Improving the Performances of Distributed Coordinated Scheduling in IEEE 802.16 Mesh Networks

Shie-Yuan Wang, *Senior Member, IEEE*, Chih-Che Lin, *Student Member, IEEE*, Han-Wei Chu, Teng-Wei Hsu, and Ku-Han Fang

***Abstract*—The IEEE 802.16 mesh network is a promising next-generation wireless backbone network. In such a network, setting the holdoff time for nodes is essential to achieving good performances of medium-access-control-layer scheduling. In this paper, we propose a two-phase holdoff time setting scheme to improve network utilization. Both static and dynamic approaches of this scheme are proposed, and their performances are compared against those of the original schemes. Our simulation results show that both approaches significantly increase the utilization of the control-plane bandwidth and decrease the time required to establish data schedules. In addition, both approaches provide efficient and fair scheduling for IEEE 802.16 mesh networks and generate good application performances.**

***Index Terms*—Distributed scheduling, IEEE 802.16(d), mesh network.**

I. Introduction

In RECENT years, the IEEE 802.16 standard (WiMAX) [1] has been attracting much attention. This next-generation broadband communication technology aims to provide high network throughput, low packet delay, and low packet loss rate. In this standard, two operational modes are defined: 1) the point-to-multipoint (PMP) mode and 2) the mesh mode. The PMP mode provides for one-hop communication between a base station (BS) and several subscriber stations (SSs). In contrast, the mesh mode constructs a multihop wireless backbone network.

In the mesh mode, packets can be transferred in a peer-to-peer manner, i.e., two SS nodes can directly communicate with each other without the aid of the BS node. In this mode, network accesses are managed in a manner much like time-division multiple access. As shown in Fig. 1, the network bandwidth is first divided into frames, each of which is subdivided into one control and one data subframe. A control subframe is further divided into transmission opportunities (TxOpp), whereas a data subframe is further divided into minislots. Control messages and data packets are transferred over transmission opportunities and minislots, respectively.

To avoid conflict when using transmission opportunities, the mesh mode defines two scheduling modes: 1) the centralized mode and 2) the distributed mode. In the centralized scheduling mode, a network is partitioned into tree-based clusters. Each cluster has a BS node that is responsible for allocating network resources to the SS nodes that it services. Although the centralized scheduling mode provides collision-free transmissions for control and data packets, it has several disadvantages, which are described here.

First, the number of routes that can be utilized is unnecessarily reduced. The reason is that the centralized scheduling mode uses a tree-based topology, which cannot exploit all possible routes in a network, as compared with a mesh-based topology. For the same reason, the only route between two nodes on the tree may not be the shortest one between them if, instead, a mesh-based topology were used. In addition, the root node of the tree is likely to become the performance bottleneck, because many packets need to pass through it to reach their destination nodes. Second, it is difficult to efficiently exploit the spatial reuse property of wireless communication in the centralized scheduling mode. The message format defined in this mode only allows a BS node to notify an SS node of the bandwidth allocated to it. There is no field in the message that allows a BS node to specify the start and end minislot offsets for an allocation. As such, to avoid interference, each SS node has to take a conservative approach to derive its own data schedule. Allocating minislots in this way is collision-free but results in only one active SS node per cluster at any given time. More detailed explanations about this problem are provided in the Appendix.

In contrast, the distributed scheduling mode provides two advantages: First, the distributed scheduling mode uses a mesh topology. This allows all possible routing paths to be utilized to avoid performance bottlenecks. In addition, spatial reuse of wireless communication can be exploited to increase network capacity. Second, the distributed scheduling mode establishes data schedules on an on-demand basis; thus, network bandwidth can be more efficiently utilized.

In the IEEE 802.16 mesh network standard, the distributed scheduling mode is further divided into two operational modes:

Manuscript received May 15, 2007; revised August 16, 2007 and October 4, 2007. This work was supported in part by the Ministry of Education Program for Promoting Academic Excellence of Universities under Grant 95-2752-E-009-014-PAE and in part by the National Science Council under Grant 96-2628-E-009-073. The review of this paper was coordinated by Dr. Q. Zhang.

S.-Y. Wang, C.-C. Lin, H.-W. Chu, and T.-W. Hsu are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: shieyuan@csie.nctu.edu.tw).

K.-H. Fang is with the Cyberlink Corporation, Taipei 231, Taiwan, R.O.C.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2007.912330

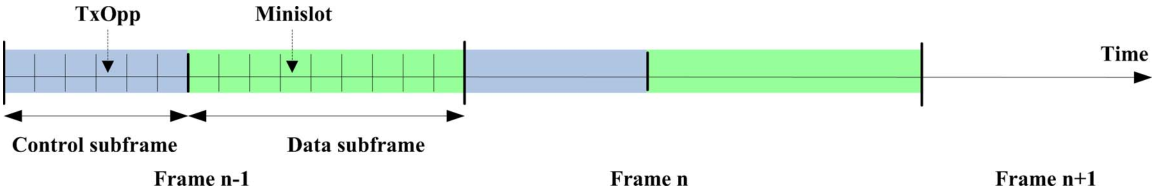


Fig. 1. Frame structure of the IEEE 802.16 mesh mode.

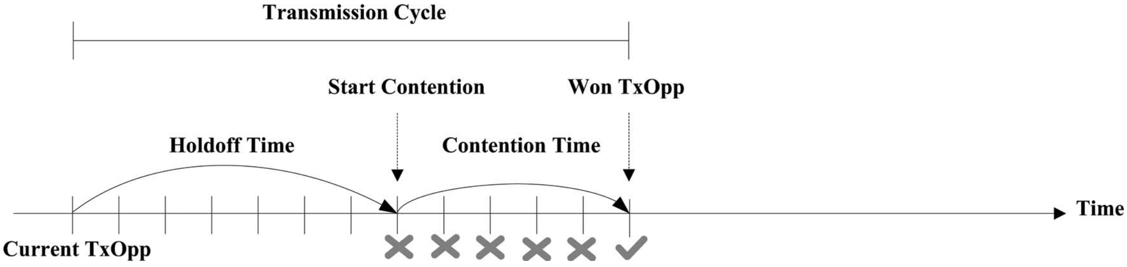


Fig. 2. Node's transmission cycle comprises the holdoff time and the contention time.

1) the coordinated mode and 2) the uncoordinated mode. In the distributed coordinated scheduling mode, the control messages required to establish data schedules are transmitted over transmission opportunities without collisions. In contrast, in the distributed uncoordinated scheduling mode, such control messages can only be transmitted on the transmission opportunities left from the distributed coordinated scheduling mode or on unallocated minislots. Because of this design, the distributed coordinated scheduling mode provides better quality-of-service (QoS) supports than the distributed uncoordinated scheduling mode. In this paper, we focus only on the distributed coordinated scheduling mode.

In the distributed coordinated scheduling mode, the role of every network node is the same. Each node uses the same pseudo-random election algorithm to resolve contention of transmission opportunities. Control messages, such as mesh network configuration (MSH-NCFG) and mesh distributed coordinated scheduling (MSH-DSCH) messages, are transmitted on transmission opportunities determined by this algorithm. (The MSH-NCFG message is used to carry information for network initialization, and the MSH-DSCH message is used to carry information for data scheduling.) The contention resolution of MSH-DSCH message transmissions is explained here to illustrate how this pseudo-random algorithm works.

During operation, each node listens to the MSH-DSCH messages advertised by its neighboring nodes. Based on the scheduling information carried in the MSH-DSCH messages, each node knows the transmission interval (more details about this will be illustrated and explained in Fig. 9) of each of its neighboring nodes. More specifically, it knows for which transmission opportunities its neighboring nodes may contend. Each node then uses the same election algorithm to determine the winning node for a given transmission opportunity. This algorithm takes a specified transmission opportunity number and the IDs of all the nodes contending for this transmission opportunity as input. It outputs the ID of the winning node whose computed value is the largest among all the competing nodes. Since every node uses the same algorithm and the same input, every node knows which node will win a given transmission opportunity within its two-hop neighborhood (explained in Section III-A2). As such, no collision will occur on any transmission opportunity. For a node, if it cannot win a given transmission opportunity, it repeats the aforementioned process, with the next transmission opportunity (i.e., the previous transmission opportunity number plus one) as input, until it eventually wins one transmission opportunity.

After a node wins a transmission opportunity, the IEEE 802.16 standard requires it to refrain from contending for another transmission opportunity in a certain number of consecutive transmission opportunities, which is called the holdoff time. As shown in Fig. 2, a node's transmission cycle comprises the holdoff time and the contention time. The contention time is defined as the number of consecutive transmission opportunities in which a node should contend for access (i.e., participate in the algorithm computation) until it wins one. Although this holdoff time design may increase delays in transmitting control messages, it allows for fair accesses to transmission opportunities among competing nodes. Using this design, every winning node has to suspend its contention for its next MSH-DSCH transmission opportunity during its holdoff time. As such, a winning node cannot obtain more than one transmission opportunity in one transmission cycle, preventing a node from monopolizing a wireless channel.

