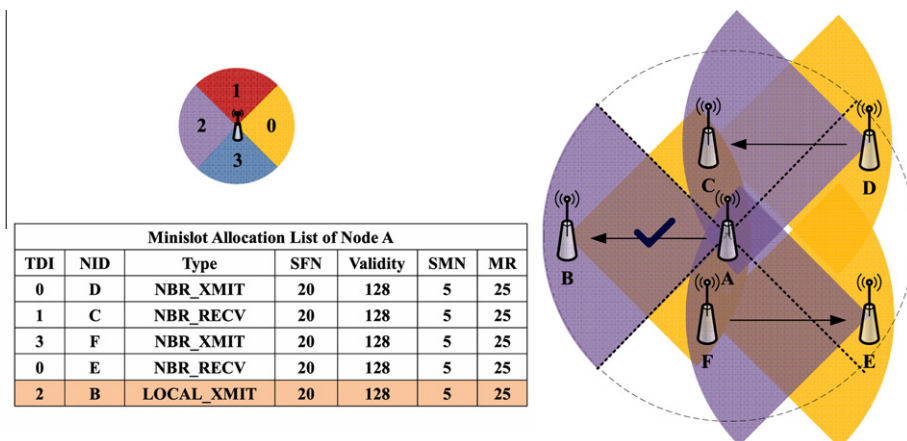


Fig. 6. An example of data scheduling in a network using SSBAs.

NBR_XMIT allocations with node B being the transmitting node for the same duration. If yes, it should not schedule ($A \rightarrow B$) because node B cannot receive its packets in that duration. If no, node A finally checks whether the list contains a learned NBR_RECV allocation in the TD where node B resides for the same duration. If yes, it should not schedule ($A \rightarrow B$); otherwise, this transmission will interfere with the neighboring node's data packet reception. If no, ($A \rightarrow B$) can be scheduled. In this example, because no such an allocation exists, node A can schedule this ($A \rightarrow B$) at the same time when ($D \rightarrow C$) and ($F \rightarrow E$) take place. This example scheduling shows that the proposed TMSA can utilize the spatial-reuse advantage of SSBA to increase network capacity on the data plane.

5. Performance evaluation

In this section, we use the NCTUns network simulator [20] to evaluate the performances of our proposed scheme. NCTUns is an advanced network simulator with two unique advantages: (1) directly using the real-world TCP/IP protocol stacks in the Linux kernel; and (2) allowing real-world network application programs to run over its simulated networks. By these two features, NCTUns can provide high-fidelity simulation results with real implementations of TCP and application traffic generator.

5.1. Simulation environment

We created a 25-node grid network topology shown in Fig. 7 as our simulation topology. The main parameter settings used in the simulations are listed in Table 1. Two different traffic types are used to generate network traffic. One is TCP and the other is greedy UDP. The word "greedy" means that the source node of a flow will transmit data as many as it can.

In a simulation case, each node sets up a greedy UDP flow to each of its 1-hop neighboring nodes. UDP is a

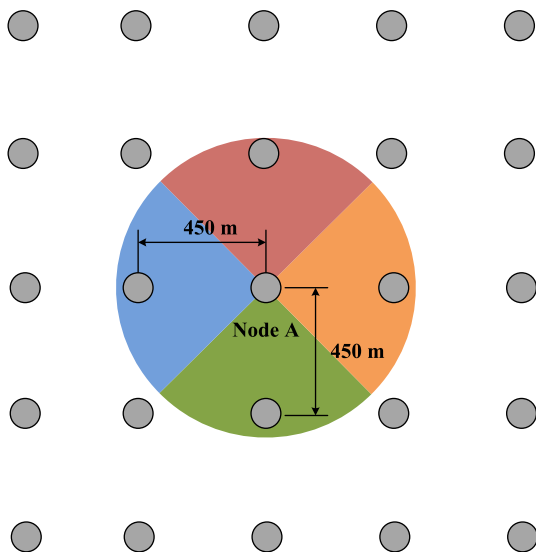


Fig. 7. The topology of the simulated network.

Table 1

The parameter settings used in the simulations.

Parameter name	Value
Number of TxOpps per frame	8
Number of TxOpps per frame used by the CDS mode	8
Number of mini-slots per frame	220
Number of OFDM symbols per mini-slot	3
Requested mini-slot size per THP	10, 20, 30, 40, 50, 60, 70
Requested frame length	$2^0, 2^1, 2^2, 2^3, 2^5, 2^7$
Modulation/CODING SCHEME	64QAM-3/4
Channel model	ECC-33
Frame duration	10 ms
TCP version	Binary increase Congestion control TCP (BIC-TCP)

transportation layer protocol that simply transmits data down to the MAC-layer; thus, the throughputs obtained by all greedy UDP flows of a node are equivalent to the MAC-layer throughputs that can be obtained by the nodes minus the bandwidth overheads introduced by MAC-layer, network-layer, and transport-layer headers. For convenience, we use aggregate UDP throughputs to evaluate the capacity gain of an 802.16(d) mesh CDS-mode network using SSBA.

In addition to evaluate the raw network capacity, we also use TCP flows to conduct simulations in the same scenario. TCP is a complicated transportation-layer protocol that employs sophisticated error-control and congestion-control mechanisms to guarantee error-free in-order data delivery. It is widely used by many network applications, e.g., the File Transfer Protocol (FTP) and the HTTP web access. Thus, observing how TCP works over this new network using SSBA is important for transport-layer study and application development.

Each of our simulation case was run ten times, each time using a different random number seed. The simulated time of each run was 500 s. In a simulation, the traffic generator programs are activated at the 200-th second of the simulated time. This arrangement is to ensure that they transmit data packets after the simulated network has been stabilized.² In each of the figures shown below, both the average and the standard deviation of the collected simulation results are presented.

In the simulated grid network, each node is spaced 450 m away from its neighboring nodes and equipped with an SSBA. The gain pattern of the used SSBA on the horizontal plane is plotted in Fig. 8, which is derived from [28]. For wireless channel modeling, we conducted simulations with several channel models, such as the two-ray model, Erceg's model with terrain type B, and the ECC-33 model. In our experiences, in absence of dynamic fading effects (e.g., the Rayleigh fading), by properly setting the physical-layer parameters and the distance between nodes, the link connectivity and signal quality experienced by nodes can be adjusted to the same level. Thus, to save space we

² A stabilized network is defined as a network in which all of its nodes have joined the network.

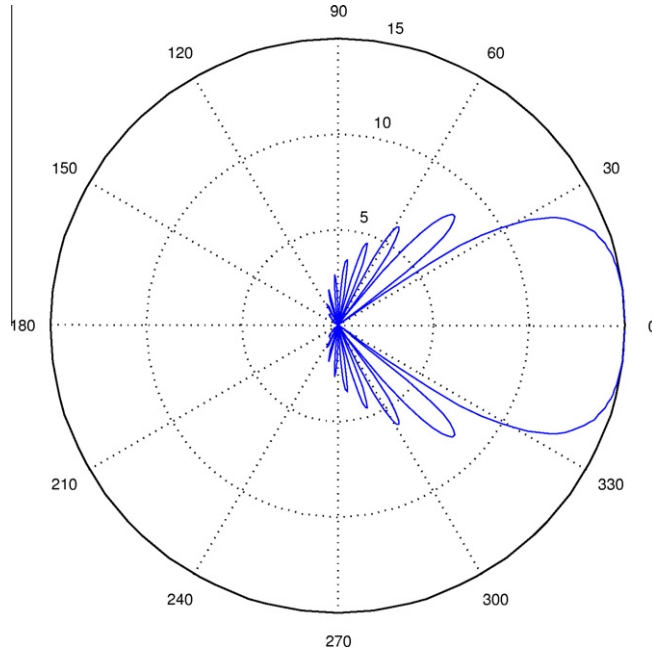


Fig. 8. Gain pattern of the used SSBA.

only present the simulation results using the ECC-33 model in this section and discuss the effects of the Rayleigh fading in Section 5.3.4.

