Performance Evaluation for Minislot Allocation for Wireless Mesh Networks

Mohsen Guizani, *Senior Member, IEEE*, Phone Lin, *Senior Member, IEEE*, Shin-Ming Cheng, *Member, IEEE*, Di-Wei Huang, and Huai-Lei Fu

***Abstract*—The IEEE 802.16 Standard defines the mesh mode for media access control for external and internal packet transmission. In the IEEE 802.16 mesh mode, the allocation of minislots is handled by centralized scheduling and distributed scheduling, which are independently exercised. This paper proposes the Combined Distributed and Centralized (CDC) and Combined Distributed and Centralized with Queue capability (CDCQ) schemes to combine distributed scheduling and centralized scheduling so that the minislot allocation can be more flexible and the utilization is increased. Two scheduling algorithms, i.e., the Round-Robin (RR) and Greedy algorithms, are used as the baseline algorithms for centralized scheduling. This paper proposes an analytical model and conducts simulation experiments to investigate the performance of the CDC-series schemes with the RR and Greedy algorithms in terms of the acceptance rate of both external and internal packet data. Our study indicates that the CDC-series schemes outperform the scheme proposed in the IEEE standard.**

***Index Terms*—IEEE 802.16, minislot allocation, scheduling, wireless mesh network (WMN).**

NOMENCLATURE

α	Maximum percentage of minislots that can be allocated for external packets in a data subframe for the Partition scheme.
δ_A	Delay bound for the internal packets.
δ_I	Delay bound for the external packets.
λ_A	Internal packet arrival rate to an SS.
λ_I	External packet arrival rate to an SS.
$1/\mu_A$	Expected internal packet transmission time.
$1/\mu_I$	Expected external packet transmission time.
K	Number of layers in the centralized scheduling tree.
M	Total number of minislots in a data subframe.
m_A	Number of minislots an internal packet requests for transmission.
m_I	Number of minislots an external packet requests for transmission.
$P_{A,d}$	Dropping probability for the internal packet.
$P_{I,d}$	Dropping probability for the external packet.
t_s	Length of a scheduling period.
v_{λ_A}	Variance of Gamma distributed internal packet interarrival times.
v_{λ_I}	Variance of Gamma distributed external packet interarrival times.
v_{μ_A}	Variance of Gamma distributed internal packet transmission times.
v_{μ_I}	Variance of Gamma distributed external packet transmission times.

Manuscript received May 17, 2007; revised October 27, 2007, November 17, 2007, December 16, 2007, and January 16, 2008. First published February 15, 2008; current version published November 12, 2008. The work of P. Lin was supported in part by the National Science Council (NSC), R.O.C., under Contract NSC-96-2627-E-002-001, Contract NSC-96-2811-E-002-010, Contract NSC-96-2628-E-002-002-MY2, and Contract NSC-95-2221-E-002-091-MY3, Ministry of Economic Affairs (MOEA), R.O.C., under Contract 93-EC-17-A-05-S1-0017, the Telcordia Applied Research Center, Taiwan Network Information Center (TWNIC), the Excellent Research Projects of National Taiwan University under 95R0062-AE00-07, and the Chunghwa telecom M-Taiwan program M-Taoyuan Project. The review of this paper was coordinated by Prof. B. Mark.

M. Guizani is with the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008-5201 USA (e-mail: mguizani@cs.wmich.edu).

P. Lin, D.-W. Huang, and H.-L. Fu are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: plin@csie.ntu.edu.tw; dwhuang@pcs.csie.ntu.edu.tw; vicfu@pcs.csie.ntu.edu.tw).

S.-M. Cheng is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: smcheng@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TVT.2008.918712

I. INTRODUCTION

THE IEEE 802.16 Standard [1] (also known as WiMax) defines air interface, including the physical and media access control (MAC) layers, for wireless metropolitan area networks. It provides fixed broadband wireless access with the same level of quality of service as the traditional cabled access network, e.g., fiber optical links, coaxial links, and digital subscriber line links. The data transmission rate is up to 70 Mb/s. An IEEE 802.16 network consists of base stations (BSs) and subscriber stations (SSs). The BS serves as a gateway between the IEEE 802.16 network and the external network. The SS acts like a client-side terminal through which mobile users can access the network. The IEEE 802.16 network operates at the 10–66 GHz or the 2–11 GHz band [1], [2]. In the 10–66 GHz band, the signal propagation between a BS and an SS is through line of sight (LOS). The 2–11 GHz band supports non-LOS communication. The IEEE 802.16 Standard specifies two modes for sharing the wireless medium: 1) the point-to-multipoint (PMP) mode and 2) the mesh mode. The PMP mode is designed for wireless communication between the BS and the SSs, and the mesh mode enables communication among SSs [1], [3]. In this paper, we concentrate on studying the mesh mode. To simplify our description, we refer to the IEEE 802.16 Standard, with the mesh mode as the wireless mesh network (WMN).

Fig. 1(a) shows the WMN network architecture. The mesh BS connects to the external network with backhaul links. Any SS pair without direct wireless links can relay the data to

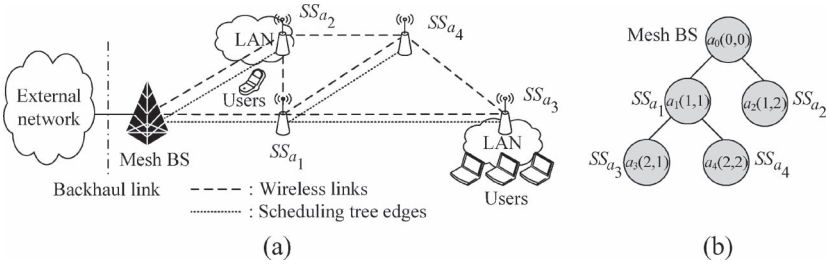

Fig. 1. IEEE 802.16 mesh network architecture. (a) Network topology. (b) Scheduling tree with index numbers.

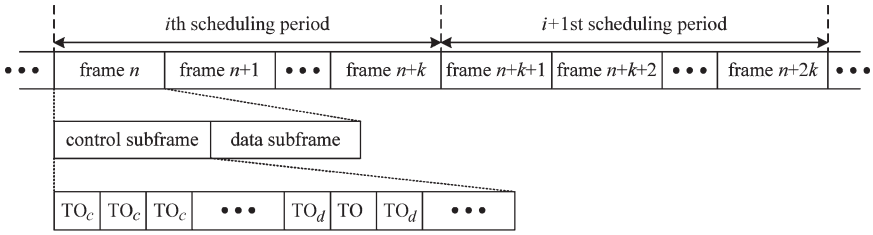

Fig. 2. Frame structure in the IEEE 802.16 mesh network.

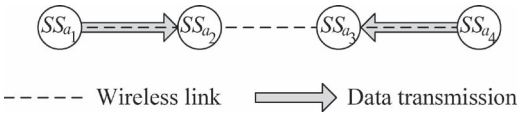

Fig. 3. Example for the minislot reuse property.

each other through other SSs. The users connect to the SS by adopting various existing local area network technologies, e.g., IEEE 802.3 (Ethernet) [4] or IEEE 802.11 (WiFi) [5]. In this paper, we focus on the MAC protocol between the SSs and between the SS and the mesh BS. The WMN adopts the time-division multiplexing (TDM) radio access technology between SSs and between the SS and the mesh BS, where a radio channel is divided into physical slots (PSs) using time sharing, and multiple PSs are grouped as a frame. As shown in Fig. 2, a frame is divided into a control subframe and a data subframe. The control subframe consists of transmission opportunities (TOs) used to carry signaling messages for centralized scheduling and distributed scheduling (to be discussed later), which are denoted as TO_c and TO_d, respectively. The numbers of TO_cs and TO_ds in a control subframe are configured by the operator. The data subframe carries the user data, which is further divided into at most 256 minislots [1]. The transmission rate r (in bits per second) of a minislot depends on several factors (e.g., channel coding, modulation, and frequency band). More details about the discussion of these factors are available in [1]. A minislot can be reused by multiple SSs (i.e., multiple SSs may transmit packets in the same minislot), as long as the SSs are geographically separated [6], [7] (i.e., they do not interfere with each other). For example, in Fig. 3, SS_{a_1} and SS_{a_4} can transmit data to SS_{a_2} and SS_{a_3} in the same minislot, respectively. This property is called "minislot reuse" in TDM.

The IEEE 802.16 Standard proposes centralized scheduling and distributed scheduling for minislot allocation. In centralized scheduling, the mesh BS works like a cluster head that maintains the topology of WMN, receives the minislot allocation request from each SS, and determines transmission and reception minislots for each SS. Thus, all the control and data packets need to go through the mesh BS. Distributed scheduling is exercised in two neighboring SSs to reserve minislots for data transmission between them. The data traffic in the WMN can be divided into two categories: 1) external traffic (routed between the SS and the application server out of WMN) and 2) internal traffic (routed between two SSs in the same WMN). A data subframe can simultaneously carry the internal traffic packets and the external traffic packets. Since the external traffic should be routed in WMN and then delivered to the application server through the mesh BS, and the internal traffic only occurs between two SSs within the WMN, it is desirable to adopt centralized scheduling and distributed scheduling to allocate minislots for the external and internal traffic, respectively.

The IEEE 802.16 Standard suggests that the data subframe be partitioned into two parts. In this standard [1], the MSH_CSCH_DATA_FRACTION parameter is defined to store the proportion of minislots allocated to centralized scheduling and distributed scheduling, which is carried in the mesh network configuration (MSH-NCFG) message. This implies that the minislots in a data subframe are separately allocated to centralized scheduling and distributed scheduling. In [1], it is mentioned that centralized scheduling and distributed scheduling can be executed in the same data subframe.

This implies that the data subframe is partitioned into two parts to serve the external and internal traffic, respectively, and is known as the "Partition" scheme. However, this may not be the best solution, because the partition boundary may not precisely capture the traffic pattern and the minislots may not be fully utilized. The two main objectives of this paper are given as follows:

1) developing the Combined Distributed and Centralized (CDC) and Combined Distributed and Centralized with Queue capability (CDCQ) schemes to break the partition

boundary so that the minislot allocation can be more flexible;
2) proposing an analytical model and conduct simulation experiments to investigate the performance for the Partition, CDC, and CDCQ schemes.