The effect of the holdoff time value can be discussed from two aspects: On one hand, if the holdoff time is set to a too large value, network nodes will suffer from long delays in transmitting control messages and thus cannot fully utilize the link bandwidth. In the distributed coordinated scheduling mode, a network node can transmit a data packet only after it has established a data schedule with the next-hop node of the packet. To establish a data schedule, a node needs to perform a three-way handshake procedure, which requires three transmission opportunities to exchange three MSH-DSCH messages. Since using a large holdoff time value will increase the transmission cycle for transmitting a control message, the time required to establish a data schedule (and, thus, sending a data packet) will be increased as well. On the other hand, if the holdoff time is set to a too-small value, the number of nodes competing for a transmission opportunity will be large. This will lead to high contention for transmission opportunities. Under such a congested condition, nodes may experience unpredictable packet delays and unfairly share transmission opportunities. In summary, the holdoff time should be set to an appropriate value that is large enough to avoid congestion but small enough to avoid large transmission delays.

In the standard, every node in a network is required to use a holdoff time value that is greater than or equal to 16. This design cannot optimally perform for all networks, because it does not consider the possibility that node density may vary at different locations of the network. Ideally, if a node is located in a high-node-density area, it should use a larger holdoff time value to avoid congestion. In contrast, if a node is located in a low-node-density area, it should use a smaller holdoff time value to reduce unnecessary large transmission delays.

To address these shortcomings of the standard, we propose a two-phase holdoff time setting scheme in this paper. This scheme uses different holdoff time values for the network initialization phase and the data scheduling phase. As such, the success of network initialization can be guaranteed, and good scheduling performances can be achieved. In addition, the scheme allows different nodes to use different holdoff time values to match the node densities around them, which reduces unnecessary transmission delays and improves network utilization without causing network congestion. Static and dynamic approaches of this scheme are proposed, and their performances are compared with those of the original schemes under various network conditions. Our simulation results show that the dynamic approach significantly improves network performances without causing network congestion for IEEE 802.16 mesh networks.

The remainder of this paper is organized as follows: In Section II, we survey related work. In Section III, the necessity of the two-phase holdoff time setting scheme is explained. In Section IV, we present the design of the two-phase holdoff time setting scheme. We also present the details of the static and dynamic approaches. In Section V, we compare the performances of the static and dynamic approaches with those of the schemes that use a fixed holdoff time value for all nodes. Finally, we conclude this paper in Section VI.

II. Related Work

So far, very few papers in the literature have studied the performances of the distributed coordinated scheduling mode of the IEEE 802.16 mesh network. Most papers (e.g., [2]–[4]) focus on the centralized scheduling mode. For the distributed coordinated scheduling mode, Cao *et al.* [5] studied the effect of the holdoff time value in terms of control message transmission cycles and the time required to establish data schedules. The definition of the holdoff time specified in the standard is shown in

$$\text{holdoff time} = 2^{exp+base} \tag{1}$$

where the base should be set to 4, and the exponent value can vary. The authors vary the holdoff time exponent value from 0 to 4 to observe the effects of the resultant holdoff time value. In [6], Bayer *et al.* analyzed control message transmission cycles and round trip times (RTTs) using different holdoff time base values.

In [7], Bayer *et al.* proposed a dynamic holdoff time setting scheme to improve the scheduling performance of the distributed scheduling mode. The goal of their paper is similar to that of our paper. However, there are fundamental design and implementation differences between these two papers. In the following, we present some details of the paper to point out the differences.

The main idea of the scheme proposed in [7] is given as follows: Network nodes that are transmitting, receiving, or forwarding data packets should use smaller holdoff time values to exchange MSH-DSCH messages more quickly. In contrast, nodes that are not transmitting, receiving, or forwarding data packets should use larger holdoff time values to reduce contentions for transmission opportunities. Bayer *et al.* classify network nodes into four classes: 1) BS; 2) active; 3) sponsoring; and 4) inactive. The BS class comprises all BS nodes in the network; the sponsoring class consists of nodes that are sponsoring new nodes (i.e., allocating data schedules for forwarding the control messages initiated from a new node's network registration procedure). A sponsoring node is a node that has been selected by one of its neighbors as the next-hop node toward the destination. Network nodes that are transmitting, receiving, or forwarding data packets are called active nodes; nodes that are idle are called inactive nodes. Each of these four classes has its own range of holdoff time exponent values. The upper bounds of these classes' holdoff time exponent values are shown in the following:

$$0 \leq \mathrm{ME}_{\mathrm{BS}} \leq \mathrm{ME}_{\mathrm{act}} < \mathrm{ME}_{\mathrm{sp}} < \mathrm{ME}_{\mathrm{inact}} \leq 7 \tag{2}$$

where $\mathrm{ME}_{\mathrm{BS}}$ denotes the maximum holdoff time exponent value of the BS class, $\mathrm{ME}_{\mathrm{act}}$ denotes the maximum holdoff time exponent value of the active class, $\mathrm{ME}_{\mathrm{sp}}$ denotes the maximum holdoff time exponent value of the sponsoring class, and $\mathrm{ME}_{\mathrm{inact}}$ denotes the maximum holdoff time exponent value of the inactive class.

Since a smaller holdoff time exponent value results in a smaller holdoff time value, nodes using a smaller holdoff time exponent value will refrain themselves from contending for transmission opportunities for a shorter period of time. Such nodes, therefore, can, on average, win a transmission opportunity faster than nodes using a larger holdoff time exponent value. Due to this reason, the nodes of the BS class, on average, can transmit MSH-DSCH messages faster than (or as fast as if $\mathrm{ME}_{\mathrm{BS}} = \mathrm{ME}_{\mathrm{act}}$) the nodes belonging to the active class. Similarly, the nodes belonging to the active class, on average, can transmit MSH-DSCH messages faster than the nodes belonging to the sponsoring or inactive class.

Although this scheme provides some advantages, it has several disadvantages described here.

First, this scheme needs to rely on a collaborative routing protocol to determine if a node should belong to the active class or the inactive class. For each node, it should consult the routing protocol to check whether it is selected as a potential forwarding node for a routing path. If so, it should switch to the active class. Otherwise, it should belong to the inactive class.

Second, when a mesh network becomes highly loaded, it is very likely that every node has data to send most of the time. In such a condition, most of the nodes will belong to the active class and thus will have the same range of holdoff time values. As such, the multi-class design of this scheme will degenerate to the original single-class design in this condition. To avoid this problem, which is caused by a heavy load, an admission control mechanism can be used.

Third, not all active nodes actually need to use smaller holdoff time values at all times. Active nodes need not establish data schedules over all transmission opportunities that they win. This is because an established data schedule can be valid for N frames, where $1 \leq N \leq 128$. (Note that each frame can contain M transmission opportunities, where $2 \leq M \leq 15$, depending on the network setting.) If an active node excessively wins transmission opportunities without considering whether it has data to send, it will waste many transmission opportunities that could otherwise be given to other nodes that have data to send.

Lastly, in this scheme, when switching to a higher priority class, a node should first set its holdoff time exponent value to zero and then gradually increment this value by one until its and its neighboring nodes' Mx values are no longer above a predefined threshold. This means that, before a node can stabilize its holdoff time exponent value, it will excessively contend for transmission opportunities and thus waste the control-plane bandwidth. To understand the meaning of Mx, the standard defines that each node should use two shorter fixed-length fields exp and Mx to represent its next transmission opportunity number. The relationship between these two fields and a transmission opportunity number is given as follows:

$$2^{exp} * Mx < next\_TxOpp <= 2^{exp} * (Mx + 1) \qquad (3)$$

where $next\_TxOpp$ denotes a node's next transmission opportunity number. The detailed information about (3) will be explained in Section IV-C. In contrast, our proposed two-phase holdoff time setting scheme does not incur the aforementioned problems. In addition to improving the scheduling performances of the distributed coordinated mode, our proposed scheme ensures the success of network initialization without wasting the control-plane bandwidth. The dynamic approach of our proposed scheme can dynamically reduce a node's holdoff time exponent value when the node needs to establish a data schedule. As such, the dynamic approach can more efficiently use transmission opportunities than the scheme proposed in [7]. Besides, the dynamic approach need not employ a collaborative routing protocol; therefore, its design and implementation complexities are much lower than those of the scheme proposed in [7].

III. Necessity of the Two-Phase Holdoff Time Setting Scheme

In this section, we explain why the proposed two-phase holdoff time setting scheme is necessary and important. The holdoff time value affects not only the efficiency of medium access control (MAC)-layer scheduling but the success of network initialization as well. Network initialization is very important, because a node must successfully initialize and attach itself to the network before it can start transmitting and receiving packets. When designing a new holdoff time setting scheme, it is important to ensure that the network nodes under the new scheme can still successfully attach themselves to the network. When evaluating a holdoff time setting scheme, three aspects must be carefully considered: 1) the success of network initialization; 2) the efficiency of MAC-layer scheduling; and 3) the fairness of resource sharing. In the following, we first define several relevant performance metrics and explain their meanings. These metrics will be used throughout this paper. Then, we elaborate on the effect of the holdoff time value using the simulation results of four fixed-value holdoff time setting schemes. Finally, the necessity of the two-phase holdoff time setting scheme is explained.

A. Performance Metrics

In the following, several performance metrics used throughout this paper are defined.