Without the presence of the dynamic fading, the effective transmission range and interference range in our simulations are adjusted to 500 and 850 m, respectively. With the presence of the dynamic fading, there are no explicit values for the transmission range and the interference range of a node. In this condition, a proper receive power threshold is set for each simulation case to maximize flow throughputs. Because the antenna gain patterns of the used SSBA and the omnidirectional antenna greatly differ, the transmit powers of the radios used in the MTD and STD schemes were set to 22 dBm and 50 dBm, respectively, to maximize their respective network performances.

The ECC-33 model was originally proposed for modeling the path loss effect over 3.5 GHz omnidirectional radios. For this reason, when SSBA are used, the transmit and receive antenna gains used in this model will be re-computed using the gain pattern shown in Fig. 8.

5.2. Performance metrics

Two performance metrics are used in this work: (1) the Average Throughput of a UDP Flow in a network (ATUF) and (2) the Average Throughput of a TCP Flow in a network (ATTF). In addition, the coefficients of variation (CVs) of the ATUF and ATTF values of all flows in a network are presented to study the fluctuation degree of application throughputs under the evaluated schemes. In addition to these application-layer metrics, we use two MAC-layer performance metrics to evaluate the performance gain of the proposed scheme. One is the Average TxOpp Utilization of Nodes (ATOUN) and the other is the Average Three-way

Handshake Procedure Time (ATHPT). The definitions of these performance metrics are given below.

5.2.1. ATUF and ATTF

The ATUF metric is defined as the average throughput of a UDP flow across all network nodes, which is defined as follows:

$$\text{ATUF} = \frac{\text{TTAF}}{\text{TNST}}, \quad (8)$$

where TTAF (total throughputs of all flows) denotes the sum of all nodes' average UDP-flow throughput samples across the different runs of a case, which is defined by the following equation:

$$\text{TTAF} = \sum_{m=1}^{nc} \sum_{i=1}^{nn} \sum_{n=ts_time+1}^{te_time-1} \text{udp_th}_{ijm}(n), \quad \forall j \in \text{nbr}_1(i), \quad (9)$$

where nc denotes the number of runs that a simulation case was performed and nn denotes the number of nodes in the simulated network. The ts_time denotes when a UDP-flow sender program starts transmitting data packets in seconds and the te_time denotes when a UDP-flow sender program stops transmitting data packets in seconds. In our simulations, ts_time and te_time are set to 200 and 499 s, respectively. The $\text{udp_th}_{ijm}(n)$ denotes the average throughput of a UDP flow from node i to node j during the n th second in the m th run. The definition of $\text{nbr}_1(i)$ has been given in Eq. (2).

TNST (total number of sampled throughput³) denotes the total number of sampled UDP-flow throughputs across all flows and all runs of a case and is defined as follows:

³ In our simulations, a flow is sampled every one second to record its average throughput in the past one second.

$$\text{TNST} = \sum_{m=1}^{nc} \sum_{i=1}^{nm} \sum_{n=ts_time+1}^{te_time-1} |nbr_1(i)|. \quad (10)$$

The definition of ATTF is similar to that of ATUF, except that ATTF uses the following formula to calculate its TTAF:

$$\text{TTAF} = \sum_{m=1}^{nc} \sum_{i=1}^{nm} \sum_{n=ts_time+1}^{te_time-1} tcp_th_{ijm}(n), \quad \forall j \in nbr_1(i), \quad (11)$$

where $tcp_th_{ijm}(n)$ denotes the average throughput of a TCP flow from node i to node j during the n th second in the m th run.

5.2.2. CV-ATUF and CV-ATTF

The coefficients of variation (CV) of ATUF and ATTF values (denoted as CV-ATUF and CV-ATTF) are used to estimate the fluctuation degree of a traffic flow's throughput during simulation. The zero value of CV-ATUF (or CV-ATTF) indicates that the flow throughput in the network is stabilized at any given time, while a large CV-ATUF (CV-ATTF) value indicates the flow throughput in the network fluctuates greatly over time. The definition of CV-ATUF is defined as follows:

$$\text{CV-ATUF} = \frac{\text{StdDev_ATUF}}{\text{ATUF}}, \quad (12)$$

where StdDev_ATUF denotes the standard deviation of all UDP-flow throughput samples in a case, which is defined by the following equation:

$$\text{StdDev_ATUF} = \sqrt{\frac{\sum_{m=1}^{nc} \sum_{i=1}^{nm} \sum_{n=ts_time+1}^{te_time-1} (udp_th_{ijm}(n) - \text{ATUF})^2, \quad \forall j \in nbr_1(i)}{\text{TNST}}}. \quad (13)$$

The definition of CV-ATTF is similar to that of CV-ATUF and is shown as follows:

$$\text{CV-ATTF} = \frac{\text{StdDev_ATTF}}{\text{ATTF}}, \quad (14)$$

where StdDev_ATTF denotes the standard deviation of all TCP-flow throughput samples in a case, which is defined as follows:

$$\text{StdDev_ATTF} = \sqrt{\frac{\sum_{m=1}^{nc} \sum_{i=1}^{nm} \sum_{n=ts_time+1}^{te_time-1} (tcp_th_{ijm}(n) - \text{ATTF})^2, \quad \forall j \in nbr_1(i)}{\text{TNST}}}. \quad (15)$$

5.2.3. ATOUN

The ATOUN metric evaluates the efficiency of utilizing the control-plane bandwidth for a network. The average TxOpp utilization viewed from a node j using SSBA can be defined as follows:

$$\text{AvgTxOppUtil}(j) = \frac{\left(\sum_{k=0}^{\text{NumTD}-1} \sum_{i \in (nbr(j_k) - \{j\})} \text{txnum}(i) \right) + \text{txnum}(j)}{\text{total}(j)}, \quad (16)$$

where $\text{txnum}(i)$ denotes the number of TxOpps won by a node i ; $\text{total}(j)$ denotes the number of total TxOpps that have elapsed since node j joins the network; NumTD

denotes the number of TDs that node j has; and the definition of $nbr(j_k)$ has been given in Eq. (7). On the other hand, the average TxOpp utilization viewed from a node j using an omnidirectional antenna can be defined as:

$$\text{AvgTxOppUtil}(j) = \frac{\sum_{i \in nbr(j)} \text{txnum}(i)}{\text{total}(j)}. \quad (17)$$

The ATOUN metric is defined as the average of the AvgTxOppUtil values across all nodes in a case. The definition of ATOUN is shown as follows:

$$\text{ATOUN} = \frac{\sum_{j=1}^m \text{AvgTxOppUtil}(j)}{m}, \quad (18)$$

where m is the number of nodes in a simulation case. Each ATOUN result presented in Section 5.3 is the average across ten runs. A higher value of ATOUN indicates that a network case has a higher utilization of TxOpps while a lower value of this metric indicates that a network case has a lower utilization of TxOpps.

5.2.4. ATHPT

The ATHPT metric is defined as the average time required to complete a THP across all nodes in a case. A node is allowed to transmit data packets to another node only after it has scheduled a minislot allocation with that node. For this reason, ATHPT significantly influences the packet delay time experienced by applications. Its definition is explained here. We first define THPT (i) as the average time required to establish minislot allocations for a node i in a case, which is given below:

$$\text{THPT}(i) = \frac{\sum_{j=1}^n t_{ij}}{n}, \quad (19)$$

where t_{ij} denotes the time required to establish the j th minislot allocation of node i with one of its neighboring nodes and n denotes the number of minislot allocations that node i establishes during simulation. We then compute the ATHPT value of a case as follows:

$$\text{ATHPT} = \frac{\sum_{i=1}^m \text{THPT}(i)}{m}, \quad (20)$$

where m is the number of nodes in a simulation case. Similar to ATOUN, each ATHPT result presented in Section 5.3 is the average across 10 runs.