A. Literature Review

The related literature for the minislot allocation of WMN is summarized as follows: In [8], Wei *et al.* targeted designing a greedy-like algorithm and a routing path establishment algorithm for only centralized scheduling. In [9], Kim and Ganz proposed an algorithm based on the weights of SSs to fairly allocate minislots for SSs to deliver the external traffic (i.e., for only centralized scheduling). In this paper, they investigated the fairness for minislot allocation. In [10], Chen *et al.* considered only centralized scheduling, where they proposed an odd–even alternation mechanism for minislot allocation. The three papers considered only centralized scheduling. They did not consider the interaction of centralized scheduling and distributed scheduling. Furthermore, there is no complete performance analysis in these studies. Therefore, the two ideas being investigated in this paper are CDC and CDCQ. The study in [11] focused on the design of the centralized scheduling algorithm. This is beyond the scope of this paper. In [12], Cao *et al.* discussed the performance evaluation for distributed scheduling. The interaction and combination of the distributed scheduling and the centralized scheduling are not considered in [12]. In this paper, we consider and provide a more complete performance evaluation for minislot allocation in WMNs.

The rest of this paper is organized as follows: Section II first illustrates distributed scheduling and centralized scheduling, and then presents the Partition scheme in the IEEE 802.16 Standard and the proposed CDC schemes. An analytical model, simulation experiments, and performance evaluation are given in Section III. Finally, Section IV concludes our study.

II. Partition and CDC Schemes

Let SS_{a_i} denote an SS with node ID a_i. Assume that the total number of minislots in a data subframe is M. The variable s_j indicates the status of the jth minislot in a data subframe. $s_j = \mathtt{F}$ indicates that the minislot is fresh, $s_j = \mathtt{TX}$ indicates that the minislot is used to transmit data, and $s_j = \mathtt{RX}$ indicates that the minislot is used to receive data. SS_{a_i} maintains its own array $\mathbf{S}_{a_i} = \{s_1, s_2, s_3, \ldots, s_M\}$ to store the status of each minislot, i.e., how it functions in that minislot.

Upon receipt of an internal packet request (requesting transmission rate R_{A,a_u} to be transmitted to SS_{a_u}'s neighbor SS_{a_v}), SS_{a_u} exercises a three-way handshaking procedure with SS_{a_v} to select $\lceil R_{A,a_u}/r \rceil$ collision-free[1] minislots to transmit the packet. The three-way handshaking procedure is known as "distributed scheduling," the details of which can be found in our previous work [13]. Note that distributed scheduling introduces a delay for the transmission of packets due to the coordination among SSs.

For data communication between the mesh BS and the SSs, the IEEE 802.16 Standard proposes centralized scheduling, where the mesh BS acts as a scheduler and determines transmission and reception minislots for each SS. A scheduling tree rooted at the mesh BS (i.e., the routing path between each SS and the mesh BS) is established. Upon receipt of an external packet request (requesting transmission rate R_{I,a_i} to be transmitted to the mesh BS), SS_{a_i} and the mesh BS exercise the centralized scheduling procedure to reserve transmission rate R_{I,a_i} in all SSs along the routing path from SS_{a_i} to the mesh BS.

The centralized scheduling consists of three stages. In the first stage, each SS buffers all external packet requests. In the second stage, each SS forwards buffered requests to the mesh BS. The mesh BS executes the scheduling algorithms (to be discussed later) to check the status arrays $\mathbf{S}$'s of all SSs along the routing path from SS_{a_i} to the mesh BS and tries to reserve enough collision-free minislots in all SSs along the routing path, which is known as the "scheduling result." Take Fig. 1 as an example. Each SS (along the routing path from the SS_{a_4} to the mesh BS, i.e., SS_{a_4} and SS_{a_1}) requires $\lceil R_{I,a_4}/r \rceil$ collision-free minislots to serve the external packet request. In the third stage, the mesh BS distributes the scheduling result to all SSs, and the SSs reserve minislots according to the result. The time period for exercising the three stages is defined as a *scheduling period* (denoted as t_s). The minislots of all SSs are re-reserved by centralized scheduling at the beginning of every scheduling period. The scheduling result in the ith scheduling period is referenced in the $(i+1)$th scheduling period, as shown in Fig. 2.

In the centralized scheduling, the minislot utilization is highly dependent on the scheduling algorithm. To date, some studies [11], [14], [15] have touched on this issue. It has been shown that the scheduling algorithm is an *NP-hard* problem, i.e., we may not find the optimal solution for scheduling when the number of SSs in a WMN is sufficiently large. In [11], it has been proven that, under certain WMN network conditions, the centralized scheduling is no longer an *NP-complete* problem. In this paper, we use the two general algorithms Round-Robin (RR) and Greedy as the baseline centralized scheduling algorithms (that have widely been adopted to solve many scheduling problems and are easily deployable in real systems) to investigate the performance of the centralized scheduling. The two algorithms will be described in Appendix I.

As suggested by the IEEE 802.16 Standard [1], the minislots in a data subframe are partitioned for both distributed scheduling and centralized scheduling. In this paper, we refer to this operation as the Partition scheme. The possible implementation for the Partition scheme is elaborated.

Partition Scheme. The minislots in a data subframe are partitioned into two parts: one for the centralized scheduling and the other for the distributed scheduling. The centralized scheduling and distributed scheduling are independently executed. Each SS maintains a first-in–first-out (FIFO) queue `reqlist` to buffer the external packet requests arriving at the SS during the scheduling period. The Partition scheme consists of six

[1]The kth minislot in the data subframe is called "collision free" if it satisfies the following two properties: 1) $s_k \neq \mathtt{RX}$ for all neighbors of SS_{a_u} and 2) $s_k \neq \mathtt{TX}$ for all neighbors of SS_{a_v}.

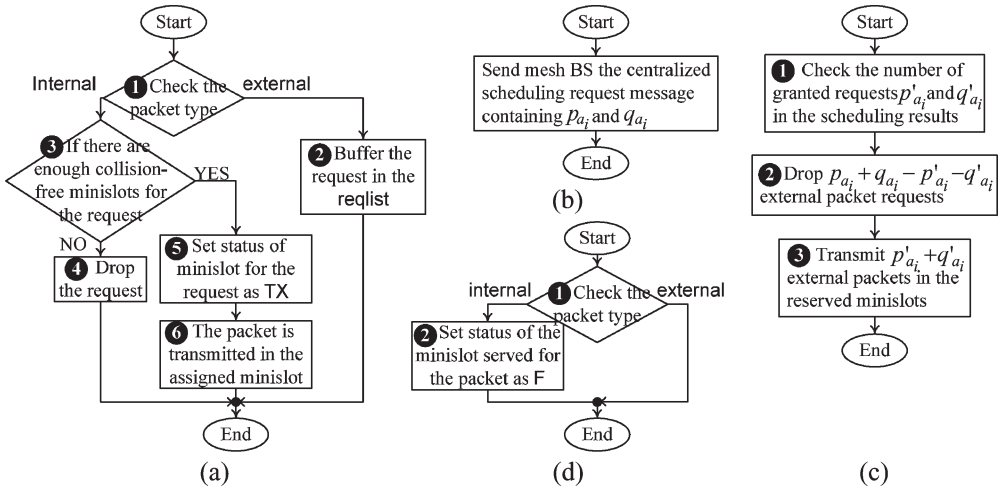

Fig. 4. Flow chart for the Partition scheme. (a) Arrival. (b) Request. (c) Transmission. (d) Departure.

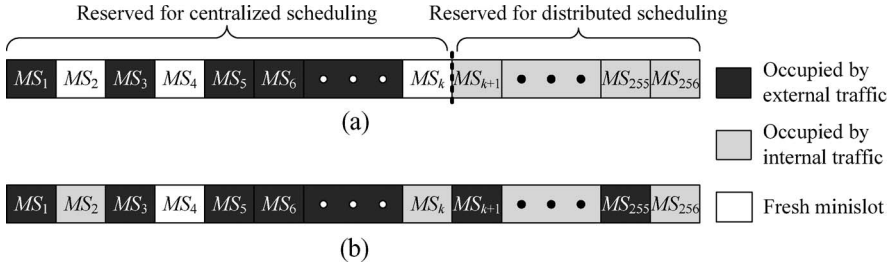

Fig. 5. Examples for minislot allocation for (a) Partition and (b) CDC schemes.

procedures: Initial, Arrival, Request, Schedule, Transmission, and Departure.

1) *Initial*. This procedure initiates the system parameters. The mesh BS broadcasts to all SSs the number M_c of minislots in a data subframe that can be allocated for centralized scheduling.
2) *Arrival*. See Fig. 4(a). This procedure is exercised when a new packet request arrives during the scheduling period. The SS checks the type of request [see (1) in Fig. 4(a)]. If it is an external packet request, the SS buffers this request in `reqlist` [see (2) in Fig. 4(a)]. If it is an internal packet request, the SS immediately executes the distributed scheduling to reserve minislots for this request [see (3)–(5) in Fig. 4(a)]. Then, the internal packet can be transmitted in the reserved minislots [see (6) in Fig. 4(a)].
3) *Request*. See Fig. 4(b). At the end of the scheduling period, each SS executes the Request procedure to reserve minislots for external packet transmission during the next scheduling period. Suppose that, at the end of the scheduling period, there are p_{a_i} external packet requests buffered in the `reqlist` and q_{a_i} external packets being served in SS_{a_i} during the current scheduling period. SS_{a_i} sends the centralized scheduling request message carrying the number $p_{a_i}+q_{a_i}$ to the mesh BS.
4) *Schedule*. After the mesh BS receives the centralized scheduling requests from all SSs, it executes the Scheduling procedure. The mesh BS releases all minislots for centralized scheduling by setting their status to `F`. Then, it executes the centralized scheduling algorithm (see Appendix I) to reserve minislots for the $p_{a_i}+q_{a_i}$ external packet requests. The mesh BS sends the scheduling result to all SSs.
5) *Transmission*. See Fig. 4(c). When an SS SS_{a_i} receives the scheduling result from the mesh BS, SS_{a_i} executes this procedure to transmit external packets in the assigned minislots. Suppose that $p'_{a_i}+q'_{a_i}$ external packet requests are granted to be served [see (1) in Fig. 4(c)]. Then, SS_{a_i} drops $p_{a_i}+q_{a_i}-p'_{a_i}-q'_{a_i}$ external packets [see (2) in Fig. 4(c)] and transmits $p'_{a_i}+q'_{a_i}$ external packets in the reserved minislots [see (3) in Fig. 4(c)].
6) *Departure*. See Fig. 4(d). This procedure is invoked in each SS when a packet transmission completes. If the type of the packet is an internal packet, the SS releases the minislots serving this packet by setting their status to `F` [see (2) in Fig. 4(d)]. If it is an external packet, the procedure does nothing, i.e., the status of the minislot is unchanged during the current scheduling period.