1) SRNI: The success rate of network initialization (SRNI) is defined as

$$\mathrm{SRNI} = \frac{\mathrm{NC}_{\mathrm{success}}}{\mathrm{NC}_{\mathrm{total}}} \qquad (4)$$

where $\mathrm{NC}_{\mathrm{success}}$ denotes the number of cases in which the network succeeds in initialization, and $\mathrm{NC}_{\mathrm{total}}$ denotes the number of total cases.

The success of a network initialization is defined as follows: For a network case, if all of its nodes successfully initialize and attach themselves to the network, the initialization of this network case succeeds. In contrast, if any node fails to perform its initialization and attachment procedures, the initialization of this network case fails. As stated before, the success of network initialization is very important, because, before a node can start sending or receiving packets, it must successfully initialize and attach itself to the network. In [8], we show that, in dense networks, excessive control message transmissions can lead to severe message collisions due to the well-known "*hidden terminal*" problem. As such, if the holdoff time value is set to a value that is too small, the control messages containing information for a node to attach itself to the network will be excessively dropped. In such a condition, the node's initialization process will fail, causing the SRNI to decrease.

2) ATOUN: The utilization of a node's control-plane bandwidth is an important metric used to evaluate the efficiency of a holdoff time setting scheme. To define the utilization of the control-plane bandwidth from the perspective of an SS node, one should first understand the notion of the two-hop neighborhood. The two-hop neighborhood of a node is defined as the set comprising all of its one-hop and two-hop neighboring nodes (including itself), i.e.,

$$nbr(j) = \{i \mid \text{node } i \in \text{node } j'\text{s two-hop neighborhood}\}. \qquad (5)$$

For an SS node, since the *hidden terminal* problem can occur only with nodes that are in its two-hop neighborhood, the

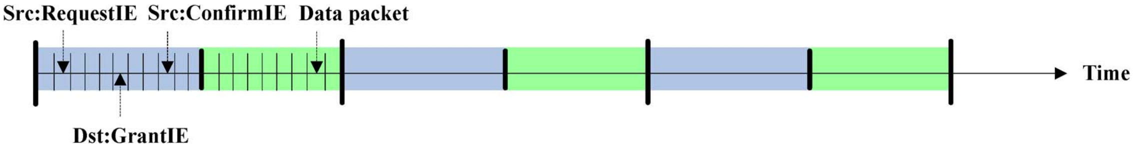


Fig. 3. Good case for establishing a data schedule in the distributed coordinated scheduling mode when the network is not congested.

IEEE 802.16 standard requires that each node resolves the contention of each transmission opportunity with the nodes in its two-hop neighborhood. As such, within a node's two-hop neighborhood, only one node can transmit a control message at any given transmission opportunity.

The average transmission opportunity utilization viewed from node j is defined as follows:

$$\text{AvgTxOpp}(j) = \frac{\sum_{i \in nbr(j)} txnum(i)}{total(j)} \tag{6}$$

where $txnum(j)$ denotes the number of transmission opportunities won by node j, and $total(j)$ denotes the number of total transmission opportunities since node j has attached itself to the network. This definition indicates how well the nodes in node j's two-hop neighborhood (including node j itself) together utilize the network's transmission opportunities. Ideally, the average transmission opportunity utilization viewed from each node should be 100%, which indicates that, from the perspective of each node, each transmission opportunity is used by one and only one node, and no transmission opportunity is left unused.

The average transmission opportunity utilization of nodes (ATOUN) metric of a network case is defined as follows:

$$\text{ATOUN} = \frac{\sum_{j=1}^{m} \text{AvgTxOpp}(j)}{m} \tag{7}$$

where m is the number of nodes in a network case. It is the average across all the nodes' AvgTxOpp values in a network case.

The ATOUN metric reflects the utilization of the control-plane bandwidth from the aggregate of the local view of each node. A higher value of this metric indicates that a network case has higher control-plane bandwidth utilization; a lower value indicates that a network case has lower control-plane bandwidth utilization.

The ATOUN metric does not measure the fairness of bandwidth sharing in a network. To solve this problem, we designed another metric, which is explained in Section III-A4, to evaluate how fairly network nodes share the control-plane bandwidth.

3) ATHPT: The average three-way handshake procedure time (ATHPT) metric is defined as the average time required by the three-way handshake procedure to establish a data schedule across all network nodes in a case. This metric is computed as follows: For a network case, we first use the following expression to average the times required to establish data schedules for every node:

$$\text{THPT}(j) = \frac{\sum_{i=1}^{n} t_{ij}}{n} \tag{8}$$

where t_{ij} denotes the time required to establish the ith data schedule of node j, and n is the number of node j's data schedules. We then use the following expression to compute the case's ATHPT value, which is the average across all the nodes' THPT values:

$$\text{ATHPT} = \frac{\sum_{j=1}^{m} \text{THPT}(j)}{m} \tag{9}$$

where m is the number of nodes in a network case. Like the ATOUN metric, for each scheme, the average and standard deviation of its ATHPT values across all simulation cases will be presented.

ATHPT is a common metric used in the literature to evaluate the effect of the holdoff time value. The three-way handshake procedure requires transmitting three MSH-DSCH messages, each of which contains the request, grant, and confirm information elements (IEs), respectively. The detailed procedure is described here.

First, the requesting node transmits a request IE to the peer node. The request IE specifies the number of requested minislots on the peer node and the available minislots on the requesting node from which the peer node can choose. On receiving the request IE, the peer node decides whether it would like to accept this request. If not, it ignores this message. Otherwise, out of its own available minislots, it allocates a data schedule from the requesting node's available minislots. The peer node then transmits a grant IE containing the information of the allocated data schedule to the requesting node. Upon receiving the grant IE, the requesting node broadcasts a confirm IE to all of its neighboring nodes to notify them of this allocation information.

The reasons for the preceding procedure are clear. First, the minislots of the requesting and peer nodes are already synchronized over the time axis in an IEEE 802.16 mesh network. Second, when the requesting node is transmitting data to the peer node, the peer node must be able to receive the data at the same time. Therefore, the requesting node must negotiate with the peer node to find a range of minislots that is available to both of them (i.e., good for transmitting at the requesting node and good for receiving at the peer node) and can accommodate the requested number of minislots.

Figs. 3 and 4 are two examples showing the effect of the holdoff time value. (Note that these two figures are for illustration purposes only, and the minimum number of slots between subsequent MSH-DSCH messages should be 16, according to the standard.) Fig. 3 shows a good case for establishing a data schedule when the network is not congested. In this case, all control messages required to establish a data schedule are exchanged within one control subframe due to the use of a small holdoff time value. As such, data packets can be quickly transmitted within the same frame in which the control messages are transmitted. In contrast, Fig. 4 shows a bad case for establishing a data schedule when the network is not congested. In this case, the three control messages are transmitted

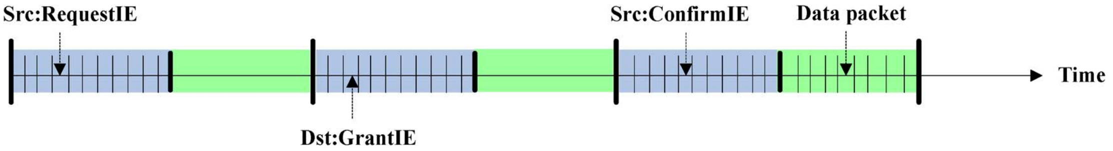


Fig. 4. Bad case for establishing a data schedule in the distributed coordinated scheduling mode when the network is not congested.

over three different frames due to the use of a large holdoff time value. In such a condition, the data packets can only be transmitted over the minislots that are at least two frames away from the transmission of the request IE. Since a node is allowed to transmit data packets to its neighboring node only after they have established a data schedule, the increased delay of the three-way handshake procedure directly degrades the network quality experienced by application programs.

4) IICMTs: We define a new performance metric called *inefficiency index of control message transmissions (IICMT)* to evaluate how fairly network nodes share the control-plane bandwidth. To understand IICMT, one first realizes that, viewed from a node, if every node in its two-hop neighborhood has data to send at any given time, the optimal way to schedule these nodes' control message transmissions is to schedule them roughly in a round-robin fashion. That is, on average, a node should transmit one and only one control message every N transmission opportunities, where N is the number of nodes in its two-hop neighborhood (including itself). We call this round-robin scheme "the static optimal scheme" in this paper. This is the optimal design for a static network in which every node has data to send at all times. This is because, when a node wants to transmit a control message, the transmission must be resolved among all the nodes in its two-hop neighborhood. As such, to avoid congestion while reducing transmission delays, on average, a node can only transmit a control message every N transmission opportunities, where N is as previously defined.

For a fixed-value holdoff time setting scheme, to avoid any congestion from occurring in the network, the maximum of the N values of all nodes should be used as the fixed value for all nodes. Since, in a general network, not all nodes have the same N value, this fixed-value approach will waste the transmission opportunities of the nodes whose N values are smaller than the maximum one.

We define a node's transmission opportunity utilization during a period as the ratio of the number of transmission opportunities that it wins during the period to the total number of transmission opportunities that are available during that period. If such ratios of all nodes under a holdoff time setting scheme closely approximate their counterparts under the static optimal scheme, this holdoff time setting scheme is considered to perform as well as the static optimal scheme.