5.3. Simulation results

5.3.1. Effects of holdoff exponent values

In this section, we studied the effects of nodes' holdoff times on network performances. For the STD scheme, a given holdoff time exponent value x means that all nodes use the 2^x TxOpps as their holdoff times in simulations, while for the MTD scheme a given holdoff time exponent value x means that the S_{active} set used in the TMEA-D algorithm is $\{x, x+1, x+2, \dots, \min(x + \text{Num_of_ActTDs} - 1, \text{max_exp})\}$, where x is from 0 to 4 in our simulations. In this series of simulations, the requested frame duration and the number of requested minislots per frame in a THP were set to 32 frames and 30 minislots, respectively.

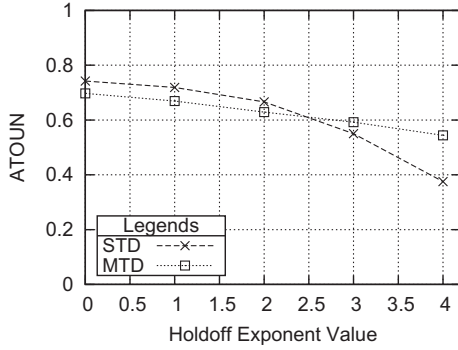


Fig. 9. ATOUN results over different holdoff time exponent values.

Fig. 9 shows the ATOUN results of the STD and MTD schemes over different holdoff time exponent values. As one knows, increasing the holdoff time exponent value will increase the transmission intervals of nodes' control messages. Thus, the TxOpp utilization of the network will decrease, when this value increases. One interesting phenomenon is that the TxOpp utilization of the STD scheme is higher than that of the MTD scheme, when the holdoff time exponent value is very small (i.e., below 2), but drops more rapidly than that of the MTD scheme, when this value increases. We explain this phenomenon from two aspects. First, when using the MTD scheme a node i uses multiple MEALs to manage its transmission domains. Due to this design, from the perspective of node i 's neighboring nodes, on TxOpp T the set of TxOpps for which node i will not contend is denoted as $S_{inact_tx}^i(T)$ and given as follows:

$$S_{inact_tx}^i(T) := \cap H_j^i(T), \quad 0 \leq j \leq \text{Num_of_ActTDs} - 1, \quad (21)$$

where $H_j^i(T)$ denotes the holdoff time interval of node i 's MEAL j known by node i 's neighboring nodes on TxOpp T . For node i , it is impossible that $H_j^i(T)$, \forall existing MEAL j , exactly overlap. Thus, the size of $S_{inact_tx}^i(T)$ in the MTD scheme is much less than that in the STD scheme. This means that nodes in the MTD scheme more conservatively choose TxOpps to transmit control messages to avoid inter-node scheduling conflicts. As a result, when the flexibility of TxOpp scheduling is large (e.g. all nodes can use small holdoff time exponent values), the TxOpp utilization of the STD scheme can be better than that of the MTD scheme. Second, the reason why the TxOpp utilization of the MTD scheme decreases slower than that of the STD scheme, as the holdoff time exponent value increases, is explained here. The MTD scheme employs multiple MEALs to manage directional control message transmissions and all of these MEALs need to find a conflict-free TxOpp to transmit their control messages in their respective TDs. Nodes using the MTD scheme therefore needs to consume more TxOpps than the STD scheme. As a result, when nodes' holdoff time intervals becomes large, nodes in the MTD scheme will use more TxOpps than those in the STD scheme, which makes the TxOpp utilization of the MTD scheme drops more slowly than that of the STD scheme.

However, as can be seen in Fig. 10, the time required for a node to establish a minislot allocation is insensitive to

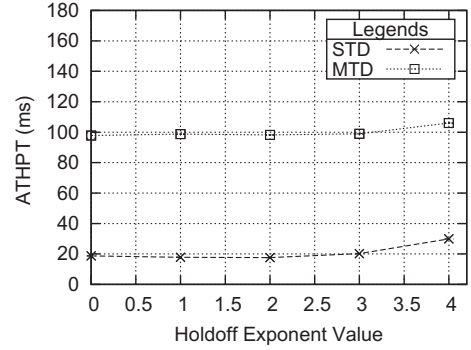


Fig. 10. ATHPT results over different holdoff time exponent values.

the holdoff time exponent value when it is below 3. This is because the procedure to establish a minislot allocation is three-way based, which should finish a “requester–granter–requester” control message transmission sequence. Due to the randomness of the distributed TxOpp scheduling used in the 802.16(d) mesh CDS mode, the requesting and granting nodes may not always achieve the most efficient TxOpp scheduling to minimize the time for establishing a minislot allocation. In addition, although decreasing the holdoff time exponent value increases the TxOpp scheduling flexibility of nodes, it does not guarantee that a node that is performing a THP can always win a smaller TxOpp. (This is affected by how many MEALs on this node and its neighboring nodes are performing THPs at the same time.) For these reasons, even when the holdoff time exponent value is set to very small (e.g. no more than 3), the average of the time required for establishing a minislot allocation remains the same.

As one also sees, a larger holdoff time exponent value (e.g., 4) can increase the time required for establishing a minislot allocation. These results confirm the findings presented in [5,6,2,21]. Another noticeable phenomenon is that the average minislot allocation establishment time of a node using the MTD scheme is greatly higher than that of a node using the STD scheme. Such a phenomenon results from two reasons. One is that, due to the directivity of SSBAs, a node using the MTD scheme cannot exchange control messages with a specific peer node on every TxOpp that it wins. Instead, it is only allowed to communicate with a specific peer node on TxOpps won by its MEAL that manages the TD where this peer node resides. In contrast, nodes using the STD scheme can communicate with any of its neighboring nodes on every TxOpp that it wins. Due to this difference, nodes using the MTD scheme require more time to complete a THP and obtain a minislot allocation.

However, because the MTD scheme can utilize the spatial reuse advantage of SSBAs, it still outperforms the STD scheme on UDP and TCP flow throughputs. As shown in Figs. 11 and 12, the MTD scheme can on average outperform the STD scheme on ATUF by a factor of 2.71 and on ATTF by a factor of 5.88, regardless of the used holdoff time exponent value. The reason why TCP performs worse than UDP on average throughput is that TCP uses a complicated congestion control algorithm to prevent network bandwidth from being exhausted by a single flow, which

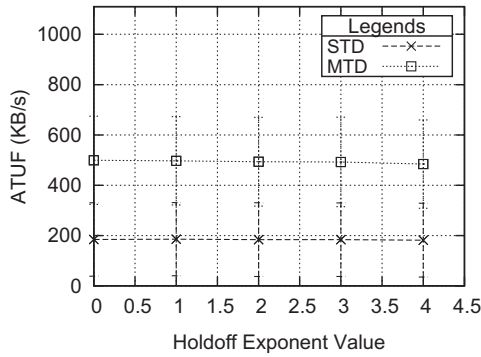


Fig. 11. ATUF results over different holdoff time exponent values.

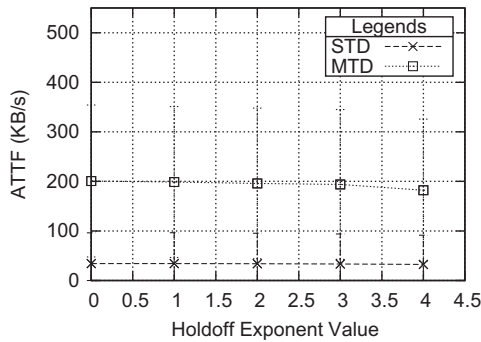


Fig. 12. ATTF results over different holdoff time exponent values.

usually regards packet losses as an indication of network congestion. Because the IEEE 802.16(d) mesh CDS mode schedules minislots in a distributed manner, the time for a node to obtain a minislot allocation may fluctuate and the number of minislots obtained in a minislot allocation may greatly vary. In this condition, an outgoing network interface needs to temporarily store packets in its own packet queue. If the packet queue of an interface becomes full, packets sent from upper-layer applications will be dropped, which may make TCP unnecessarily reduce its congestion window size and under-utilize link bandwidth.