Fig. 5(a) shows an example of the scheduling result of the Partition scheme, where the minislots in a data subframe are labeled as $MS_1, MS_2, \ldots, MS_{256}$.

In the Partition scheme, once the WMN is configured, the minislots cannot be flexibly reserved for the centralized scheduling and distributed scheduling until the next configuration. We propose the CDC scheme to release the aforementioned limitation, whose details are given here.

CDC Scheme. In this scheme, the partition boundary is removed, and both distributed scheduling and centralized scheduling can allocate all minislots in a data subframe. We

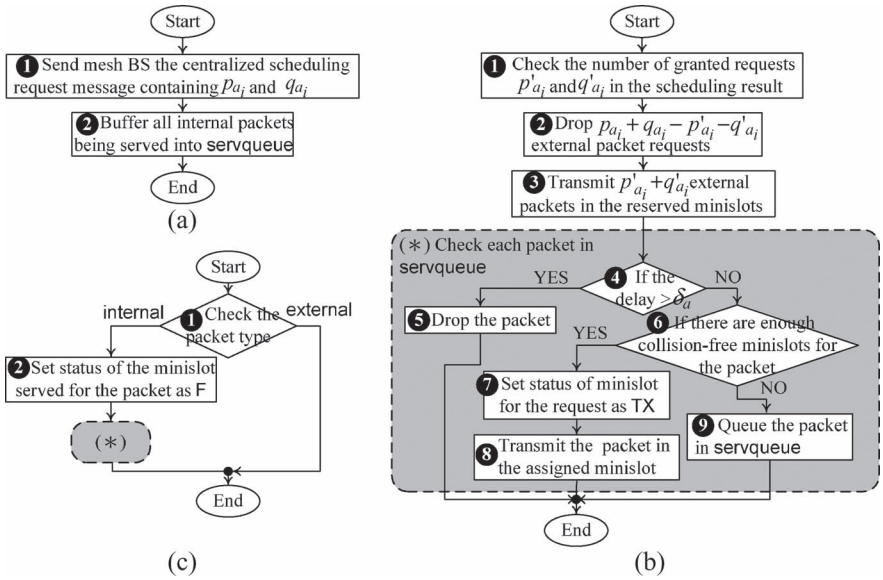

Fig. 6. Flow chart for the CDC scheme. (a) Request. (b) Transmission. (c) Departure.

slightly modify the procedures of the Partition scheme. Due to the delay of the process for the three-way handshake, the transmission for the internal packet request may not be completed within a scheduling period. Thus, each SS maintains a FIFO queue `servqueue` to buffer the internal packets, which are not completely transmitted in a scheduling period. The queued external packets can be served in the next scheduling period. If the time period when internal packet requests wait in the `servqueue` is longer than a delay bound δ_A, these packets are dropped. The four procedures are modified as described here.

1) *Arrival*. The procedure is similar to that of the Partition scheme. For an external packet request, the SS buffers this request in `reqlist`. For an internal packet request, the SS immediately executes the distributed scheduling to reserve minislots for this request. The SS can allocate the total minislots to internal packet requests without boundary.
2) *Request*. See Fig. 6(a). At the end of the scheduling period, SS_{a_i} sends the centralized scheduling request message containing the number $p_{a_i}+q_{a_i}$ to the mesh BS [see (1) in Fig. 6(a)]. Then, SS_{a_i} queues all the internal packets currently being served into `servqueue` [see (2) in Fig. 6(a)].
3) *Schedule*. The mesh BS schedules the total M minislots in a data subframe to serve external packet requests by exercising the centralized scheduling algorithm (to be elaborated upon in Appendix I).
4) *Transmission*. See Fig. 6(b). This procedure is similar to that of the Partition scheme. The difference is described as follows: Each SS transmits the external packet first and then serves the internal packets in `servqueue` by executing distributed scheduling [see (4)–(9) in Fig. 6(b)].
5) *Departure*: See Fig. 6(c). When an internal packet transmission is complete [see (1) and (2) in Fig. 6(c)], the SS takes the same actions in the Transmission procedure.

Note that, in the CDC scheme, the minislots are first allocated to the external packets by centralized scheduling at the beginning of every scheduling period. During the scheduling period, if internal packet requests arrive, the left minislots are allocated to these requests by distributed scheduling. Fig. 5(b) shows an example for minislot allocation with the CDC scheme.

CDC takes higher priority to serve external packets than internal packets. We propose an enhanced scheme CDCQ to increase the acceptance of the internal packets. In CDCQ, two FIFO queues `IQ` and `AQ` are maintained in each SS to buffer the dropped external packet requests and the dropped internal packet requests, respectively. The dropped requests can temporarily be buffered in `IQ` and `AQ` and may be served in the following scheduling periods. If the external (internal) packet requests in `IQ`(`AQ`) wait longer than a delay bound $\delta_I(\delta_A)$ for the external (internal) packets, the requests are dropped.

Note that the IEEE 802.16 Standard defines the two messages mesh centralized scheduling (MSH-CSCH) and mesh centralized schedule configuration (MSH-CSCF) to broadcast scheduling results to all SSs [1]. In the implementation for the CDC and CDCQ schemes, we may reutilize these two messages. Therefore, we do not introduce any new message types (or overhead) to the WMN.

In the rest of paper, to simplify our description, we use Partition_RR, Partition_Greedy, CDC_RR, CDC_Greedy, CDCQ_RR, and CDCQ_Greedy to denote Partition combined with RR and Greedy, CDC combined with RR and Greedy, and CDCQ with RR and Greedy, respectively.

III. Performance Evaluation

We conduct simulation experiments to investigate the performance of the Partition, CDC, and CDCQ schemes with the RR and Greedy algorithms. Our simulation model is based on the event-driven approach, which has widely been adopted to simulate the wireless network [16]–[20]. The notations used in the simulation experiments are listed in the Nomenclature list. Our study assumes that all SSs are identical. Following the IEEE 802.16 Standard [1], we model the WMN as a regular hexagonal topology, as shown in Fig. 7, where a four-layer

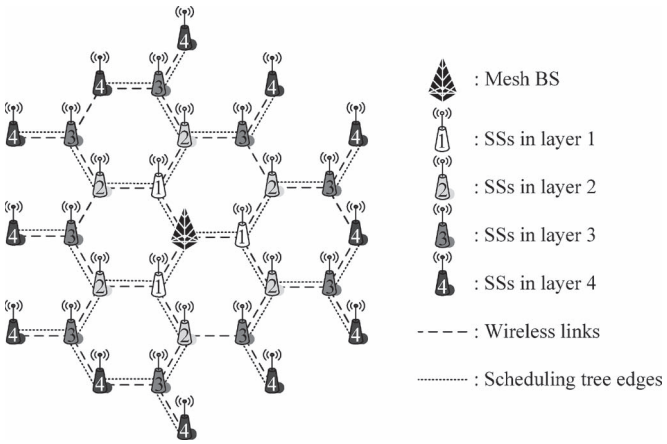

Fig. 7. Four-layer WMN layout structure.

WMN is considered. Comparing with the square-grid topology adopted in [21], it is more general to adopt a hexagonal topology to simulate the WMN.

Note that every IEEE 802.16 transmission includes two or three orthogonal frequency-division multiplexing (OFDM) symbols of overhead, which is specified in the physical layer [1]. Our work focuses on the performance of the minislot allocation for both centralized scheduling and distributed scheduling (i.e., the MAC layer). The detailed performance analysis for the physical layer of OFDM is beyond the scope of this paper since it has been treated in many previous studies (e.g., [22] and [23]).

Let layer-k SS denote an SS in layer k of the scheduling tree. In this topology, a layer-k SS is k hops (where $k \geq 1$) away from the mesh BS. There are $3k$ SSs in layer k.

We assume that an external packet and an internal packet request m_I and m_A minislots for data transmission, respectively. Let α be the maximum percentage of minislots that can be allocated for external packets in a data subframe for the Partition scheme, where $0\% \leq \alpha \leq 100\%$. Then, $M_c = \lfloor \alpha M \rfloor$. For the CDC-series schemes, all minislots can be allocated to the external packets.

We simulate 10 million packet arrivals in each experiment to ensure that the confidence interval of the 95% confidence level of the output measure is less than 3% of the mean value [24]. Let N_I and N_A be the number of external and internal packet arrivals in an SS, respectively. Let $N_I^{(k)}$ be the total number of external packet arrivals in layer-k SSs, and we have $N_I^{(k)} = 3kN_I$. Let $N_{I,d}^{(k)}$ be the total number of dropped external packet arrivals (i.e., the packet arrival cannot be served) in layer-k SSs. The dropping probability $P_{I,d}^{(k)}$ for the external packet arrivals in layer-k SSs is obtained by

$$P_{I,d}^{(k)} = \frac{N_{I,d}^{(k)}}{N_I^{(k)}} = \frac{N_{I,d}^{(k)}}{3kN_I}. \tag{1}$$

The dropping probability $P_{I,d}$ for the external packet arrivals in a K-layer WMN is obtained using

$$P_{I,d} = \frac{\sum_{k=1}^{K} 3kP_{I,d}^{(k)}}{f} \tag{2}$$

where $f = 3K(K+1)/2$ is the total number of SSs in a K-layer WMN. Let $N_{A,d}^{(a_i)}$ be the number of dropped internal packet arrivals in SS_{a_i}. The dropping probability $P_{A,d}$ for the internal packet arrivals in a K-layer WMN is calculated by

$$P_{A,d} = \frac{\sum_{i=1}^{f} N_{A,d}^{(a_i)}}{fN_A}.$$

This study uses an analytical model to partially validate the simulation model, where we consider the minislot allocation in the external packets in the Partition_RR and CDC_RR schemes. In the analytical model, we assume that, in an SS, the external and internal packet arrivals form Poisson processes with rates λ_I and λ_A, respectively. Assume that the transmission times of the external and internal packets are exponentially distributed with means $1/\mu_I$ and $1/\mu_A$, respectively. By using exponential assumptions, our analytical model can validate the simulation model where we consider general external and internal packet interarrival and transmission time distributions.