The following explains the steps used to compute IICMT. First, we convert the actual utilization ratio of node i into the logarithmic form given as follows:

$$R1(i) = -\log_2\left(\frac{\mathrm{NumTxOpp}_{\mathrm{win}}(i)}{\mathrm{NumTxOpp}_{\mathrm{total}}}\right) \tag{10}$$

where $\mathrm{NumTxOpp}_{\mathrm{win}}(i)$ denotes the number of transmission opportunities that node i wins in the period, and $\mathrm{NumTxOpp}_{\mathrm{total}}$ denotes the total number of transmission opportunities that are available in the period. In addition, the logarithmic form of the static optimal utilization ratio for a node i is shown in the following:

$$R2(i) = -\log_2\left(\frac{1}{|nbr(i)|}\right) \tag{11}$$

where $|nbr(i)|$ denotes the number of nodes in node i's two-hop neighborhood. Second, the absolute value of the difference between the two logarithms is computed and given by

$$Diff(i) = AbsoluteValue\left(R1(i) - R2(i)\right). \tag{12}$$

One can use this value to evaluate the inefficiency and unfairness degree of a node's control message scheduling. The best value for this difference is zero, which means that the scheduling generated by the used scheme for this node is equivalent to that generated by the static optimal scheme. If this difference value increases, it means that the scheme performs worse than the static optimal scheme.

Finally, the IICMT metric is defined as the sum of the $Diff$ values of all the nodes in a network and given by

$$\mathrm{IICMT} = \sum_{i=1}^{m} Diff(i) \tag{13}$$

where m is the number of nodes in a network case.

The rationale for this metric is that, when evaluating a holdoff time setting scheme, one should consider its impacts on all network nodes. This sum shows the degree of inefficient and unfair use of available transmission opportunities across all the nodes in a network. A zero IICMT value means that a holdoff time setting scheme schedules transmission opportunities for the whole network as if the static optimal scheme were used. In contrast, a nonzero IICMT value indicates that the scheme either inefficiently or unfairly schedules transmission opportunities when compared with the static optimal scheme. As expected, a high IICMT value indicates that the used holdoff time setting scheme deviates much from the static optimal scheme.

B. Effect of the Holdoff Time Value

We use the NCTUns network simulator [9] to evaluate the effect of the holdoff time value. We use ten random connected topologies and derive average simulation results from them to eliminate the effects that may be caused by using a single specific topology. To generate such topologies, we randomly distribute one BS node and 99 SS nodes within a square area with a side length of 2500 m. We then check whether the generated topology is partitioned or not. If it is partitioned, it is discarded, and the aforementioned process is repeated

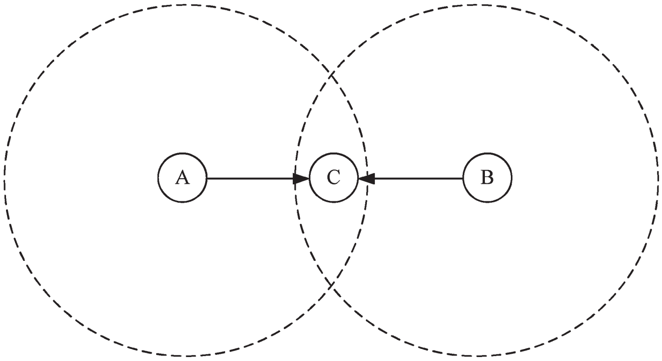


Fig. 5. Example of MSH-NCFG message collisions.

until a random connected topology is generated. The whole process is repeated ten times to generate ten random connected topologies. These topologies represent different random dense wireless backbone networks. For each studied holdoff time setting scheme, we conducted its simulations on each topology five times, each time using a different random number seed. Therefore, we have 50 runs in total to derive average simulation results. The simulated time for each run is set to 1000 s. During simulation, each node runs a MAC-layer pseudo-data scheduler to periodically establish data schedules with its neighboring nodes in a round-robin manner. The frequency is chosen to be one data schedule every 3 s. It generates a moderate traffic load that allows the performances of the studied holdoff time setting schemes to be distinguished.

In [8], we pointed out two reasons the initialization of an IEEE 802.16 mesh network may fail. In this paper, we applied the revised network initialization process proposed in [8] to all studied schemes, including the original fixed-value schemes. This revised process can significantly alleviate the MSH-NCFG message collision problem.

After this revised process is applied, message collisions now only result from excessive MSH-NCFG message transmissions by a new node's neighboring nodes. A typical example is shown in Fig. 5. Suppose that node C is a new node trying to enter the network and nodes A and B are its neighboring nodes that have attached themselves to the network. The dotted circles represent the signal coverage of nodes A and B, respectively. Before node C attaches itself to this network, nodes A and B transmit their own MSH-NCFG messages without considering whether their MSH-NCFG messages can be successfully received at the location of node C. Consequently, many MSH-NCFG messages transmitted by nodes A and B may collide at node C. However, since node C, so far, has not been a functional node in this network, such message collisions do not hinder node C's normal operation at this moment.

However, in case nodes A and B transmit their own MSH-NCFG messages very frequently (for example, these two nodes use a very small holdoff time value to schedule their MSH-NCFG message transmissions), it is very likely that node C cannot successfully receive any MSH-NCFG message transmitted by these two nodes. In such a condition, node C cannot proceed its network initialization process, because it cannot obtain the necessary information required to start its network initialization process.

The other reason that causes the network initialization process to fail is the absence of routing paths from a new SS node to a BS node. For an SS node, the success of its network initialization process relies on the availability of a routing path from itself to the BS node. On performing the registration procedure (one of the necessary procedures in a node's network initialization process), a new node must have a routing path to communicate with a BS node. If no available routing path exists, the new node's network initialization process will fail.

To eliminate the effect of the aforementioned routing problem on the performance results, we adopted a design to guarantee that every new SS node has a sponsor node and every new SS node has a routing path to the BS node. To provide such a guarantee, we first generated routing paths among all nodes using Dijkstra's shortest path algorithm. Then, we let an SS node choose the first-hop node on its routing path to the BS node as its sponsor node. Such a design guarantees that, when an SS node is performing the registration procedure, at least one routing path exists for the SS node to communicate with the BS node. With this design, the problem of network initialization processes failing due to lack of routes to the BS node no longer exists. As such, the simulation results reflect solely the effect of different holdoff time values rather than the mixed effects of different holdoff time values and the used routing protocol.

In the following, we compare four different fixed-value holdoff time setting schemes. As previously mentioned, the standard regulates that the holdoff time base value be set to 4. Therefore, here, we set the holdoff time base value used by all schemes to 4 while varying the holdoff time exponent value used by these schemes from 0 to 3. The resultant holdoff time values are thus 16, 32, 64, and 128, respectively.

Table I shows the performances of the four fixed-value holdoff time setting schemes. In total, 50 runs of simulations were conducted. SRNI, as defined in (4), measures the success rate of network initialization of these 50 runs. The ATOUN-Avg. and ATOUN-Std.dev. in the table are the average and standard deviation of the ATOUNs of the 50 runs, respectively. The ATHPT-Avg. and ATHPT-Std.dev. are the average and standard deviation of the ATHPTs of the 50 runs, respectively. Finally, the IICMT-Avg. and IICMT-Std.dev. are the average and standard deviation of the IICMTs of the 50 runs, respectively.

From the table, one sees that, when the holdoff time value decreases, the ATOUN-Avg. increases, and both the ATHPT-Avg. and the IICMT-Avg. decrease. As discussed before, all of these trends are expected and reasonable. These trends show that using a smaller holdoff time value results in better performances when the network is uncongested.

However, the SRNI results reveal a serious problem when small holdoff time values are used. One sees that using large holdoff time values (e.g., 64 and 128) results in 100% SRNI. However, using small holdoff time values (e.g., 16 and 32) results in a success rate of less than 100%, which means that some SS nodes cannot successfully initialize and attach themselves to the network.

These simulation results show that a fixed holdoff time value cannot provide good scheduling performances while guaranteeing the success of network initialization processes. Based on

TABLE I
PERFORMANCES OF THE FOUR FIXED-VALUE HOLDOFF TIME SETTING SCHEMES

	SRNI	ATOUN		ATHPT (ms)		IICMT	
	Avg.	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
Holdoff Time 16	96%	0.481	0.164	47.423	25.220	95.336	6.083
Holdoff Time 32	98%	0.363	0.165	66.453	15.706	150.262	8.252
Holdoff Time 64	100%	0.237	0.131	104.146	7.518	226.507	10.077
Holdoff Time 128	100%	0.132	0.077	192.719	4.903	318.771	10.629

this observation, we propose a new holdoff time setting scheme to achieve both of the aforementioned goals. In the following, we explain the proposed two-phase holdoff time setting scheme in detail.