In this section, we showed that the holdoff time exponent value has great impacts on TxOpp utilization. However, because the IEEE 802.16(d) mesh CDS mode uses a distributed three-way handshake design to schedule minislot allocations on the data plane, as long as the used holdoff time exponent value is not too large (e.g. above 4), the average time for nodes to negotiate a minislot allocation is insensitive to the holdoff time exponent value. Thus, the average UDP and TCP flow throughput results of both the MTD and STD schemes are unchanged when the holdoff time exponent value is between 0 and 4.

5.3.2. Effects of requested frame duration per THP

In this section, the holdoff time exponent value was set to 0, for maximizing the scheduling flexibility of both schemes. The number of requested minislots per frame in a THP was set to 30. Following [1], the requested frame duration for a minislot allocation was set to $2^0, 2^1, 2^2, 2^3,$

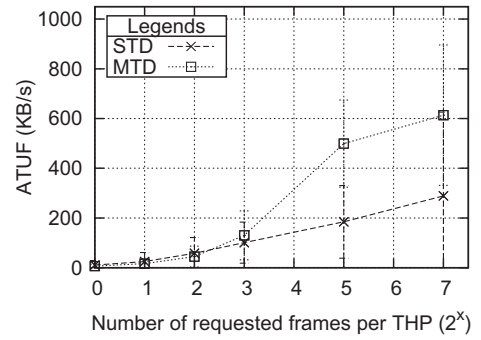


Fig. 13. ATUF results over different requested frame durations in a THP.

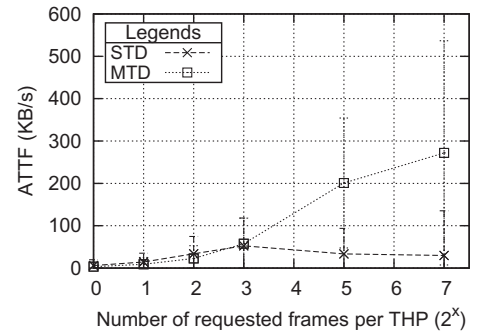


Fig. 14. ATTF results over different requested frame durations in a THP.

$2^5,$ and $2^7,$ respectively. Figs. 13 and 14 show the ATUF and ATTF results over different requested frame duration in a THP, respectively. One intuitive result is that increasing the requested frame duration in a THP can increase the utilization of minislots, which results in increased UDP and TCP flow throughputs for both of the evaluated schemes.

A noticeable phenomenon is that, when the requested frame duration per THP is below 2^3 frames, the UDP and TCP flow throughputs achieved by the MTD scheme is only the same as those achieved by the STD scheme. This is because, as discussed previously, using the MTD scheme nodes on average need 100 ms (i.e., 10 MAC-layer frames) to obtain a minislot allocation. To prevent a node from monopolizing link bandwidth, in our implementation a node A will be triggered to perform a THP with a neighboring node B, only when it has data destined to node B and does not possess any valid minislot allocation granted by node B. Due to this design and the long ATHPT property of the MTD scheme, if the requested frame duration in each THP does not exceed $\lceil (ATHPT/Num_of_ActTDs) \rceil$ frames, nodes using the MTD scheme will not be able to schedule minislots as tight as those using the STD scheme. In contrast, when the requested frame duration greatly exceeds $\lceil (ATHPT/Num_of_ActTDs) \rceil$ frames, due to the spatial reuse advantage, the MTD scheme can greatly outperform the STD scheme on UDP and TCP flow throughput performances.

We plotted the ATHPT results of the two evaluated schemes under the UDP and TCP traffic cases over different requested frame durations in Figs. 15 and 16. The results

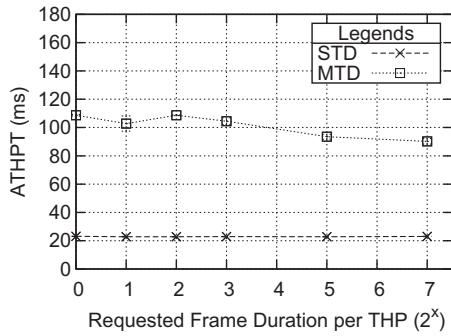


Fig. 15. ATHPT results over different frame durations under the UDP flow case.

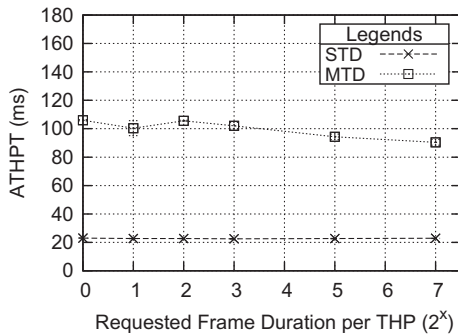


Fig. 16. ATHPT results over different frame durations under the TCP flow case.

show that the time required for a node to complete a THP is less related to the requested frame durations.

5.3.3. Effects of MAC-layer traffic loads

In this section, we changed the number of requested minislots per frame in a THP from 0 to 70, to generate different MAC-layer traffic loads. The holdoff time exponent value and the requested frame duration in a THP were set to 0 and 2⁷ frames, respectively.

As shown in Fig. 17, the ATUF under the MTD scheme significantly outperform those under the STD scheme over most of the evaluated MAC-layer traffic loads. These

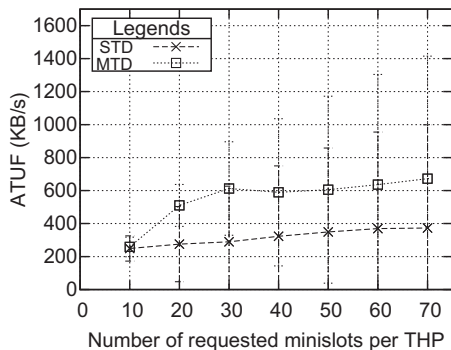


Fig. 17. ATUF results over different numbers of requested minislots in a THP.

results evidence that the MTD scheme can effectively exploits spatial-reuse advantages of SSBA to provide more network capacity. One may notice that, when the number of requested minislots per THP is 10 only, the ATUF results of the two evaluated schemes are close. This is because, when the generated traffic load is very light, the STD scheme can accommodate it without causing network congestion. However, when the number of requested minislots in a THP increases (i.e., the generated MAC-layer traffic load increases), the STD scheme quickly reaches its saturation point and cannot keep up with the performance of the MTD scheme.

Fig. 18 shows the CV-ATUF results over different numbers of requested minislots in THPs, which indicate the fluctuation degree of the achieved throughputs of flows in a network over time and the fairness of network bandwidth allocation among them. One can see that the CV-ATUF values of the MTD scheme are much lower than those of the STD scheme, showing that network applications can achieve a more stable throughput over time under the MTD scheme than under the STD scheme. These results also indicate that network bandwidth sharing among these competing UDP flows is fairer under the MTD scheme.

Fig. 19 shows the ATTF results of the two evaluated schemes over different numbers of requested minislots per THP. There are several findings about this figure. First,

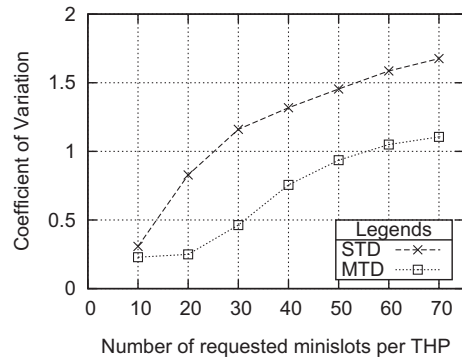


Fig. 18. CV-ATUF results over different numbers of requested minislots in a THP.