We use a K-dimensional (K-D) Markov process to model the minislot allocation for external packet requests for Partition_RR and CDC_RR. For the purpose of illustration, the analytical model for a four-layer WMN is presented. In the Markov process, a state (w, x, y, z) denotes that, in the WMN, there are w, x, y, and z external packets being served in layer-1, layer-2, layer-3, and layer-4 SSs, respectively.

In the Partition and CDC schemes, to serve an external packet arrival at a layer-k SS, other $k-1$ SSs (whose layer numbers are lower than k) will relay the packet to the mesh BS. Each of the layer-$(k-1)$, layer-$(k-2)$, ..., layer-2, and layer-1 SSs requires m_I minislots to relay an external packet to the mesh BS. With the RR algorithm, each minislot is allocated to at most one SS. For the Partition_RR and CDC_RR schemes, the total number of minislots required to serve an external packet arrival at a layer-k SS is km_I. The state space $\mathbf{S}$ for the Markov process is

$$\begin{aligned}\mathbf{S} = \Bigg\{ (w,x,y,z) | 0 \leq w + 2x + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor \\ 0 \leq w \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor, 0 \leq x \leq \left\lfloor \frac{\alpha M}{2m_I} \right\rfloor \\ 0 \leq y \leq \left\lfloor \frac{\alpha M}{3m_I} \right\rfloor, \text{and } 0 \leq z \leq \left\lfloor \frac{\alpha M}{4m_I} \right\rfloor \Bigg\}. \end{aligned} \tag{3}$$

Let $\pi_{w,x,y,z}$ be the steady-state probability for state (w, x, y, z). By convention, $\pi_{w,x,y,z} = 0$ if state $(w, x, y, z) \notin \mathbf{S}$. For all legal states $(w, x, y, z) \in \mathbf{S}$, we have

$$\sum_{(w,x,y,z) \in \mathbf{S}} \pi_{w,x,y,z} = 1. \tag{4}$$

Fig. 8 shows the state transition diagram for this Markov process. Due to page limitations, the description of the state transitions is omitted, and the transitions are listed in

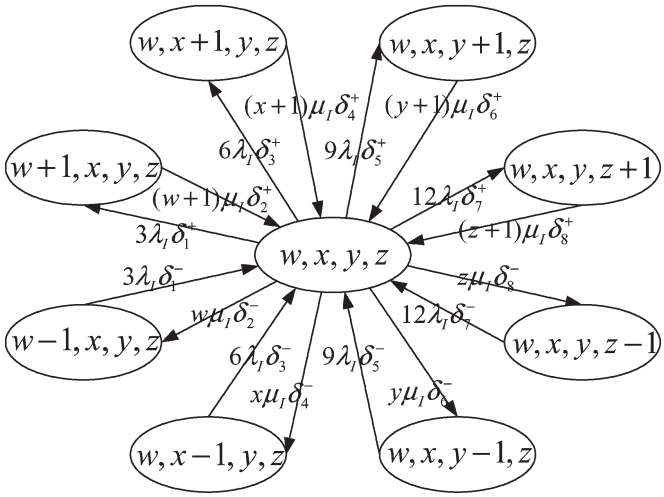

Fig. 8. Markov chain for centralized scheduling in Partition_RR or CDC_RR.

Appendix II. The balance equations for the Markov process are expressed as

$$\begin{aligned}&\left[\left(\delta_1^+ + 2\delta_3^+ + 3\delta_5^+ + 4\delta_7^+\right)3\lambda_I\right.\\&\left.\quad + \left(w\delta_2^- + x\delta_4^- + y\delta_6^- + z\delta_8^-\right)\mu_I\right]\pi_{w,x,y,z}\\&\quad = (w+1)\mu_I\delta_2^+\pi_{w+1,x,y,z} + (x+1)\mu_I\delta_4^+\pi_{w,x+1,y,z}\\&\qquad + (y+1)\mu_I\delta_6^+\pi_{w,x,y+1,z} + (z+1)\mu_I\delta_8^+\pi_{w,x,y,z+1}\\&\qquad + 3\lambda_I\delta_1^-\pi_{w-1,x,y,z} + 6\lambda_I\delta_3^-\pi_{w,x-1,y,z}\\&\qquad + 9\lambda_I\delta_5^-\pi_{w,x,y-1,z} + 12\lambda_I\delta_7^-\pi_{w,x,y,z-1}. \qquad (5)\end{aligned}$$

If an external packet request arrives at the layer-k SS at the state where $w + 2x + 3y + 4z > \lfloor \alpha M/m_I \rfloor - k$, the request is dropped. Therefore, the dropping probability $P_{I,d}^{(k)}$ for the external packets in layer-k SSs is

$$P_{I,d}^{(k)} = \sum_{(w,x,y,z)\in \mathbf{S} \text{ and } w+2x+3y+4z > \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - k} \pi_{w,x,y,z}. \qquad (6)$$

From (4)–(6), the steady-state probability $\pi_{w,x,y,z}$ and $P_{I,d}^{(k)}$ can be solved by using the *successive overrelaxation* method [25]. By applying (6) to (2), we can obtain $P_{I,d}$. In Fig. 9, the solid and dashed curves plot the analysis and simulation results for the $P_{I,d}$ performance for Partition_RR, respectively. This figure indicates that both analysis and simulation results are consistent.

In the following, we investigate the $P_{I,d}$ and $P_{A,d}$ performances for the Partition, CDC, and CDCQ schemes with the RR and Greedy algorithms based on the simulation models. In our study, the input parameters λ_I, λ_A, μ_I, δ_I, δ_A, and t_s are normalized by μ_A. For example, if the expected internal packet transmission time is $(1/\mu_A) = 120$ s, then $\lambda_I = 6\ \mu_A$ means that the expected external packet interarrival time is 20 s.

$P_{I,d}$ Performance for External Packets. Fig. 10 plots $P_{I,d}$ as a function of $\lambda_I/\lambda_I + \lambda_A$, where $K = 4$, $M = 256$, $m_I = m_A = 1$, and $\mu_I = \mu_A$. Fig. 10(a) and (b) studies the effects of α, where $\alpha = 60\%$ and 90%. Fig. 10(a) and (c) investigates

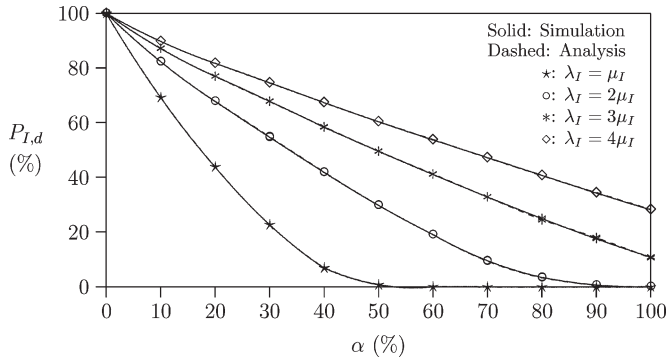

Fig. 9. Comparison of the analytical and the simulation results for Partition_RR ($K = 4$; $M = 256$; $m_I = m_A = 1$; $t_s = 0$).

the effects of scheduling period t_s, where $t_s = \mu_A$ (short scheduling period), and $t_s = 10\ \mu_A$ (long scheduling period). Fig. 10(a) and (d) evaluates the effects of queue delay boundary δ_I, where $\delta_I = \mu_A$ (short delay boundary), and $\delta_I = 10\ \mu_A$ (long delay boundary).

The general phenomena in Fig. 10 are described as follows: The $P_{I,d}$ values increase as $\lambda_I/\lambda_I + \lambda_A$ increases for all schemes. This is because more external packets arrive during a period for larger $\lambda_I/\lambda_I + \lambda_A$ setup, and they compete for the minislots. All schemes with the Greedy algorithm outperform those with the RR algorithm due to the minislot reuse property. Furthermore, the CDC-series schemes outperform the Partition-series schemes in terms of $P_{I,d}$ performance. In the CDC-series schemes, there is no partition boundary. The minislots are allocated to the external packets first (i.e., centralized scheduling is executed), and the remaining minislots are allocated to the internal packets (i.e., distributed scheduling is exercised). Therefore, we observe that the CDC-series schemes gain better $P_{I,d}$ performance for the external packets. We also observe that the queuing mechanism (i.e., the `IQ` queue) does not improve the $P_{I,d}$ performance. To summarize, CDC_Greedy (either with or without the `IQ` queue) has the best $P_{I,d}$ performance among the six schemes.

1) *Base Case:* In Fig. 10(a), where $\alpha = 60\%$, the $P_{I,d}$ values for Partition_Greedy, CDC_Greedy, and CDCQ_Greedy slightly increase and then rapidly increase as $\lambda_I/\lambda_I + \lambda_A$ increases. For example, $P_{I,d}$ for CDC_Greedy slightly increases as $(\lambda_I/\lambda_I + \lambda_A) \leq 0.5$ and then rapidly increases as $(\lambda_I/\lambda_I + \lambda_A) > 0.5$. The minislot allocation for the external packets is exercised at the beginning of the scheduling period, and all external packet requests that arrive during the current scheduling period are buffered before they are processed in the next scheduling period. The external packets are batch arrivals. Consequently, if the number of arrivals during a scheduling period is less than the system capacity, then these arrivals can be served, and $P_{I,d}$ approaches 0. On the other hand, if the number of arrivals during a scheduling period is larger than the system capacity, these requests are more likely to compete for minislots, and the $P_{I,d}$ performance drops fast.
2) *Effects of α:* In Fig. 10(b), we increase α to 90%. The figure indicates that the $P_{I,d}$ performance for the