IV. TWO-PHASE HOLDOFF TIME-SETTING SCHEME

In Section III-B, we show that using a fixed holdoff time value cannot guarantee the success of network initialization while providing good scheduling performances. As such, we propose a two-phase holdoff time setting scheme that uses different holdoff time values for the two phases. In this scheme, after being powered on, each node initially stays in the network initialization phase. It remains in that phase until all of the nodes in its two-hop neighborhood have successfully initialized and attached themselves to the network. When this condition is met, the node then enters the data transmission phase. Because every node should succeed in initializing and attaching itself to the network, when a node is still in its network initialization phase, the proposed scheme sets its holdoff time to a large value (e.g., 64 or higher) to eliminate the potential hidden terminal problem. This design ensures that the network initialization processes of all nodes will eventually succeed. After all the nodes have successfully attached themselves to the network, they will have switched their phases to the data transmission phase. In this phase, the hidden terminal problem no longer occurs, because the neighbor relationships among all nodes have been known and stabilized. As such, a small holdoff time value can be used in this phase to improve MAC-layer scheduling performances.

In this scheme, a node uses only its local knowledge to determine when to switch from the network initialization phase to the data transmission phase. According to the standard, each node should maintain a node list to record the scheduling information of the nodes in its two-hop neighborhood. This node list is an input to the pseudo-random election algorithm for scheduling the node's control message transmissions. Assume that, for each node, the network operator has given it the total number of nodes in its two-hop neighborhood. (This assumption can be easily met for a static network, where dynamic fading effects are not significant.) With this information, each node can locally determine when it can switch to the data transmission phase by comparing the number of nodes that are currently in its node list with that given by the network operator. When these two numbers match, it can safely enter the data transmission phase for better performances.

In the following, we propose two approaches of this two-phase holdoff time setting scheme. The first one considers the static locations of nodes, whereas the second one further considers the dynamic bandwidth needs of nodes. In the rest of this paper, we call the former the static approach and the latter one the dynamic approach for brevity.

A. Static Approach

For a network node, only nodes in its two-hop neighborhood can contend for transmission opportunities with itself. As such, instead of using the same holdoff time value for all nodes, we propose an approach that allows different nodes to use different holdoff time values based on the node densities around them. The holdoff time value of a node is statically set based on its two-hop neighborhood node number, as shown in

$$\text{Holdoff time of node } j = 2^{\text{floor}(\log_2(|nbr(j)|))} \tag{14}$$

where $nbr(j)$ is as defined in (5).

As shown in Fig. 2, the transmission cycle of an MSH-DSCH message comprises the holdoff and contention times. In a dense network, where a node normally has a large number of nodes in its two-hop neighborhood, a node's holdoff time set by (14) will be large. Due to this reason, the experienced contention time will be small, because each node now must refrain itself from contending for transmission opportunities for a long period of time, which reduces the probability of contention. Since, in this situation, the holdoff time represents a large portion of the transmission cycle, the resultant transmission cycle in the static approach will approach that generated in the static optimal scheme, which only considers the two-hop neighborhood node number without considering the contention time.

Regarding a node in a sparse network, the situation is reversed. In a sparse network, where a node normally has a small number of nodes in its two-hop neighborhood, a node's holdoff time set by (14) will be small. As such, the experienced contention time will be relatively large. This is because each node now only needs to suspend itself for a small period of time, which increases the probability of contention. In this situation, the resultant transmission cycle will be longer than that generated in the static optimal scheme because it has a large contention time in the transmission cycle. As such, the static approach in a sparse network may perform worse than the static optimal scheme. However, as our simulation results will show, the static approach still performs better than any fixed-value holdoff time setting scheme. This is because each node can use a more appropriate holdoff time value in the static approach than in a fixed-value holdoff time scheme.

B. Dynamic Approach

The dynamic approach is based on the static approach and further considers the dynamic bandwidth needs of nodes. Recall

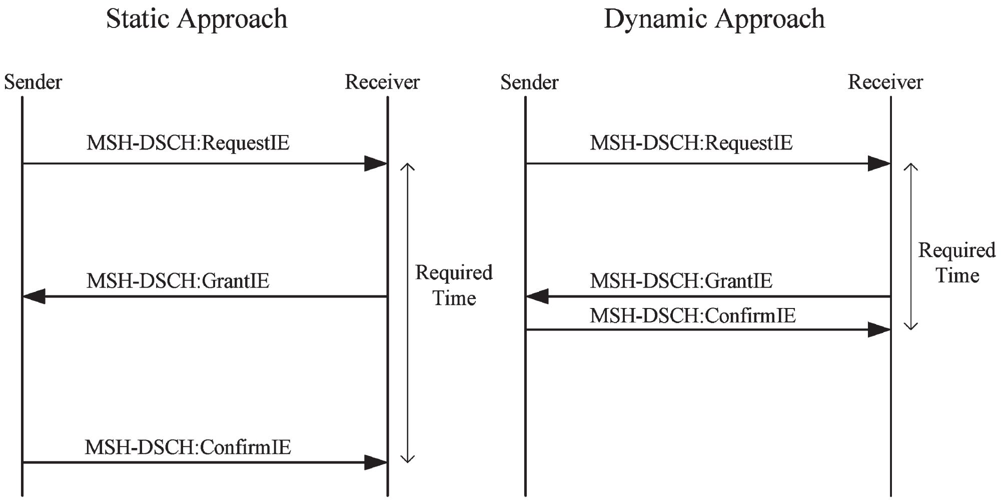


Fig. 6. Comparison between the time required for establishing a data schedule under the static and dynamic approaches of the proposed scheme.

that the three-way handshake procedure used to establish a data schedule requires transmitting three MSH-DSCH messages. When the static approach of the two-phase holdoff time setting scheme is used, the network statically schedules each node's MSH-DSCH message transmissions roughly in a round-robin fashion without considering the dynamic bandwidth needs of nodes. As such, for a node, the transmission cycle between any two consecutive MSH-DSCH message transmissions is fixed, regardless of whether it has data to send. The idea of the dynamic approach is to shorten the transmission cycles of the nodes that have data to send. This will result in decreased per-hop (as well as end-to-end) data transmission delays and increased per-hop (as well as end-to-end) data transmission throughputs.

Using the dynamic approach, a requesting node can reduce the elapsed time between transmitting a request IE and transmitting a confirm IE. On receiving a grant IE from the granting node, a requesting node tries to transmit a confirm IE as soon as possible. As shown in Fig. 6, the dynamic approach can, on average, save about half of the time required to establish a data schedule, compared with the static approach.

The detailed algorithm of the dynamic approach is presented in Fig. 7. In the dynamic approach, the holdoff time base value is set and fixed to 0 rather than the default 4. Initially, the dynamic approach uses the static approach to determine the holdoff time value used for each node. If no node has data to send, the operation of the dynamic approach degenerates to the operation of the static approach. That is, each node uses a holdoff time value determined by the static approach to regularly transmit its MSH-DSCH messages to keep the pseudo-random election algorithm operating correctly. Note that, even though there is no data to send in the network, a node still needs to regularly send out its MSH-DSCH messages to maintain the operation of the pseudo-random election algorithm. The transmitted MSH-DSCH messages are used to notify this node's neighboring nodes of its next MSH-DSCH message transmission time. These MSH-DSCH messages, however, need not carry request, grant, or confirm IEs.

When a node wants to set up a data schedule (i.e., it needs to send out a request IE), the dynamic approach temporarily takes over the task of determining the node's holdoff time value. After the data schedule is set up, this task will be passed back to the static approach to determine the node's holdoff time value for its regular MSH-DSCH message transmissions. The detailed procedure is explained here.

First, after the requesting node sends out the request IE, it calculates the earliest transmission opportunity where it can transmit the confirm IE to the granting node. Transmitting the confirm IE must be performed later than receiving a grant IE from the granting node. To ensure this sequence, the requesting node's target transmission opportunity (i.e., the next transmission opportunity used to transmit the confirm IE) is initially set to the next transmission opportunity of the granting node plus one. (Note that the requesting node knows this information, because this information is regularly exchanged among nodes via the MSH-DSCH messages.) Then, it uses the difference between its current and target transmission opportunities to calculate the maximum target holdoff exponent value using

$$\text{max target holdoff } exp = ceil\left(\log_2(difference)\right). \quad (15)$$

This calculated exponent value is used as the initial exponent value to find a transmission opportunity that is later than the target transmission opportunity. The found transmission opportunity is the output of the pseudo-random election algorithm and will be larger than the target transmission opportunity due to the existence of the contention time. Later on, the dynamic approach then goes through an iteration to find the smallest exponent value that makes the transmission of the confirm IE as close as possible to the reception of the granting IE. During each step of the iteration, the target holdoff time exponent is decremented by one to explore whether this smaller value can still meet the requirement. On the last step of the iteration, where the requirement can no longer be met, the exponent value used in the previous step (which is stored in the optimized holdoff time exponent variable) is the exponent value that is both feasible and the smallest. This value is then returned and used to derive the transmission opportunity to transmit the confirm IE.