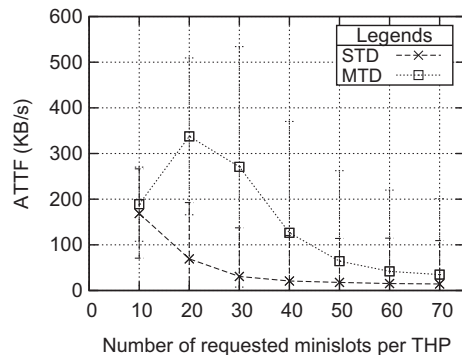


Fig. 19. ATTF results over different numbers of requested minislots in a THP.

the MTD scheme greatly outperforms the STD scheme over different traffic loads. Second, when the number of requested minislot is larger than 20 minislots, the average TCP flow throughputs achieved by the two evaluated schemes start to decrease. These results are explained here. When the number of requested minislots in each THP increases, the number of minislots that each node can obtain in a THP will drastically fluctuate. As explained in Section 5.3.1, when outgoing packets cannot be sent immediately, they will be stored in the packet output queue of the interface card. However, TCP speculates about the amount of data that it can transmit mainly based on detected packet losses and therefore it does not take the number of minislots obtained in each THP at the MAC-layer into account. For this reason, TCP may inject an excessive number of packets down to the MAC-layer and thus may generate packet dropping at the MAC-layer. Such packet dropping will trigger TCP's congestion control mechanism to reduce its transmission speed.

Although TCP is more sensitive to network congestion, as shown in Fig. 20, the MTD scheme still significantly outperforms the STD scheme on CV-ATTF, indicating that a TCP flow under the MTD scheme will achieve a more stable throughput over time than that under the STD scheme. These results also indicate that these competing TCP flows share the network bandwidth more fairly under the MTD scheme.

We finally studied the ATOUN and ATHPT results of the two schemes under MAC-layer loads. The results are as expected. Varying the number of minislots requested in THPs only affects how minislots on the data plane will be allocated; thus, changing this variable has minor effects on the TxOpps utilization. Varying the MAC-layer traffic load does not change the frequency that a node performs a THP, either; thus, it has minor effects on the time required for finishing a THP. For brevity, we do not present these results here.

5.3.4. Effects of Rayleigh fading

We finally studied the throughput performances of the MTD and STD schemes when Rayleigh fading is present. The presence of Rayleigh fading means that (1) there is no line-of-sight path for a pair of transmitting and receiving nodes and (2) there are some objects moving in the network. Such a scenario is possible to occur in

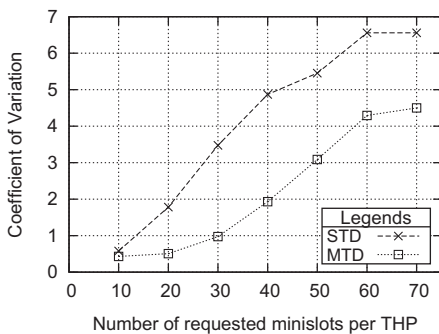


Fig. 20. CV-ATTF results over different numbers of requested minislots in a THP.

metropolitan areas where network nodes are surrounded by high buildings and many vehicles and pedestrians move at different speeds. The related discussion about using the small-scale fading (e.g., the Rayleigh fading) model on a non-omnidirectional scattering channel can be found in [29].

We plotted the obtained ATUF results and packet drop ratios over the Rayleigh fading variance in Figs. 21 and 22, respectively. A larger value of the Rayleigh fading variance means a higher fluctuation level of received signal power on a receiving node while a smaller value of the Rayleigh fading variance means a lower fluctuation level of received signal power on a receiving node.

By observing the results shown in Fig. 21, several findings are drawn. First, it is as expected that, when the fading variance value increases, the throughput performances of both the MTD scheme and the STD scheme drastically decrease. This is because, when the fading variance value is large, the power of received signal greatly fluctuates. On one hand, the reduction of received signal power due to Rayleigh fading increases the number of bit errors in received packets, which significantly decreases the goodputs obtained by nodes. On the other hand, the fluctuation of received signal power due to Rayleigh fading increases the signal interference level among neighboring nodes. In this condition, two bad effects result. One is that the Signal to Interference and Noise Ratio (SINR) values will be greatly decreased on receiving nodes, which also increases the number of bit errors in received packets. The other is that

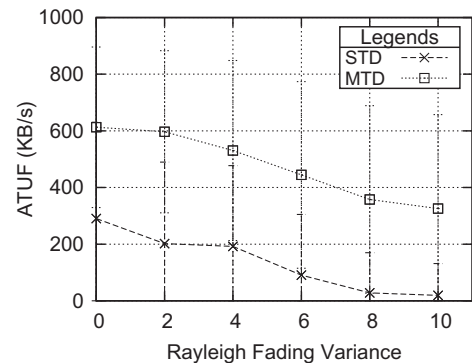


Fig. 21. The effects of the fading variance on UDP flow throughputs.

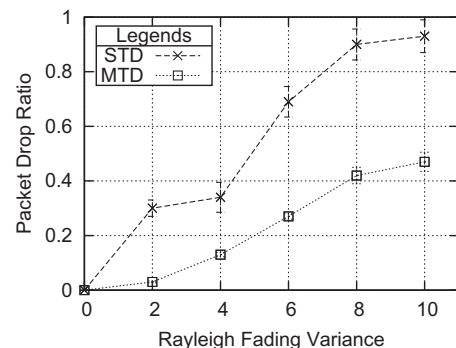


Fig. 22. The logged packet drop ratios over the fading variance.

the number of control message collisions and data packet collisions will be increased due to undesired signal interference. These effects can be evidenced by the packet drop ratio results shown in Fig. 22.

Although wireless transmission is difficult over a Rayleigh fading channel, one can see that nodes using the MTD scheme is superior than those using the STD scheme in resisting the signal strength fluctuation resulting from the Rayleigh fading effect. This phenomenon can be explained from two aspects. First, SSBA is capable of focusing its gain on only several directions. Thus, a node using an SSBA will not leak too much signal power in the directions that are not covered by the main lobe and the side lobes of its antenna. Second, due to the use of TMEA-D, nodes using the MTD scheme can know when and to which directions they should point their antennas to receive control messages and data. Such knowledge allows receiving nodes to achieve the largest receive antenna gain in the direction where the transmitting node is.

In addition, the core algorithm of TMEA-D considers the contention of all neighboring nodes in $\cup_{m \in \text{node } i} \forall \text{TD } m \in \text{node } i$. Thus, using the MTD scheme when node i transmits a control message to node j , nodes that are two-hop away from nodes i and j can avoid pointing their own antennas to these two nodes, which prevents themselves from being greatly interfered by node i 's transmission. In contrast, due to the omnidirectional transmission and reception property, nodes using omnidirectional antennas (such as those in the STD scheme) cannot alleviate the signal interference resulting from the Rayleigh fading effect.

Such signal inference on control message transmissions generates more collisions of MSH-DSCH and MSH-NCFG messages, resulting in two bad effects. One is that it is more difficult for nodes using the STD scheme to complete THPs (and thus obtain minislot allocations to transmit data). The other is that it is more difficult for nodes using the STD scheme to join the network because in this bad channel condition successfully receiving an MSH-NCFG message (which is necessary for the network entry process of a node) is more difficult for nodes using the STD scheme.