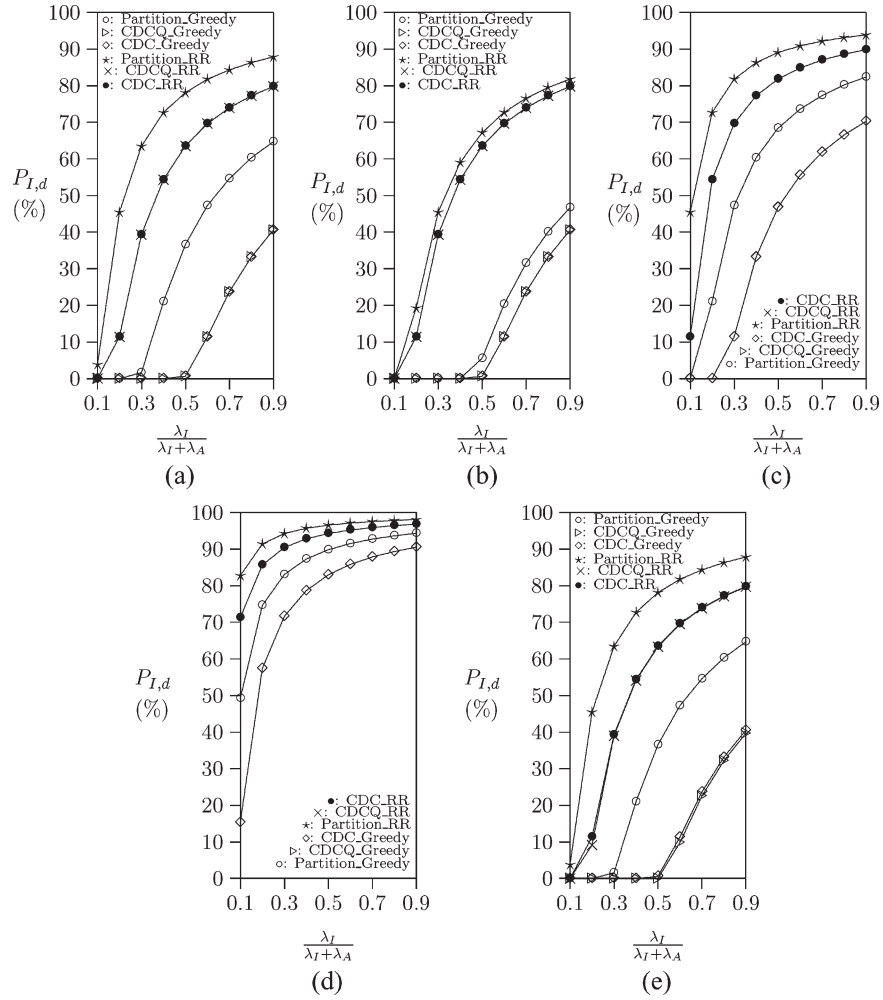

Fig. 10. Comparison for the $P_{I,d}$ performance ($K = 4$; $M = 256$; $m_I = m_A = 1$; $\mu_I = \mu_A$). (a) $\alpha = 60\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = \mu_A$; $\delta_I = \delta_A = \mu_A$. (b) $\alpha = 90\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = \mu_A$; $\delta_I = \delta_A = \mu_A$. (c) $\alpha = 60\%$; $\lambda_I + \lambda_A = 20\ \mu_A$; $t_s = \mu_A$; $\delta_I = \delta_A = \mu_A$. (d) $\alpha = 60\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = 10\ \mu_A$; $\delta_I = \delta_A = \mu_A$. (e) $\alpha = 60\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = \mu_A$; $\delta_I = 10\ \mu_A$; $\delta_A = \mu_A$.

Partition-series schemes approximates that for the CDC-series schemes. This phenomenon is explained as follows: The Partition-series and CDC-series schemes allocate minislots for external packets first. Furthermore, as mentioned in Section II, all minislots are reallocated at the beginning of the scheduling period. Therefore, as α increases, the number of minislots (that can be allocated to external packet arrivals) in the Partition-series schemes approaches that in the CDC-series schemes.

3) *Effects of Total Traffic:* In Fig. 10(c), we increase the total traffic to $\lambda_I + \lambda_A = 20\ \mu_A$, and the $P_{I,d}$ performance for the six schemes becomes worse. In this case, more external packets arrive during a scheduling period; these arrivals compete for minislots, and the worse $P_{I,d}$ performance is observed.
4) *Effects of t_s:* In Fig. 10(d), the length of the scheduling period (i.e., t_s) is ten times that in Fig. 10(a). With larger t_s, we may have two facts: 1) More external packet requests arrive in a scheduling period. 2) It is more likely that an external packet completes during a scheduling period, and the minislots for the external packet may be idle until the next scheduling period, which results in poor minislot utilization. The two aforementioned facts cause the $P_{I,d}$ performance to drop fast for both the Partition-series and CDC-series schemes.
5) *Effects of δ_I:* In Fig. 10(e), we increase the queue delay boundary δ_I for CDCQ_RR and CDCQ_Greedy ten times that in Fig. 10(a). Compare the "×" and "▷" curves in Fig. 10(a) and (e); we observe that increasing δ_I does not significantly improve the $P_{I,d}$ performance. This phenomenon guides us that it is not worth providing long delay boundary for the external packets.

$P_{A,d}$ Performance for Internal Packets: Fig. 11 plots $P_{A,d}$ as a function of $\lambda_I/\lambda_I + \lambda_A$, where $K = 4$, $M = 256$, $m_I = m_A = 1$, and $\mu_I = \mu_A$. Like Fig. 10, Fig. 11 shows the effects of α, the total traffic, scheduling period t_s, and queue delay boundary δ_A.

1) *Base Case:* Consider Fig. 11(a), where $\alpha = 60\%$. In this figure, the $P_{A,d}$ values for Partition_RR and Partition_Greedy are zero and are not affected by $\lambda_I/\lambda_I + \lambda_A$. For the Partition-series schemes, the number of minislots reserved for internal packets is $\lceil (1-\alpha)M \rceil = 103$, and the internal traffic load is $(\lambda_A/\mu_A) = 9$. There are

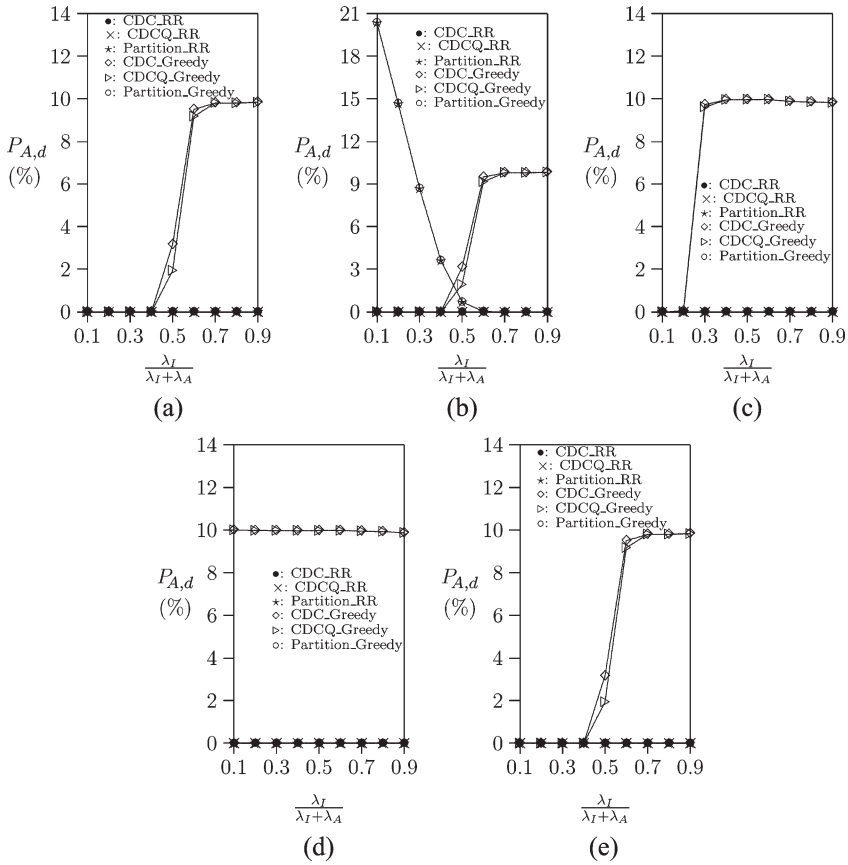

Fig. 11. Comparison for the $P_{A,d}$ performance ($K = 4$; $M = 256$; $m_I = m_A = 1$; $\mu_I = \mu_A$). (a) $\alpha = 60\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = \mu_A$; $\delta_I = \delta_A = \mu_A$. (b) $\alpha = 90\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = \mu_A$; $\delta_I = \delta_A = \mu_A$. (c) $\alpha = 60\%$; $\lambda_I + \lambda_A = 20\ \mu_A$; $t_s = \mu_A$; $\delta_I = \delta_A = \mu_A$. (d) $\alpha = 60\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = 10\ \mu_A$; $\delta_I = \delta_A = \mu_A$. (e) $\alpha = 60\%$; $\lambda_I + \lambda_A = 10\ \mu_A$; $t_s = \mu_A$; $\delta_I = \mu_A$; $\delta_A = 10\ \mu_A$.

sufficient minislots that can serve internal packets. The $P_{A,d}$ values for CDC_RR and CDCQ_RR approach zero. This is because, with the RR algorithm, CDC_RR and CDCQ_RR cannot reutilize the minislots to serve external packets, and more minislots can serve internal packets.

For CDC_Greedy and CDCQ_Greedy, when $(\lambda_I/\lambda_I + \lambda_A) < 0.4$, $P_{A,d}$ is 0; when $0.4 < (\lambda_I/\lambda_I + \lambda_A) < 0.6$, $P_{A,d}$ rapidly increases; and when $(\lambda_I/\lambda_I + \lambda_A) > 0.6$, $P_{A,d}$ approaches 10%. With the Greedy algorithm, CDC_Greedy and CDCQ_Greedy enable the minislot reuse for centralized scheduling, and two facts hold.

Fact 1) The minislots have more chance to be fully utilized for centralized scheduling.

Consider Fig. 7. In the WMN topology, the mesh BS is located at the center. All external packets should be routed through the layer-1 SSs. Thus, we have the following fact:

Fact 2) When $\lambda_I/\lambda_I + \lambda_A$ increases, the minislots of the layer-1 SSs are more likely to be occupied by external packets.

Due to Facts 1 and 2, in CDC_Greedy and CDCQ_Greedy, internal packet requests to the layer-1 SSs are more likely to be dropped. The $P_{A,d}$ values for CDC_Greedy and CDCQ_Greedy are bounded by 10%. The total number of internal packet arrivals in all SSs is ten times that in layer-1 SSs (see Fig. 7).