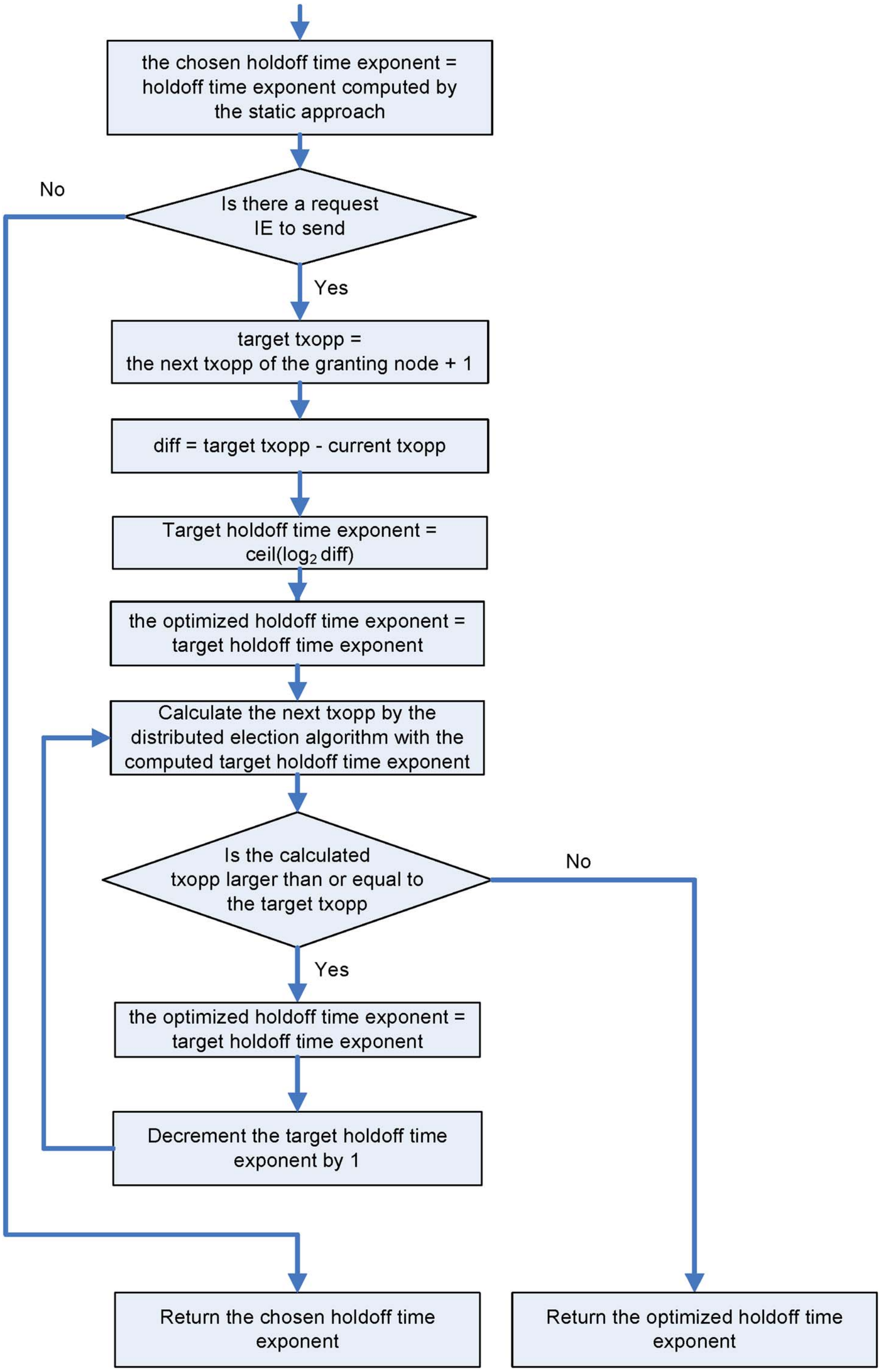
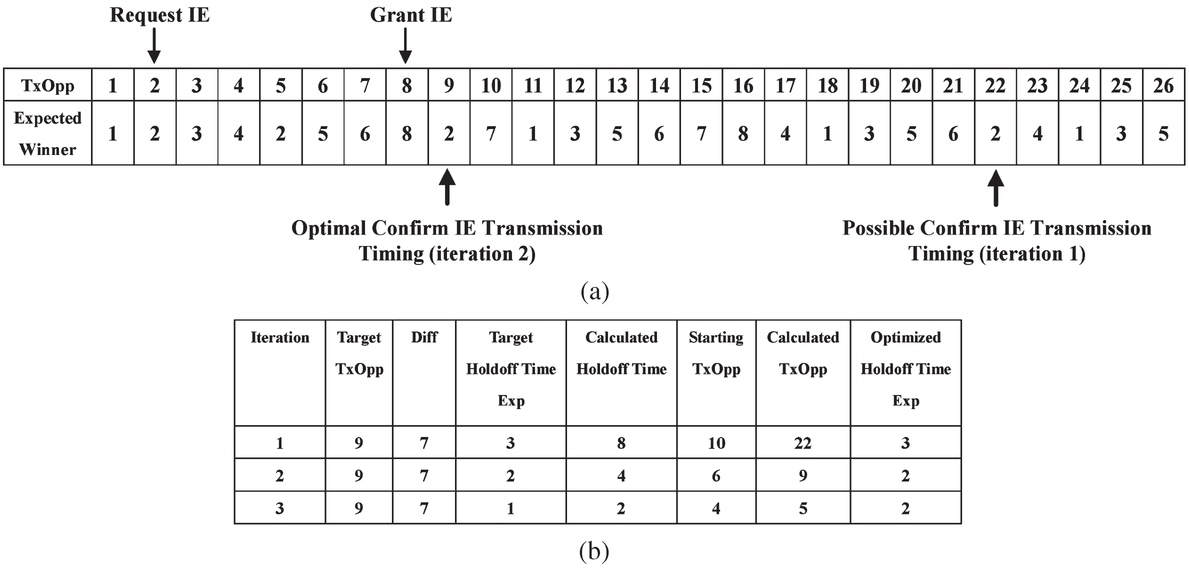

the chosen holdoff time exponent = holdoff time exponent computed by the static approach
Is there a request IE to send
No
Yes
target txopp = the next txopp of the granting node + 1
diff = target txopp - current txopp
Target holdoff time exponent = ceil(log2 diff)
the optimized holdoff time exponent = target holdoff time exponent
Calculate the next txopp by the distributed election algorithm with the computed target holdoff time exponent
Is the calculated txopp larger than or equal to the target txopp
No
Yes
the optimized holdoff time exponent = target holdoff time exponent
Decrement the target holdoff time exponent by 1
Return the chosen holdoff time exponent
Return the optimized holdoff time exponent

Fig. 7. Algorithm of the dynamic approach of the proposed scheme.

Fig. 8 shows an example illustrating the operation of the proposed dynamic approach. Suppose that the two-hop neighborhood of node 2 comprises eight nodes, including node 2 itself. Fig. 8(a) shows the winning nodes of the transmission opportunities numbered from 1 to 26. Here, the winning node of a transmission opportunity is defined as the node that wins this transmission opportunity when all of the eight nodes contend for this transmission opportunity. (This condition will occur if the holdoff time base value is set to zero. The effect of the holdoff time base value will be explained in detail in Section IV-C.) Assume that node 2 wants to establish a data schedule with node 8. Before transmitting the request IE to node 8 on transmission opportunity 2, node 2 should perform the algorithm depicted in Fig. 7. The detailed steps executed by this algorithm are explained here.

As shown in Fig. 8(b), the dynamic approach algorithm first sets the target transmission opportunity to 9, which is right after the transmission opportunity that node 8 is likely to transmit its grant IE (assuming 8 in this example). The algorithm calculates the transmission opportunity that node 2 can win. During the


Request IE
Grant IE
Optimal Confirm IE Transmission Timing (iteration 2)
Possible Confirm IE Transmission Timing (iteration 1)
(a)
(b)

TxOpp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Expected Winner	1	2	3	4	2	5	6	8	2	7	1	3	5	6	7	8	4	1	3	5	6	2	4	1	3	5

Iteration	Target TxOpp	Diff	Target Holdoff Time Exp	Calculated Holdoff Time	Starting TxOpp	Calculated TxOpp	Optimized Holdoff Time Exp
1	9	7	3	8	10	22	3
2	9	7	2	4	6	9	2
3	9	7	1	2	4	5	2

Fig. 8. Example illustrating the operation of the dynamic approach.

first iteration, the algorithm first finds that node 2 can win transmission opportunity 22. Since the calculated transmission opportunity 22 is still larger than the target transmission opportunity, the algorithm stores the current target holdoff time exponent value into the optimized holdoff time exponent variable, decrements the target holdoff time exponent value by one, and starts the second iteration. During the second iteration, it finds that node 2 can win transmission opportunity 9. However, because the calculated transmission opportunity 9 is still larger than or equal to the target transmission opportunity, it enters the third iteration to probe further optimization. During the third iteration, the algorithm finds that the calculated transmission opportunity is 5, which is now less than the target transmission opportunity. Thus, it stops this iterative procedure and returns the value stored in the optimized holdoff time exponent variable as its output. (Note that value is the target holdoff time exponent value calculated in the previous iteration.) As shown in Fig. 8(b), upon performing the pseudo-random election algorithm with the optimized holdoff time exponent value (2 in this example), node 2 will win transmission opportunity 9, which is the optimal transmission timing to transmit its confirm IE in this example.