For the above reasons, nodes using the MTD scheme have higher probability to complete THPs (and therefore schedule data transmissions) than those using the STD scheme with the presence of Rayleigh fading, showing that the MTD scheme has higher efficiency on performing THPs than the STD scheme. In addition, nodes using the MTD scheme can successfully join the network with presence of Rayleigh fading. In contrast, using the STD scheme several nodes may fail to join the network in this condition. As a result, the throughput performances achieved by the MTD scheme are greatly better than those achieved by the STD scheme over wireless channels with the presence of the Rayleigh fading.

Fig. 23 shows the TCP flow throughput results of the two evaluated schemes over different Rayleigh fading variance values. One noticeable phenomenon is that, when the Rayleigh fading variance value increases, the average flow throughput achieved by TCP decreases more sharply than that achieved by UDP. This is because on a Rayleigh fading channel with a larger variance value the transmis-

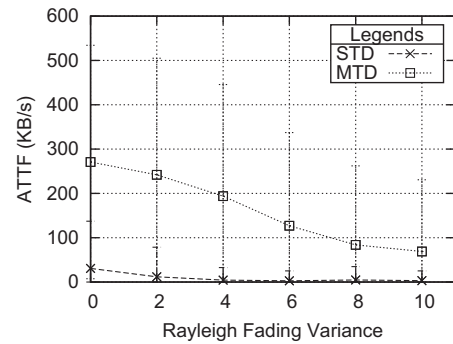


Fig. 23. The effects of the fading variance on TCP flow throughputs.

sions of MSH-DSCH messages are no longer guaranteed to be collision-free. In this condition, a THP may fail and thus the time required for a transmitting node to obtain a minislot allocation to transmit data may greatly vary over time. As explained previously, such fluctuation of the available link bandwidth may make TCP excessively transmit packets down to the MAC-layer, which generates undesired packet losses on the local interface. Upon detecting such packet losses, TCP will decrease its transmission speed and thus under-utilize the link bandwidth.

6. Discussion

The proposed MTD scheme successfully uses SSBA to enhance the performance of the IEEE 802.16(d) mesh CDS-mode network with minimum changes to the existing standard. Its performance has been comprehensively studied in Section 5.2. However, two issues of the MTD scheme should be noted. First, due to the use of multiple TDs, the time required for completing THPs greatly increases, as compared with the STD scheme. For this reason, the MTD scheme is more suitable for long-lasting traffic and aggregated traffic. (The latter is common for a relay node in wireless mesh networks.)

Second, to fully utilize the performance of SSBA, our proposal needs to add a new "offset" field at the end of control messages to solve the "Imprecise TxOpp Representation" problem, as discussed previously. This would decrease the compatibility between the standardized STD scheme and our proposed MTD scheme. Even if this compatibility issue can be solved by letting nodes using the standardized STD scheme ignore the added offset field, the performance of a network mixing the STD and MTD schemes will significantly decrease. This is because, for nodes using SSBA, both the flexibility of control message scheduling and data scheduling are decreased, when they are one-hop or two-hop away from nodes that still use the standardized imprecise representation of TxOpp in control messages or use omnidirectional antennas. This scheduling inefficiency may spread farther because the IEEE 802.16 mesh CDS mode uses a peer-to-peer handshake design to schedule data. Thus, the current MTD scheme can achieve its optimal performance only when all of the nodes in the network use it.

7. Conclusion

In this paper, we identify and address the important issues when equipping nodes with SSBA in an IEEE 802.16(d) mesh CDS-mode network. We propose a complete scheme to solve these issues and improve the performances of such networks. This scheme, called the MTD scheme, is implemented on the NCTUns network simulator. Its performances are extensively studied and compared against those of the original standard. Our simulation results show that, because of exploiting the spatial-reuse advantage of SSBA, the proposed MTD scheme significantly increases the network capacity of the IEEE 802.16(d) mesh CDS-mode network. Our simulation results also show that, as compared with the original omnidirectional network, the proposed MTD scheme can achieve more stable and fairer bandwidth sharing for applications.

Appendix A

A.1. Network Initialization with the use of SSBA

In an 802.16(d) mesh CDS-mode network, a new node is required to transmit its network-joining control messages on MSH-NENT TxOpps in a contention-based manner while a node that has been operational (called an operational node) should transmit its network-management control messages on MSH-NCFG TxOpps and its data-scheduling control messages on MSH-DSCH TxOpps in a collision-free manner (with the aid of MEA/TMEA-S/TMEA-D). Initializing an 802.16(d) mesh CDS-mode network using SSBA encounters four problems: (1) how a new node can receive network-management messages transmitted by neighboring functional nodes to synchronize its clock and MAC-layer framing with those of the network; (2) when a new node can transmit its network-joining messages so that its chosen sponsoring operational node can receive them; (3) how an operational node can receive a network-joining message transmitted from a neighboring new node; and (4) when an operational node can transmit its network-management messages so that neighboring new nodes can have chances to receive them. We answer these questions and briefly explain our refined network entry process for a new node using an SSBA below.

First, each operational node is required to periodically transmit its network-management messages to each of its TDs regardless of the inexistence of operational nodes in those TDs. That is, the MEAs of all the TDs of an operational node should be activated at all time after the node has joined the network. With this design, a new node can have a chance to receive network-management messages of its neighboring operational nodes. For a new node, it should first scan all of its TDs to detect whether any operational node exists. This scanning procedure is described here. When a new node boots, it should iteratively point its antenna to each of its TD to try receiving any incoming messages. When pointing its antenna to a TD, the new node should continuously listen to this TD for 2^8 NCFG TxOpps, unless it receives network-management messages

sent from operational nodes in the TD. Because each operational node is allowed to use holdoff times ranged from 2^0 to 2^7 , continuously listening to a TD for 2^8 NCFG TxOpps is sufficient for a new node to receive an operational node's network-management messages, if any operational node exists in that TD. If the new node does not receive any messages in the TD, after listening for 2^8 NCFG TxOpps, the new node should point its antenna to the next TD to continue its scanning process. A new node should repeat the above process until it has scanned all of its TDs and detected the existence of neighboring operational nodes.

According to [1], a new node cannot transmit its own network-joining message until receiving the network-management messages sent from the same node twice. Thus, after detecting the existence of an operational node, the new node can stay in the TD where the detected operational node resides to wait for its next network-management message. If multiple operational nodes in different TDs are detected, the new node can first wait for the messages transmitted by the operational node with the highest SINR value.

After receiving network-management messages from the same node twice, the new node can choose one of its neighboring operational node as its sponsoring node and start its network entry process. In the network entry process, the sponsoring node is responsible for allocating a temporary bidirectional minislot allocation to relay network-joining control messages transmitted by the base station (BS) node and the new node. This temporary minislot allocation is required because a new node has not been operational and thus cannot negotiate a minislot allocation with any operational node.

In the network entry process, the new node is required to transmit several network-joining and registration control messages to the BS node. These control messages should be relayed via its chosen sponsoring node and be first sent to the sponsoring node using MSH-NENT messages on NENT TxOpps. To achieve this goal, two problems need to be solved. One problem is when a new node can transmit its MSH-NENT messages on certain NENT TxOpps so that its chosen sponsoring node can receive them. The other problem is how an operational node can receive such network-joining control messages from an unknown (new) node on NENT TxOpps. We first answer the second problem. An operational node is required to iteratively point its antenna to each of its TDs on NENT TxOpps. That is, on each NENT TxOpp, it points its antenna to a different TD. For example, a k -TD node has to point the antenna to its TD $(i \bmod k)$ on NENT TxOpp i .