2) *Effects of α:* In Fig. 11(b), we increase α to 90%. This figure indicates that α does not affect the $P_{A,d}$ performance for CDC_RR, CDC_Greedy, CDCQ_RR, and CDCQ_Greedy. With the Partition-series schemes, the number of minislots reserved for internal packet arrival is $\lceil (1-\alpha)M \rceil = 26$. In this figure, we set $\lambda_I + \lambda_A = 10\ \mu_A$. When $(\lambda_I/\lambda_I + \lambda_A) = 0.1$, the internal traffic load is $(\lambda_A/\mu_A) = 9$, and there are insufficient minislots to serve the new internal packet arrivals. Therefore, we observe that, when $(\lambda_I/\lambda_I + \lambda_A) = 0.1$, the $P_{A,d}$ value for the Partition-series scheme is high (i.e., 20.3%). As $\lambda_I/\lambda_I + \lambda_A$ increases, the internal traffic load decreases. Fewer internal packet requests arrive in a short period, and thus, the $P_{A,d}$ values decrease and approach zero.

3) *Effects of Total Traffic:* In Fig. 11(c), we increase the total traffic $\lambda_I + \lambda_A$ to 20 μ_A. We observe a similar phenomena as in Fig. 11(a).

4) *Effects of t_s:* In Fig. 11(d), the length of the scheduling period is ten times that in Fig. 11(a). With a larger t_s setup, more external packet requests arrive in a scheduling period, fewer minislots can be allocated to internal packets, and the $P_{A,d}$ performance for CDC_Greedy and CDCQ_Greedy becomes worse.

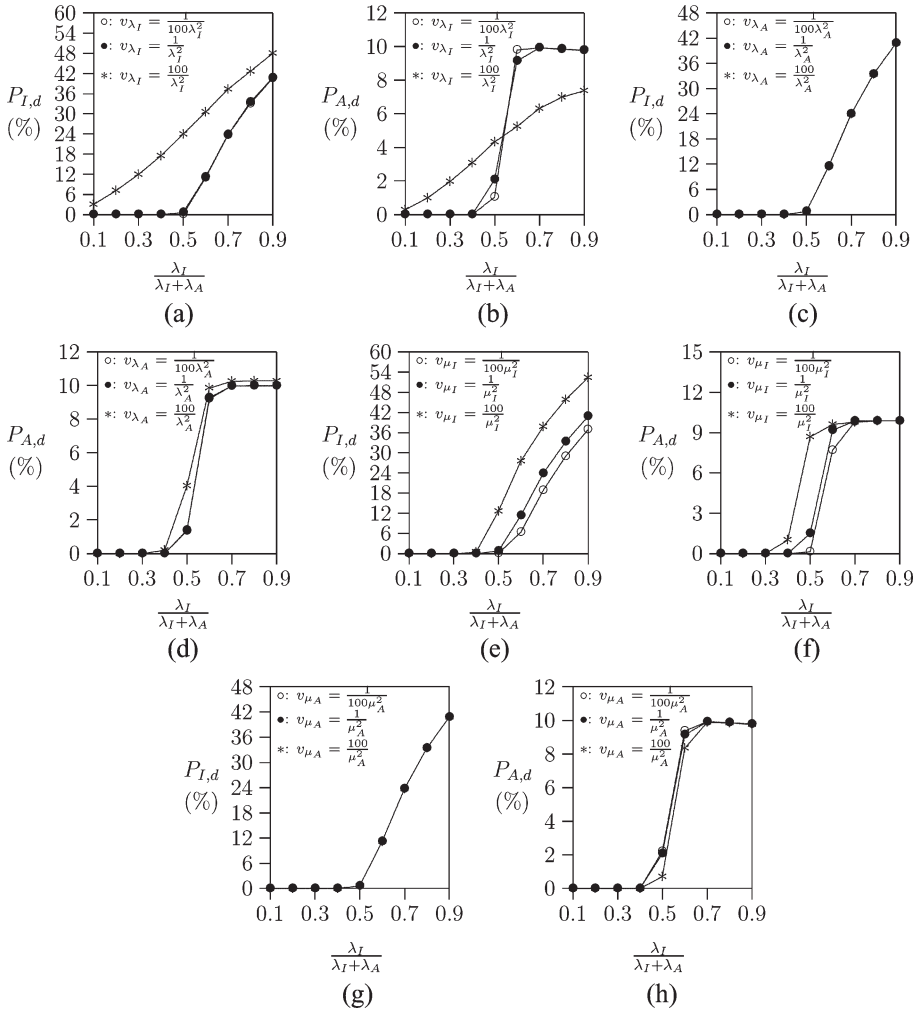

Fig. 12. Effects of the variances of external and internal packet interarrival times and transmission times (CDCQ_Greedy, $K = 4$, $M = 256$, $\alpha = 60\%$, $m_I = m_A = 1$, $\mu_I = \mu_A$, $t_s = \mu_A$, $\delta_I = \delta_A = \mu_A$). (a) and (b) Gamma-distributed external packet interarrival times. (c) and (d) Gamma-distributed internal packet interarrival times. (e) and (f) Gamma-distributed external packet transmission times. (g) and (h) Gamma-distributed internal packet transmission times.

5) *Effects of δ_A:* In Fig. 11(e), we increase the queue delay boundary δ_A for CDCQ_RR and CDCQ_Greedy to be ten times that in Fig. 11(a). Compare the "▷" curves in Fig. 11(a) and (e); we observe that increasing δ_A does not significantly improve the $P_{A,d}$ performance. This phenomenon guides us that it is not worth providing a long delay boundary for the internal packets.

Effects of the Variations of the Distributions for Packet Input Parameters: In the following, we study the effects of the variances for external and internal packet interarrival/ transmission time distributions in the CDCQ_Greedy scheme. We set the interarrival times and transmission times for external (internal) packets to have Gamma distributions with means $1/\lambda_I(1/\lambda_A)$ and $1/\mu_I(1/\mu_A)$ and variances $v_{\lambda_I} = (1/\alpha_1\lambda_I^2)(v_{\lambda_A} = (1/\alpha_2\lambda_A^2))$ and $v_{\mu_I} = (1/\alpha_3\ \ \mu_I^2)(v_{\mu_A} = (1/\alpha_4\ \mu_A^2))$, respectively, where α_1, α_2, α_3, and $\alpha_4 > 0$ are shape parameters. Gamma distributions are considered, because they can be used to approximate many other distributions [16]–[19], [26]. Fig. 12 plots $P_{I,d}$ and $P_{A,d}$ as functions of $\lambda_I/\lambda_I + \lambda_A$, where $K = 4$, $M = 256$, $\alpha = 60\%$, $m_I = m_A = 1$, $\mu_I = \mu_A$, $t_s = \mu_A$, and $\delta_I = \delta_A = \mu_A$. To investigate the impacts of variance for one input parameter, the distributions for the other parameters are set to be exponential (i.e., the shape parameters are set to one).

1) *Variance of External Packet Interarrival Time:* Consider Fig. 12(a) and (b). Fig. 12(a) indicates that the $P_{I,d}$ performance when $v_{\lambda_I} > (1/\lambda_I^2)$ is worse than that that when $v_{\lambda_I} \leq (1/\lambda_I^2)$. For the $P_{A,d}$ performance, Fig. 12(b) indicates that, when $(\lambda_I/\lambda_I + \lambda_A) \leq 0.5$, the $P_{A,d}$ values increase as v_{λ_I} increases. When $(\lambda_I/\lambda_I + \lambda_A) > 0.5$, the $P_{A,d}$ values decrease as v_{λ_I} increases. CDCQ_Greedy gains better performance when $v_{\lambda_I} \leq (1/\lambda_I^2)$.
2) *Variance of Internal Packet Interarrival Time:* Consider Fig. 12(c) and (d). Fig. 12(c) shows that $P_{I,d}$ is not affected by v_{λ_A}. In Fig. 12(d), we observe that the $P_{A,d}$ performance is slightly affected by v_{λ_A}.
3) *Variance of External Packet Transmission Time:* Fig. 12(e) and (f) indicates that both $P_{I,d}$ and $P_{A,d}$ increase as v_{μ_I} increases. We have better performance for CDCQ_Greedy when $v_{\mu_I} < (1/\mu_I^2)$.
4) *Variance of Internal Packet Transmission Time:* From Fig. 12(g) and (h), we observe that $P_{I,d}$ is not affected by v_{μ_A} and that the $P_{A,d}$ values decrease as

v_{μ_A} increases. CDCQ_Greedy performs better when $v_{\mu_A} > (1/\mu_A^2)$.

IV. Conclusion

This paper has proposed the CDC and CDCQ schemes to combine distributed scheduling and centralized scheduling in the IEEE 802.16 Standard. We expect that this will flexibly allocate the minislots in a data subframe to serve the external and internal packets in the WMN, whereas the IEEE 802.16 Standard proposed the Partition scheme for minislot allocation for two kinds of traffic packets. We conducted an analytical model and simulation experiments to investigate the performance for the Partition, CDC, and CDCQ schemes, where we used the RR and Greedy algorithms as baseline algorithms for the centralized scheduling. Our study indicates that the CDC scheme increases the external packet acceptance rate, and the queuing mechanism slightly reduces the dropping probability for external packet requests.

In terms of internal packet performance, this paper indicated that, for the CDC-series schemes with Greedy algorithm, the internal packet request dropping probability is bounded by the ratio of the number of SSs around the mesh BS over the total number of SSs in the WMN. Our study also indicates that the CDC-series schemes with RR algorithm outperform other schemes.

Finally, we investigated the performance of the variances of the external and internal packet interarrival times and transmission times for the CDCQ scheme. Our study concluded that the CDC-series scheme increases the external and internal packet acceptance rates in all cases.