C. Discussion on the Effect of Holdoff Time Base Value

As mentioned before, the IEEE 802.16 standard states that every node should set the holdoff time base value to 4. For this reason, IEEE 802.16-compliant devices can cooperate using this fixed holdoff time base value. Both of the static and dynamic approaches, however, require changing the holdoff time base value to successfully operate. To make these two approaches compatible with the IEEE 802.16 standard, a mechanism is required to notify network devices of changes to the holdoff time base value. Here, we explain the problems that may occur if this system parameter is dynamically changed. In Section IV-D, we will describe several mechanisms that can be used to change this parameter without causing problems.

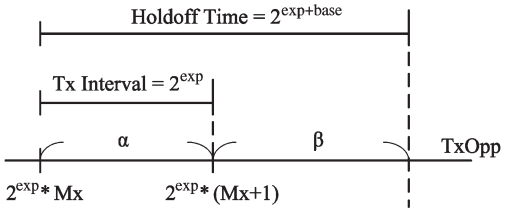

Holdoff Time = $2^{\text{exp+base}}$
Tx Interval = 2^{exp}
α
β
TxOpp
2^{exp}* Mx
2^{exp}* (Mx+1)

Fig. 9. Relationship between the holdoff time and the Tx interval.

The holdoff time base value contributes a constant time amount 2^{base} to the holdoff time. Setting the holdoff time base value to 4 means that each node should suspend its contention for at least 2^4 consecutive transmission opportunities after it has won one. This lower bound limits the smallest holdoff time value that can be assigned to nodes in the static and dynamic approaches. As such, the static and dynamic approaches may not achieve their optimal performances if the holdoff time base value is not reduced to 0. For this reason, the holdoff time base value is set to 0 in the static and dynamic approaches.

Instead of using a lengthy field, the standard defines that each node should use two shorter fixed-length fields exp and Mx to represent a transmission opportunity number. The relationship between these two fields and a transmission opportunity number has been given in (3). On receiving a control message (such as an MSH-NCFG or an MSH-DSCH message), a node should use the received exp and Mx fields to derive the next transmission interval of the transmitting node using (3). Fig. 9 shows the relationship between the holdoff time and the transmission interval (Tx interval). The holdoff time comprises the Tx interval (denoted as α in the figure) and the ineligible interval (denoted as β in the figure). The Tx interval represents the duration in which a node may contend for one transmission opportunity. On the other hand, the ineligible interval is the duration for which the node is not allowed to contend for any transmission opportunity. Based on (3) defined in the standard, the length of the Tx interval is fixed at 2^{exp}, because a node's next transmission opportunity number is within a fixed-length interval ranging from $(2^{exp} * Mx + 1)$ to $2^{exp} * (Mx + 1)$. Consequently, the length of the ineligible interval is $(2^{exp+base} - 2^{exp})$. If the base value is set to 0, the holdoff time and the Tx interval of each node will exactly overlap, causing the length of the ineligible interval to be zero. This means that, after winning a transmission opportunity, a node will immediately contend for another transmission opportunity. (This also means that a node will contend for transmission opportunities all the time.) As such, a node should consider that all the nodes in its two-hop neighborhood will contend for each transmission opportunity with itself. On the other hand, if the holdoff time base value is larger than 0, a node's holdoff time will be larger than its Tx interval. In such a condition, the contention time experienced by a node can be reduced due to a decreased number of contending nodes.

The choice of the holdoff time base value depends on the needs of a holdoff setting scheme. For the static approach, using a positive holdoff time base value can reduce the contention time of each node. For the dynamic approach, however, the holdoff time base value must be set to zero for the following two important reasons: First, if a positive holdoff time base value is used, the lower bound of the holdoff time value that can be assigned to nodes will be limited. As such, to give the dynamic approach the largest freedom to set the holdoff time of a node, the holdoff time base value should preferably be set to 0 at all times. Second, if the holdoff time base value is allowed to change during the operation of a network, after a node's holdoff time has just been changed (due to the change of the holdoff time base value), MSH-DSCH control messages may collide. The reason for this phenomenon is explained in detail here.

Figs. 10 and 11 show two cases after a node's holdoff time value has just been changed. The former shows an example in which changing the holdoff time value results in no message collisions, whereas the latter case shows an opposite example. Suppose that node A has changed its holdoff time value and broadcast the new holdoff time exponent value. In Fig. 10, node B is the next node to transmit an MSH-DSCH message. In such a case, node C will be notified of this change by node B's MSH-DSCH message in time. Thus, node C will not schedule its MSH-DSCH message transmission to collide with node A's MSH-DSCH message transmission. In contrast, in Fig. 11, node C has scheduled an MSH-DSCH message transmission before node B can notify it of node A's new holdoff time value. Under such a condition, node C's MSH-DSCH message transmission may collide with node A's MSH-DSCH message transmission, because node A's ineligible interval viewed by node C is now out of date. Figs. 12–14 illustrate three cases that can cause this problem.

In these figures, HTa denotes the holdoff time of node A viewed by node C (may be out of date), and HTa$'$ denotes the holdoff time of node A viewed by node A itself (always up to date). The symbol γ denotes the vulnerable interval that results from node A's changing its holdoff time and during which the MSH-DSCH messages of nodes A and C may collide. Suppose that node C's transmission was scheduled within node A's original ineligible interval β. Fig. 12 depicts a case where node A has just decreased its holdoff time base value; therefore, its ineligible interval is shortened

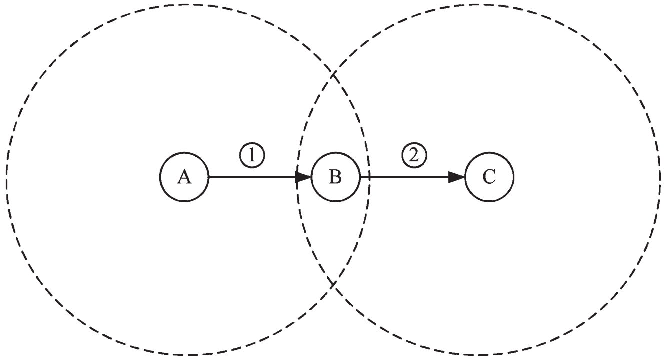


Fig. 10. Example showing that control messages will not collide after node A changes its holdoff time value.

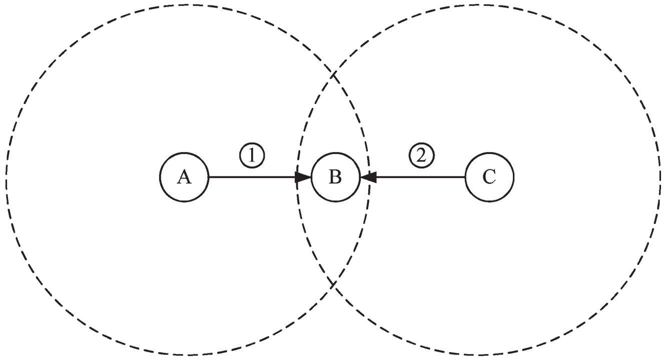


Fig. 11. Example showing that control messages will collide after node A changes its holdoff time value.

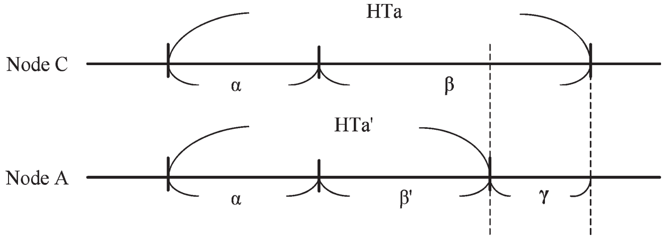


Fig. 12. Case where node A has just decreased its holdoff time base value.

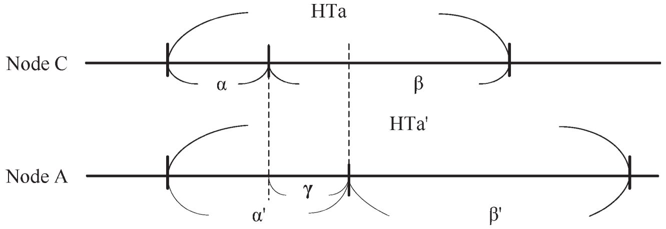


Fig. 13. Case where node A has just increased its holdoff time exponent value.

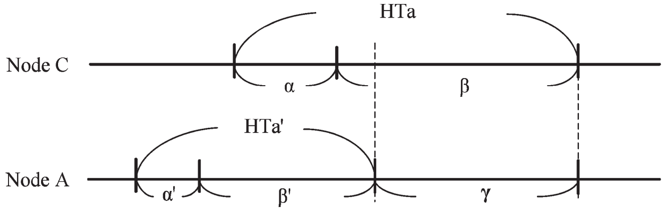


Fig. 14. Case where node A has just decreased its holdoff time exponent value.

from β to β'. This operation generates the γ vulnerable interval, because node C does not know that node A will now contend for transmission opportunities in the γ interval. Fig. 13 depicts a case where node A has just increased its holdoff time exponent value; therefore, both its Tx interval and ineligible interval are lengthened. In such a condition, node A's ineligible interval will shift on the time axis, generating the vulnerable interval shown in Fig. 13. In contrast, Fig. 14 depicts a case where node A has just decreased its holdoff time exponent value; therefore, its Tx interval and ineligible interval are shortened, resulting in the shift of node A's ineligible interval on the time axis. This shift generates the vulnerable interval shown in Fig. 14.