The solution for the first problem is explained here. For a new node, it first divides NENT TxOpps into several k -TxOpp groups. After choosing its sponsoring node, the new node should point its antenna to the TD where the sponsoring node resides on all NENT TxOpps. It then transmits its MSH-NENT messages that carry its network-joining control messages in a probabilistic manner. When the NENT TxOpp number advances to the boundary of a k -TxOpp group, a new node randomly chooses a fraction N_f between 0 and 1. If N_f is below a pre-determined value N_{th} , the new node is allowed to transmit its network-joining messages during this TxOpp group. In this

condition, it should continuously transmit its MSH-NENT messages on each of the k TxOpps in this TxOpp group. If N_f is larger than or equal to N_{th} , the new node should keep silent during this TxOpp group. Because an operational node sequentially switches its antenna to each of its k TDs on k TxOpps and a new node stays at one of its TD for k TxOpps, their antennas can meet each other on a certain TxOpp.

If the chosen sponsoring node receives such a network-joining message from a new node on NENT TxOpps, it should first add the ID of this new node into the eligible node list of each of its TD and considers the control message schedulings of this new node for MSH-NCFG and MSH-DSCH messages are in unknown statuses. That is, it should assume that the new node will contend for every MSH-NCFG and MSH-DSCH TxOpp; thus, its MEAs should always take the contention of this new node for NCFG and DSCH TxOpps into consideration. In addition, the sponsoring node should disseminate the unknown schedule information of this new node to its neighboring nodes using its MSH-NCFG and MSH-DSCH messages, such that its neighboring nodes can also take the contention of this new node in consideration. The reason behind this design is that, if the nodes in the new node's two-hop neighborhood do not consider its contention for NCFG and DSCH messages in advance, after the new node becomes operational, the one-hop neighboring nodes of the new node will not point their antenna to receive the MSH-NCFG and MSH-DSCH messages transmitted by the new node and the two-hop neighboring nodes of the new node will not properly point their antennas to avoid inter-node scheduling conflicts.

The new node can know whether its chosen sponsoring node has received and accepted its sponsorship request by checking whether the next MSH-NCFG message transmitted by the chosen sponsoring node contains the network-entry-open IE for acknowledging its sponsorship request. If the new node does not find the corresponding network-entry-open IE in the sponsoring node's next MSH-NCFG message, it should halve the value of N_{th} and repeat the above process until the chosen sponsoring node responds it with a network-entry-open IE. The reason why a new node should decrease the value of N_{th} is to reduce the contention for NENT TxOpps, if multiple new nodes exist and simultaneously choose the same operational node as their sponsoring nodes. The lower bound of N_{th} is set to 0.1 in our simulations, which is sufficient for all new nodes to join the simulated network. On the other hand, if the new node finds a network-entry-open IE for another new node in the next MSH-NCFG message of the chosen sponsoring node, it should cease its MSH-NENT message transmission to avoid disturbing the ongoing network entry process of another new node, until it finds that the chosen sponsoring node transmits an MSH-NCFG message containing a network-entry-ack IE for that new node, which indicates that new node has completed its network entry process.

If the new node detects a network-entry-open IE for itself in the chosen sponsoring node's MSH-NCFG message, it means that the sponsoring node has allocated a temporary bidirectional minislots allocation for the new node to relay its network-joining and registration control

messages. The new node should then transmit a network-entry-ack IE to its sponsoring node using its MSH-NENT message for acknowledging its sponsoring node (using the same way for transmitting its network-joining message). After this, it can transmit its capacity negotiation, authorization, and network registration control messages to the sponsoring node over the temporary minislots allocated by the sponsoring node for it. After receiving these control messages, the sponsoring node will in turn forward these messages towards the BS node in the network. Also, upon receiving control messages transmitted by the BS node and destined to the new node, the sponsoring node will forward them to the new node on the temporary minislots that it allocated for the new node.

After finishing exchanging necessary control messages with the BS node, the new node has joined the network and become operational. It should then transmit a network-entry-close IE to the sponsoring node using MSH-NENT messages on NENT TxOpps. The network-entry-close IE is used to notify the sponsoring node that the network-entry process of the new node that it is sponsoring has been completed and the temporary minislots allocation reserved for the new node is no longer needed. After receiving this IE, the sponsoring node first acknowledges the new node by sending it an MSH-NCFG message containing a network-entry-ack IE on NCFG TxOpps and then cancels the temporary minislots allocation reserved by it for the new node.

Notice that, for a node that just joined the network, it is only allowed to transmit its MSH-NCFG/MSH-DSCH messages in the TD where its chosen sponsoring node resides. The reason is that at this stage it is possible that only the nodes in the two-hop neighborhood of its chosen sponsoring node know the existence of this new operational node. If this new operational node transmits its control messages in other TDs, nodes in other TDs may not know on which TxOpps they should point their antennas toward the TDs where this new operational node is. Worse yet, they may schedule their own control message transmissions on the same TxOpps used by the new operational node, generating inter-node scheduling conflicts.

To solve this problem, we require a new operational node not to transmit its own control messages to TDs where nodes have not learned its existence. The new operational node should continuously transmit MSH-NENT messages (without containing any network-joining messages) to each of its TDs (excluding the TD where its sponsoring node is) within each of the k -NENT-TxOpp groups in the probabilistic manner described above. Only when it finds the MSH-NCFG/MSH-DSCH messages transmitted from the same node in a TD contain its next NCFG/DSCH TxOpp information (indicating that this new operational node is in the unknown scheduling status and considered to contend for every NCFG/DSCH TxOpp) **twice**, can it stop transmitting its MSH-NENT message in this TD and start scheduling its next NCFG/DSCH TxOpp in this TD. This is because now the new node can ensure that nodes in this TD know its existence and have notified their one-hop neighboring nodes of its existence. Thus, the transmissions of its MSH-NCFG/MSH-DSCH messages in this TD will not lead to any inter-node scheduling conflicts. If the new node

Table 2
Frequently-used acronyms.

Acronym	Meaning
ATUF	Average Throughput of a UDP Flow in a network
ATTF	Average Throughput of a TCP Flow in a network
ATHPT	Average Three-way Handshake Procedure Time
ATOUN	Average Transmission Opportunity Utilization of Nodes
BS	Base Station
CDS	Coordinated Distributed Scheduling
CS	Centralized Scheduling
CV-ATUF	The Coefficients of Variation of all nodes' ATUF values
CV-ATTF	The Coefficients of Variation of all nodes' ATTF values
IE	Information Element
MEA	Mesh Election Algorithm
MEAI	Mesh Election Algorithm Instance
TMEA-S	Transmission-domain-aware Mesh Election Algorithm with a Static holdoff time design
TMEA-D	Transmission-domain-aware Mesh Election Algorithm with a Dynamic holdoff time design
MSH-DSCH	Mesh Distributed Scheduling
MSH-NCFG	Mesh Network Configuration
MSH-NENT	Mesh Network Entry
MTD	Multi-transmission-domain
SBA	Switched-Beam Antenna
SSBA	Single-Switched-Beam Antenna
STD	Single-transmission-domain
TD	Transmission Domain
TDI	Transmission Domain Index
THP	Three-way Handshake Procedure
TMEA	Transmission-domain-aware Mesh Election Algorithm
TMSA	Transmission-domain-aware Minislot Scheduling Algorithm
TxOpp	Transmission Opportunity

does not receive any MSH-NCFG/MSH-DSCH messages in a TD after k^2 NENT TxOpps, it means that no operational node is present in this TD. Thus, the new node can start scheduling its MSH-NCFG/MSH-DSCH message transmissions in this TD. Such control message transmissions are still needed because there may be several new nodes in this TD.

A.2. Frequently-used acronyms

Many acronyms are used in this paper to make this paper concise. To help the reader easily refer to the meanings of these acronyms, frequently-used acronyms are collected and defined in Table 2.