Appendix I
RR and Greedy Algorithms

This section describes the scheduling algorithms, i.e., RR and Greedy, adopted in the second stage of a scheduling period. The scheduling algorithm has two purposes: 1) to calculate the number of external packet requests (from SS_{a_i}) that can be served by the WMN and 2) to determine the minislots that can serve an external packet request. Let (a_u, a_v) denote the wireless link between SS_{a_u} and SS_{a_v}, and let E be the set containing the wireless links among the SSs and the mesh BS. The operation $\mathrm{GET}(a_i, j)$ gets the status of the jth minislot from $\mathbf{S}_{a_i}$, and $\mathrm{SET}(a_i, j, \{\mathtt{TX}, \mathtt{RX}, \mathtt{F}\})$ sets the status of the jth minislot serving for SS_{a_i}. Suppose that the WMN consists of N SSs with the scheduling tree $T = \{a_0(k_{a_0}, n_{a_0}), a_1(k_{a_1}, n_{a_1}), a_2(k_{a_2}, n_{a_2}), \ldots, a_N(k_{a_N}, n_{a_N})\}$, where k_{a_i} is the layer number, n_{a_i} is the position number in layer k_{a_i}, and (k_{a_i}, n_{a_i}) is called the index number of the SS. Without loss of generality, the index number of the mesh BS is (0, 0). Fig. 1(b) shows an example for the scheduling tree for Fig. 1(a). In the two scheduling algorithms, we maintain an array A_{a_i} for SS_{a_i} to store the node IDs of the mesh BS and the SSs on the routing path from the mesh BS to SS_{a_i} in increasing order of their layer numbers. Assume that there are r_{a_i} external packet requests arriving at SS_{a_i} during a scheduling period, and each packet requests m_{I,a_i} minislots for data transmission.

Algorithm 1 Round Robin

```
1: generate the ordered list L as ⟨a'_1, a'_2, ..., a'_N⟩
2: j ← 1
3: for i ← 1 to N do
4:    r'_{a'_i} ← 0
5:    if r_{a'_i} > 0 and m_{I,a'_i} > 0 then
6:       for l ← 1 to r_{a'_i} do
7:          for p ← k_{a'_i} downto 1 do
8:             for q ← 1 to m_{I,a'_i} do
9:                SET(A_{a'_i}[p], j, TX) and SET(A_{a'_i}[p − 1], j, RX)
10:               j ← j + 1
11:               if j = 256 then
12:                  return
13:         r'_{a'_i} ← r'_{a'_i} + 1
```

Fig. 13. Pseudocode for the RR algorithm.

The scheduling algorithm is executed to calculate the number r'_{a_i} of accepted external packet requests for SS_{a_i}. Initially, the statuses of all minislots are set as F. The RR and Greedy algorithms function as follows.

A. Algorithm RR

Fig. 13 shows the pseudocodes for the RR algorithm. The scheduling algorithm outputs a variable r'_{a_i} (i.e., the number of granted external packet requests for SS_{a_i}) and status array $\mathbf{S}_{a_i}$ for SS_{a_i}. The RR algorithm does not adopt the minislot reuse property in TDM, and each minislot is allocated to at most one SS. At line 1 of Fig. 13, using the scheduling tree, the mesh BS sorts the node IDs of the SSs in increasing order of their layer numbers and generates an ordered list $L = \langle a'_1, a'_2, a'_3, \ldots, a'_N \rangle$ such that $k_{a'_1} \le k_{a'_2} \le \cdots \le k_{a'_N}$. At line 2, the variable j is used to indicate the current minislot of the algorithm processes. Initially, j is set to 1. At lines 3–13, a *for* loop is executed to set the status of the minislots for each SS, following the order in the L list. Line 4 initializes $r'_{a'_i}$ as zero. Line 5 checks whether $r_{a'_i} > 0$ and $m_{I,a'_i} > 0$. If $r_{a'_i} = 0$ or $m_{I,a'_i} = 0$ (i.e., $\mathrm{SS}_{a'_i}$ does not request any minislot for data transmission), the mesh BS skips the minislot reservation for $\mathrm{SS}_{a'_i}$. Otherwise, lines 6–13 execute a *for* loop to reserve minislots for each external packet request that arrives in $\mathrm{SS}_{a'_i}$. Lines 7–12 execute two *for* loops to reserve m_{I,a'_i} minislots for each SS in $A_{a'_i}$. Line 9 sets the jth minislots as TX for $\mathrm{SS}_{A_{a'_i}[p]}$ (p is the index number in A_{a_i}, i.e., the minislot is reserved for data transmission) and sets the jth minislot as RX for $\mathrm{SS}_{A_{a'_i}[p-1]}$ (i.e., the minislot is reserved for data reception). Line 10 increases j by 1. Line 11 checks whether $j = 256$. If so (i.e., there are insufficient minislots), the algorithm quits at line 12. As all SSs in $A_{a'_i}$ successfully reserve m_{I,a'_i} minislots for data transmission, line 13 increases $r'_{a'_i}$ by 1, which implies that the request is accepted.

B. Algorithm Greedy

Fig. 14 shows the pseudocodes for the Greedy algorithm. The Greedy algorithm is similar to the RR algorithm; however, in

Algorithm 2 GREEDY

```
generate the ordered list L as ⟨a'_1, a'_2, ..., a'_N⟩
for i ← 1 to N do
    r'_{a'_i} ← 0
    if r_{a'_i} > 0 and m_{I,a'_i} > 0 then
        for l ← 1 to r_{a'_i} do
            SAT ← TRUE
            for p ← k_{a'_i} downto 1 do
                for q ← 1 to m_{I,a'_i} do
                    for j ← 1 to 256 do
                        if COLLISION(A_{a'_i}[p], A_{a'_i}[p − 1], j) = FALSE then
                            SET(A_{a'_i}[p], j, TX) and SET(A_{a'_i}[p − 1], j, RX)
                        else
                            if j = 256 then
                                SAT ← FALSE
                    if SAT = FALSE then
                        break
                if SAT = FALSE then
                    break
            if SAT = FALSE then
                break
            else
                r'_{a'_i} ← r'_{a'_i} + 1
```

Fig. 14. Pseudocode for the Greedy algorithm.

the Greedy algorithm, we adopt the minislot reuse technology. In this algorithm, we implement the Collision function to determine whether the allocation of the jth minislot to SS_{a_u} for transmitting data to its neighbor SS_{a_v} will cause a collision, which is shown here.

```
COLLISION (a_u, a_v, j)
if GET(a_u, j) ≠ F or GET(a_v, j) ≠ F then
    return TRUE
for i ← 1 to N do
    if [(a_i, a_u) ∈ E and GET(a_i, j) = RX] or [(a_i, a_v) ∈ E
    and GET(a_i, j) = TX] then
        return TRUE
return FALSE
```

The Greedy algorithm is similar to the RR algorithm. The difference is that the mesh BS performs the COLLISION function to find a minislot with the smallest index and then allocates the minislot to $\text{SS}_{A_{a'_i}[p]}$ and $\text{SS}_{A_{a'_i}[p-1]}$.

APPENDIX II
STATE TRANSITIONS FOR THE CENTRALIZED SCHEDULING MARKOV PROCESS IN PARTITION_RR AND CDC_RR

Fig. 8 shows the state transition diagram for the Markov process. In this figure, we consider the transitions for state $(w, x, y, z) \in \mathbf{S}$ given here.

- If an external packet arrives in a layer-1 SS when the process is at state $(w, x, y, z) \in \mathbf{S}$, where $w + 2x + 3y + 4z \leq \lfloor \alpha M/m_I \rfloor - 1$, then one minislot is allocated. Define δ_1^+ as

$$\delta_1^+ = \begin{cases} 1, & \text{if } w + 2x + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 1 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

  The process moves from state (w, x, y, z) to $(w + 1, x, y, z)$ with rate $3\lambda_I \delta_1^+$ since there are three layer-1 SSs.
- If the transmission for an external packet completes when the process is at state $(w + 1, x, y, z) \in \mathbf{S}$, two minislots will be released. The process moves from state $(w + 1, x, y, z)$ to (w, x, y, z) with rate $(w + 1)\mu_I \delta_2^+$, where

$$\delta_2^+ = \begin{cases} 1, & \text{if } (w + 1, x, y, z) \in \mathbf{S} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

- If an external packet arrives in a layer-2 SS when the process is at state $(w, x, y, z) \in \mathbf{S}$, where $w + 2x + 3y + 4z \leq \lfloor \alpha M/m_I \rfloor - 2$, then two minislots are allocated. Define δ_3^+ as

$$\delta_3^+ = \begin{cases} 1, & \text{if } w + 2x + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 2 \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

  The process moves from state (w, x, y, z) to $(w, x + 1, y, z)$ with rate $6\lambda_I \delta_3^+$ since there are six layer-2 SSs.
- If the transmission for an external packet is completed when the process is at state $(w, x + 1, y, z) \in \mathbf{S}$, four minislots will be released. The process moves from state $(w, x + 1, y, z)$ to (w, x, y, z) with rate $(x + 1)\mu_I \delta_4^+$, where

$$\delta_4^+ = \begin{cases} 1, & \text{if } (w, x + 1, y, z) \in \mathbf{S} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

- If an external packet arrives in a layer-3 SS when the process is at state $(w, x, y, z) \in \mathbf{S}$, where $w + 2x + 3y + 4z \leq \lfloor \alpha M/m_I \rfloor - 3$, then three minislots are allocated. Define δ_5^+ as

$$\delta_5^+ = \begin{cases} 1, & \text{if } w + 2x + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 3 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

  The process moves from state (w, x, y, z) to $(w, x, y + 1, z)$ with rate $9\lambda_I \delta_5^+$ since there are nine layer-3 SSs.
- If the transmission for an external packet completes when the process is at state $(w, x, y + 1, z) \in \mathbf{S}$, six minislots will be released. The process moves from state $(w, x, y + 1, z)$ to (w, x, y, z) with rate $(y + 1)\mu_I \delta_6^+$, where

$$\delta_6^+ = \begin{cases} 1, & \text{if } (w, x, y + 1, z) \in \mathbf{S} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

- If an external packet arrives from a layer-4 SS when the process is at state $(w, x, y, z) \in \mathbf{S}$, where $w + 2x + 3y +$

$4z \leq \lfloor \alpha M/m_I \rfloor - 4$, then four minislots are allocated. Define δ_7^+ as

$$\delta_7^+ = \begin{cases} 1, & \text{if } w + 2x + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 4 \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The process moves from state (w, x, y, z) to $(w, x, y, z+1)$ with rate $12\lambda_I \delta_7^+$ since there are 12 layer-4 SSs.

- If the transmission for an external packet completes when the process is at state $(w, x, y, z+1) \in \mathbf{S}$, eight minislots will be released. The process moves from state $(w, x, y, z+1)$ to (w, x, y, z) with rate $(z+1)\mu_I \delta_8^+$, where

$$\delta_8^+ = \begin{cases} 1, & \text{if } (w, x, y, z+1) \in \mathbf{S} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The transitions between (w, x, y, z) and $(w-1, x, y, z)$, $(w, x-1, y, z)$, $(w, x, y-1, z)$, and $(w, x, y, z-1)$ are similar to those between (w, x, y, z) and $(w+1, x, y, z)$, $(w, x+1, y, z)$, $(w, x, y+1, z)$, and $(w, x, y, z+1)$. The balance equations for the Markov process are expressed in (5), where $\delta_1^+, \delta_2^+, \ldots, \delta_8^+$ are obtained from (7)–(14), respectively, and

$$\delta_1^- = \begin{cases} 1, & \text{if } (w-1) + 2x + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_2^- = \begin{cases} 1, & \text{if } (w-1, x, y, z) \in \mathbf{S} \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_3^- = \begin{cases} 1, & \text{if } w + 2(x-1) + 3y + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 2 \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_4^- = \begin{cases} 1, & \text{if } (w, x-1, y, z) \in \mathbf{S} \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_5^- = \begin{cases} 1, & \text{if } w + 2x + 3(y-1) + 4z \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 3 \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_6^- = \begin{cases} 1, & \text{if } (w, x, y-1, z) \in \mathbf{S} \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_7^- = \begin{cases} 1, & \text{if } w + 2x + 3y + 4(z-1) \leq \left\lfloor \frac{\alpha M}{m_I} \right\rfloor - 4 \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_8^- = \begin{cases} 1, & \text{if } (w, x, y, z-1) \in \mathbf{S} \\ 0, & \text{otherwise.} \end{cases}$$

ACKNOWLEDGMENT

The authors would like to thank the editor and the three anonymous reviewers for their valuable comments and their efforts, which have significantly improved the quality of this paper.

REFERENCES

[1] *Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std. 802.16-2004, May 2004.
[2] D. Niyato and E. Hossain, "A queuing-theoretic and optimization-based model for radio resource management in IEEE 802.16 broadband wireless networks," *IEEE Trans. Comput.*, vol. 55, no. 11, pp. 1473–1488, Nov. 2006.
[3] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi, "Performance evaluation of the IEEE 802.16 MAC for QoS support," *IEEE Trans. Mobile Comput.*, vol. 6, no. 1, pp. 26–38, Jan. 2007.
[4] *Local and Metropolitan Area Networks—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Std. 802.3-2005, Dec. 2005.
[5] *Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications*, IEEE Std. 802.11-1997, Nov. 1997.
[6] R. Nelson and L. Kleinrock, "Spatial TDMA: A collison-free multihop channel access protocol," *IEEE Trans. Commun.*, vol. COM-33, no. 9, pp. 934–944, Sep. 1985.
[7] A.-M. Chou and V. O. K. Li, "Slot allocation strategies for TDMA protocols in multihop packet radio networks," in *Proc. IEEE INFOCOM*, May 1992, pp. 710–716.
[8] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, "Interference-aware IEEE 802.16 WiMax mesh networks," in *Proc. IEEE VTC—Spring*, May 2005, pp. 3102–3106.
[9] D. Kim and A. Ganz, "Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems," in *Proc. BROADNET*, Oct. 2005, pp. 833–839.
[10] J. Chen, C. Chi, and Q. Guo, "An odd-even alternation mechanism for centralized scheduling in WiMAX mesh network," in *Proc. IEEE GLOBECOM*, Nov. 2006, pp. 1–6.
[11] M. Cao, V. Raghunathan, and P. R. Kumar, "A tractable algorithm for fair and efficient uplink scheduling of multi-hop WiMax mesh networks," in *Proc. WiMesh*, Sep. 2006, pp. 93–100.
[12] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu, "Modeling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode," in *Proc. ACM MobiHoc*, May 2005, pp. 78–89.
[13] S.-M. Cheng, P. Lin, D.-W. Huang, and S.-R. Yang, "A study on distributed/centralized scheduling for wireless mesh network," in *Proc. ACM IWCMC*, Jul. 2006, pp. 599–604.
[14] I. Chlamtac and A. Lerner, "Fair algorithms for maximal link activation in multihop radio networks," *IEEE Trans. Commun.*, vol. COM-35, no. 7, pp. 739–746, Jul. 1987.
[15] I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 23–29, Feb. 1994.
[16] P. Lin and Y.-B. Lin, "Channel allocation for GPRS," *IEEE Trans. Veh. Technol.*, vol. 50, no. 2, pp. 375–387, Mar. 2001.
[17] P. Lin, "Channel allocation for GPRS with buffering mechanisms," *ACM Wireless Netw.*, vol. 9, no. 5, pp. 431–441, Sep. 2003.
[18] P. Lin, W.-R. Lai, and C.-H. Gan, "Modeling opportunity driven multiple access in UMTS," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1669–1677, Sep. 2004.
[19] P. Lin, C.-H. Gan, and C.-C. Hsu, "OVSF code channel assignment with dynamic code set and buffering adjustment for UMTS," *IEEE Trans. Veh. Technol.*, vol. 54, no. 2, pp. 591–602, Mar. 2005.
[20] R.-G. Cheng, S.-M. Cheng, and P. Lin, "Power-efficient routing (PER) mechanism for ODMA systems," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1131–1319, Jul. 2006.
[21] A. Raniwala, K. Gopalan, and T.-C. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM Mobile Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, Apr. 2004.
[22] S. Redana and M. Lott, "Performance analysis of IEEE 802.16a in mesh operation mode," in *Proc. 13th IST SUMMIT*, Jun. 2004.
[23] C. Hoymann, "Analysis and performance evaluation of the OFDM-based metropolitan area network IEEE 802.16," *Comput. Netw.*, vol. 49, no. 3, pp. 341–363, Oct. 2005.
[24] R. Jain, Y.-B. Lin, C. Lo, and S. Mohan, "A caching strategy to reduce network impacts of PCS," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 8, pp. 1434–1444, Oct. 1994.
[25] R. Cooper, *Introduction to Queueing Theory*. New York: North-Holland, 1981.
[26] S. M. Ross, *Stochastic Processes*. Hoboken, NJ: Wiley, 1983.

Mohsen Guizani (S'87–M'90–SM'98) received the B.S. (with distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, in 1984, 1986, 1987, and 1990, respectively.

He is currently a Professor and the Chair of the Department of Computer Science, Western Michigan University, Kalamazoo. He currently serves on the editorial boards of six technical journals and is the Founder and Editor-in-Chief of *Wireless Communications and Mobile Computing Journal* (Wiley, http://www.interscience.wiley.com/jpages/1530-8669/) and the *Journal of Computer Systems, Networks and Communications*, (Hindawi, http://www.hindawi.com/journals/).He is the author of six books and more than 200 publications in refereed journals and conference proceedings. His research interests include computer networks, wireless communications and mobile computing, and optical networking.

Dr. Guizani is an active member of the IEEE Communication Society, the IEEE Computer Society, the American Society for Engineering Education, and the Association for Computing Machinery. He is the past Chair of TAOS and the Vice-Chair of WTC IEEE ComSoc Technical Committees. He has been a Guest Editor for a number of special issues of IEEE journals and magazines. He has also served as a member, Chair, and General Chair of a number of conferences. He is also the Founder and General Chair of the IEEE International Conference of Wireless Networks, Communications, and Mobile Computing (IEEE WirelessCom 2005, 2006, 2007, and 2008). He was the recipient of both the Best Teaching Award and the Excellence in Research Award from the University of Missouri, Columbia, in 1999 (a college-wide competition); the Best Research Award from the King Fahd University of Petroleum and Minerals in 1995 (a university-wide competition); and the Best Teaching Assistant for two consecutive years at Syracuse University in 1988 and 1989.

Phone Lin (M'02–SM'06) received the BSCSIE and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1996 and 2001, respectively.

From August 2001 to July 2004, he was an Assistant Professor with the Department of Computer Science and Information Engineering (CSIE), National Taiwan University, Taipei, Taiwan. Since August 2004, he has been an Associate Professor with the Department of CSIE and the Graduate Institute of Networking and Multimedia, National Taiwan University. His current research interests include personal communications services, wireless Internet, and performance modeling. He has published more than 20 international SCI journal papers (most of which are IEEE TRANSACTIONS and ACM papers).

Dr. Lin is an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a Guest Editor for *IEEE Wireless Communications* special issue on Mobility and Resource Management, and a Guest Editor for *ACM/Springer MONET* special issue on Wireless Broad Access. He is also an Associate Editorial Member for the *WCMC Journal*. He has received many research awards. He was elected as the Best Young Researcher at the 3rd IEEE ComSoc Asia-Pacific Young Researcher Awards in 2007. He was a recipient of the Research Award for Young Researchers from the Pan Wen-Yuan Foundation in Taiwan in 2004, a recipient of the K. T. Li Young Researcher Award from the ACM Taipei Chapter in 2004, a recipient of the Wu Ta You Memorial Award of the National Science Council (NSC) in Taiwan in 2005, a recipient of the Fu Suu-Nien Award of NTU in 2005 for his research achievements, and a recipient of the 2006 Young Electrical Engineering Award of the Chinese Institute of Electrical Engineering. Dr. Lin was listed in *Who's Who in Science and Engineeering* in 2006.

Shin-Ming Cheng (M'06) received the B.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2000 and 2007, respectively.

He is currently a Postdoctoral Researcher with the Department of Electrical Engineering, National Taiwan University. His research interests include mobile networking, wireless communications, and network security.

Di-Wei Huang received the B.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2004 and the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2006.

During 2006–2007, he was on active duty of obligatory military service in Taiwan. He is currently with the Department of Computer Science and Information Engineering, National Taiwan University. His current research interests include wireless networking and mobile computing.

Huai-Lei Fu received the B.S. degree in computer science from Tatung University, Taipei, Taiwan, R.O.C., in 2005 and the M.S. degree in computer science from Yuan Ze University, Taoyuan, Taiwan, in 2007. He is currently working toward the Ph.D. degree in computer science with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei.

His research interests include wireless communication, wireless mesh networks, and peer-to-peer streaming.