References

- [1] IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001). IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.
- [2] S.Y. Wang, C.C. Lin, H.W. Chu, T.W. Hsu, K.H. Fang, Improving the performances of distributed coordinated scheduling in IEEE 802.16 mesh networks, *IEEE Transactions on Vehicular Technology* 57 (4) (2008) 2531–2547.
- [3] Zhi-Xiong Liu, Fan Yang, Zheng-Bao Liu, Yu-Liang Tang, A dynamic holdoff scheme in IEEE 802.16 distributed scheduling mesh networks, in: 2010 5th International Conference on Computer Science and Education (ICCSE), August 24–27, 2010, pp. 937–940.
- [4] S. Chakraborty, D.K. Sanyal, A. Chakraborty, A. Ghosh, S. Chattopadhyay, M. Chattopadhyay, Tuning holdoff exponents for performance optimization in IEEE 802.16 mesh distributed coordinated scheduler, in: 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), vol. 1, February 26–28, 2010, pp. 256–260.
- [5] Min Cao, Wenchao Ma, Qian Zhang, Xiaodong Wang, Wenwu Zhu, Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc 2005), vol. 1, May 25–27, 2005, pp. 165–168.
- [6] Nico Bayer, Dmitry Sivchenko, Bangnan Xu, Veselin Rakocevic, Joachim Habermann, Transmission timing of signaling messages in IEEE 802.16 based mesh networks, in: European Wireless 2006, vol. 1, April 2006, pp. 165–168.
- [7] K.S. Vijayalayan, A. Harwood, S. Karunasekera, Fast channel establishment for IEEE 802.16 wireless mesh networks, in: 2010 IEEE Global Telecommunications Conference (GLOBECOM 2010), December 6–10, 2010, pp. 1–6.
- [8] F. Babich, M. Comisso, L. Mania, Multi-antenna techniques for wireless mesh networks in an outdoor environment, in: 2007 IEEE International Conference on Communications (ICC), June 24–28, 2007, pp. 4961–4966.
- [9] K. Sundaresan, R. Sivakumar, A unified mac layer framework for ad-hoc networks with smart antennas, *IEEE/ACM Transactions on Networking* 15 (3) (2007) 546–559.
- [10] Ashish Deopura, Aura Ganz, Provisioning link layer proportional service differentiation in wireless networks with smart antennas, *Wireless Networks* 13 (3) (2007) 371–378.
- [11] Karthikeyan Sundaresan, Weizhao Wang, Stephan Eidenbenz, Algorithmic aspects of communication in ad-hoc networks with smart antennas, in: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006), vol. 1, May 22–25, 2006, pp. 298–309.
- [12] Harkirat Singh, Suresh Singh, Smart-aloHa for multi-hop wireless networks, *Mobile Networks and Applications* 10 (5) (2005) 651–662.
- [13] Lichun Bao, J.J. Garcia-Luna-Aceves, Transmission scheduling in ad hoc networks with directional antennas, in: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, 2002, pp. 48–58.
- [14] Lichun Bao, J.J. Garcia-Luna-Aceves, Receiver-oriented multiple access in ad hoc networks with directional antennas, *Wireless Networks* 11 (1–2) (2005) 67–79.
- [15] J.B. Cain, T. Billhartz, L. Foore, E. Althouse, J. Schlorff, A link scheduling and ad hoc networking approach using directional antennas, in: Military Communications Conference, 2003, MILCOM 2003, IEEE, vol. 1, October 13–16, 2003, pp. 643–648.
- [16] M.E. Steenstrup, Exploiting directional antennas to provide quality of service and multipoint delivery in mobile wireless networks, in: Military Communications Conference, 2003, MILCOM 2003, IEEE, vol. 2, October 13–16, 2003, pp. 987–992.
- [17] Zhensheng Zhang, Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas. In 2005 IEEE International Conference on Communications, 2005 (ICC 2005), vol. 5, May 16–20, 2005.
- [18] Zhensheng Zhang, Bo Ryu, G. Nallamothu, Zhuochuan Huang, Performance of all-directional transmission and reception algorithms in wireless ad hoc networks with directional antennas, in: IEEE Military Communications Conference, 2005 (MILCOM 2005), vol. 1, October 17–20, 2005, pp. 225–230.
- [19] Zhensheng Zhang, Dtra: directional transmission and reception algorithms in wlan with directional antennas for qos support, *IEEE Network* 19 (3) (2005) 27–32.
- [20] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, C.C. Lin, The design and implementation of the NCTUns 1.0 network simulator, *Computer Networks* 42 (2) (2003) 175–197. <<http://NSL.csie.nctu.edu.tw/nctuns.html>> .
- [21] Nico Bayer, Bangnan Xu, Joachim Habermann, Veselin Rakocevic, Improving the performance of the distributed scheduler in IEEE 802.16 mesh networks, in: IEEE VTC Spring 2007, Dublin, April 2007.
- [22] Bong Chan Kim, Dong Gu Kwak, Heecheol Song, Hwang Soo Lee, Joong Soo Ma, An adaptive holdoff algorithm based on node state for IEEE 802.16 mesh mode with coordinated distributed scheduling, in: IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008 (PIMRC 2008), vol. 1, September 15–18, 2008, pp. 1–5.
- [23] V. Loscri, G. Aloï, Transmission hold-off time mitigation for IEEE 802.16 mesh networks: a dynamic approach, in: Wireless Telecommunications Symposium, 2008 (WTS 2008), vol. 2, April 24–26, 2008, pp. 31–37.

- [24] V. Loscri, A queue based dynamic approach for the coordinated distributed scheduler of the ieee 802.16, in: IEEE Symposium on Computers and Communications, 2008 (ISCC 2008), vol. 2, July 6–9, 2008, pp. 423–428.
- [25] Yun Li, Dengyu Wei, Hongcheng Zhuang, Haibao Wang, Ping Wang, A new congestion control method for ieee 802.16 mesh mode, in: International Conference on Communication Software and Networks, 2009 (ICCSN '09), vol. 1, February 2009, pp. 726–730.
- [26] Sudeept Bhatnagar, Samrat Ganguly, Rauf Izmailov, Design of IEEE 802.16-based multi-hop wireless backhaul networks, in: AccessNets '06: Proceedings of the 1st international conference on Access networks, ACM Press, New York, NY, USA, 2006, p. 5.
- [27] Qing Xiong, Weijia Jia, Chanle Wu, Gang Ye, Throughput enhancement with bidirectional concurrent transmission in ieee 802.16 mesh networks, in: Second International Conference on Communications and Networking in China, 2007 (CHINACOM '07), vol. 1, August 22–24, 2007, pp. 947–951.
- [28] Constantine A. Balanis, Antenna Theory Analysis and Design, second ed., Wiley, New York, 1997 (Chapter 6).
- [29] G. Durgin, T.S. Rappaport, Basic relationship between multipath angular spread and narrowband fading in wireless channels, Electronics Letters 34 (25) (1998) 2431–2432.



Chih-Che Lin currently works for the Industrial Technology Research Institute of Taiwan. He received his BS degree and Ph.D. degree in computer science from National Chiao Tung University, Taiwan, in 2002 and 2010, respectively. He was a core team member of the NCTUns network simulator project during 2002 and 2010. His current research interests include wireless networking, wireless mesh networks, vehicular networks, intelligent transportation systems, and network simulation.



S.Y. Wang is a Professor of the Department of Computer Science at National Chiao Tung University, Taiwan. He received his Master and Ph.D. degree in computer science from Harvard University in 1997 and 1999, respectively. Before that, he received his Master degree in computer science from National Taiwan University in 1992 and his bachelor degree in computer science from National Taiwan Normal University in 1990. His research interests include wireless networks, Internet technologies, network simulations, and operating systems. He authors a network simulator, called NCTUns, which is now is a famous tool widely used by people all over the world.



Teng-Wei Hsu received his BS degree in electrical engineering from Department of electrical engineering, Chang Gung University, Taiwan, in 2005. He received his MS degree in network engineering from National Chiao Tung University, Taiwan, in 2007. He was a core team member of the NCTUns network simulator project during 2005 and 2007. His major research interests include wireless mesh networks and network simulation.