To prevent the collision problem from occurring, an additional mechanism is necessary. For instance, in the case given in Fig. 11, after changing the holdoff time value, node A should defer its contention for transmission opportunities until its original ineligible interval has elapsed. Such a mechanism, however, may increase the implementation complexity of the dynamic approach and decrease the scheduling performances. To totally avoid the collision problem without wasting much network bandwidth, the dynamic approach uses 0 as the holdoff time base value for all network nodes at all times. Fixing the holdoff time base value to zero effectively eliminates every node's ineligible interval (i.e., the length of each node's ineligible interval now becomes zero), resulting in each network node considering that it should always contend for transmission opportunities with all other nodes in its two-hop neighborhood. (These two-hop neighborhood nodes are considered to be always eligible to contend for transmission opportunities.) In such a condition, if a node wants to win a transmission opportunity, it should win over all of its two-hop neighborhood nodes. This means that, for each node, the node list used as the input of its pseudo-random election algorithm will always comprise its two-hop neighborhood nodes, despite the dynamic changes of the holdoff time exponent values of its neighboring nodes. As such, packet collisions due to dynamic changes of the holdoff time exponent values can be avoided under the zero-holdoff-time-base-value condition.

D. Notification of Holdoff Time Base Value Change

Here, we propose a practical protocol that can be used to notify new SS nodes of the holdoff time base value used in a network. A BS node can use this protocol to check whether a new SS node can use a holdoff time base value other than 4. If the SS node cannot do so, the BS node should reject the network registration request (REG-REQ) from such an SS node, because this SS node cannot work well with other SS nodes in such a network.

First, the reserved "*vendor-specific information*" field can be exploited to help SS nodes know the holdoff time base value used in a network. This field is defined in the standard for the registration procedure to exchange additional information not specified in the standard. The signaling protocol is described as follows: An SS node first adds a holdoff time base query message (carried by the "*vendor-specific information*" field) into the REG-REQ message, which is destined to the BS node. On receiving this REG-REQ message, the BS node replies the SS node with a registration response (REG-RSP) message, which contains the holdoff time base value used in this network (also carried by the "*vendor-specific information*" field). If the BS node does not find the holdoff time base query message in the SS node's REG-REQ message, it should reject this SS node's REG-REQ, because this SS node may not be able to change its holdoff time base value.

TABLE II
PARAMETER SETTING USED IN SIMULATIONS

Parameter Name	Value
MSH-CTRL-LEN	8
MSH-DSCH-NUM	8
Scheduling Frames	2
Requested Mini-slot Size	20
Requested Frame Length	32
Modulation/Coding Scheme	64QAM-3/4
Maximum Transmission Range	500 meter
Frame Duration	10 ms

There are three ways to reject a REG-REQ: The first one is to simply ignore the REG-REQ message if the BS node decides to reject it. The second way is to utilize the "de/reregister command" (DREG-CMD) message defined in the standard. The DREG-CMD message can be used to notify the SS node of the rejection action. Finally, the third way is to return a REG-RSP message with the response code set to 1, indicating that this REG-REQ cannot be accepted, because the SS node does not support the dynamic approach.

V. Performance Evaluation

In this section, we evaluate the performances of the static and dynamic approaches. We compare the simulation results of these approaches with those of the three fixed-value holdoff time setting schemes studied in Section III-B (the "holdoff time 16, 32, and 64" schemes). For the evaluation, the NCTUns network simulator and emulator [9] is used, which was used in Section III-B to generate the simulation results of the three fixed-value holdoff time setting schemes. The chain, grid, and random network topologies are used for these performance studies. Each reported performance is the average of five runs using different random number seeds. For each run, the simulated time is 1000 s. Table II shows the parameter setting used in our simulations. More detailed setup specific to a particular network topology will be described in the following sections.

A. Chain Network Topology

The chain network topology is composed of 21 nodes. From left to right, the nodes are named BS and SS(1), SS(2), ..., SS(20), respectively. On this chain network, each node runs a MAC-layer pseudo-data scheduler to periodically establish data schedules with its neighboring nodes in a round-robin manner. The frequency is chosen to be one data schedule every 3 s, which has been explained before.

As shown in Table III(a), in the chain network, as the holdoff time value exponentially increases, the ATOUN value exponentially decreases, and the ATHPT value exponentially

TABLE III
PERFORMANCES OF THE THREE FIXED-VALUE HOLDOFF TIME SETTING SCHEMES AND THE STATIC AND DYNAMIC APPROACHES OF THE PROPOSED TWO-PHASE HOLDOFF TIME SETTING SCHEME. (a) CHAIN NETWORK TOPOLOGY. (b) GRID NETWORK TOPOLOGY. (c) RANDOM NETWORK TOPOLOGY

(a)

	MAC						Application					
	ATOUN		ATHPT (ms)		IICMT		TCP (KB/sec)		UDP (KB/sec)		Ping (ms)	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.	Avg	Std. dev.
Dynamic	0.641	0.114	14.295	12.236	8.292	0.335	191.281	158.249	506.640	142.675	370.878	231.555
Static	0.494	0.083	17.389	14.997	18.140	3.553e-15	186.503	160.992	504.664	142.176	425.711	260.078
Holdoff Time 16	0.210	0.033	24.184	5.428	46.271	2.132e-14	157.815	154.438	434.321	156.821	829.461	459.401
Holdoff Time 32	0.110	0.018	46.285	10.842	66.365	1.421e-14	119.598	136.690	362.390	158.164	1527.291	845.504
Holdoff Time 64	0.057	0.009	89.256	19.483	86.929	1.499e-3	71.003	101.183	221.628	151.025	2864.634	1594.184

(b)

	MAC					
	ATOUN		ATHPT (ms)		IICMT	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
Dynamic	0.725	0.086	43.398	9.430	9.661	0.146
Static	0.559	0.082	50.467	11.109	59.303	0.000
Holdoff Time 16	0.530	0.090	50.983	8.211	70.313	0.000
Holdoff Time 32	0.405	0.083	62.852	5.838	116.974	0.011
Holdoff Time 64	0.245	0.059	104.479	3.640	199.009	0.003

(c)

	MAC					
	ATOUN		ATHPT (ms)		IICMT	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
Dynamic	0.718	0.129	47.425	25.838	14.896	0.594
Static	0.543	0.112	55.312	31.847	66.718	1.903
Holdoff Time 16	0.481	0.164	55.194	23.640	95.336	6.083
Holdoff Time 32	0.363	0.165	66.453	15.706	150.262	8.252
Holdoff Time 64	0.237	0.131	104.146	7.518	226.507	10.077

increases. These results show that when the holdoff time value exponentially increases, the average transmission opportunity utilization significantly decreases, and the ATHPT significantly increases. The reasons for these phenomena have been explained before. As for the static and dynamic approaches, one sees that they significantly outperform the three fixed-value holdoff time setting schemes on the ATOUN and ATHPT metrics. One also sees that the dynamic approach outperforms the static approach. This is because the former can dynamically adjust the holdoff time value to reduce the time interval between sending a request IE and sending a confirm IE. As such, the time required to complete a three-way handshake procedure (and, thus, for establishing a data schedule) can be greatly reduced. This also explains why the dynamic approach generates a higher utilization of transmission opportunity than the static approach.

For IICMT, when the holdoff time value decreases, the IICMT value decreases as well. This result shows that using a smaller holdoff time value can achieve fairer and more efficient scheduling. One sees that the static and dynamic approaches achieve much smaller IICMT values than the fixed-value holdoff time setting schemes. The result is expected as in both approaches; the holdoff time of each node can be independently set to a different value to reflect the node density around it. In contrast, as discussed before, a fixed-value holdoff time setting scheme cannot suit the scheduling needs of all the nodes in a network. The dynamic approach outperforms the static approach on IICMT, and the reason is explained here. To reduce the time required for the three-way handshake procedure, the dynamic approach uses an iterative algorithm to decrease a node's holdoff time value. As such, the dynamic approach eliminates a part of the contention time that the static approach cannot eliminate. This makes the dynamic approach perform more closely to the static optimal scheme than the static approach.

Regarding application performances, on this chain network, we conduct a different set of simulations using three different application-layer traffic: 1) Transmission Control Protocol (TCP); 2) User Datagram Protocol (UDP); and 3) ping. For a studied holdoff time setting scheme, its performances are evaluated using 18 cases. Each case was run five times, each time using a different random number seed. In each case, a traffic flow (either TCP, UDP, or ping) is set up. The source node of the traffic flow is fixed at the SS(2) node, whereas the destination node of the traffic flow is chosen to be SS$(i + 2)$ in the ith case.

Fig. 15 shows the relationship between the TCP throughput and the hop count, Fig. 16 shows the relationship between the UDP throughput and the hop count, and Fig. 17 shows the relationship between the end-to-end RTT measured by the ping program and the hop count, respectively. As shown in Figs. 15 and 16, the static and dynamic approaches achieve much higher throughputs than the fixed-value holdoff time setting schemes over all studied hop counts. The RTT results show that the

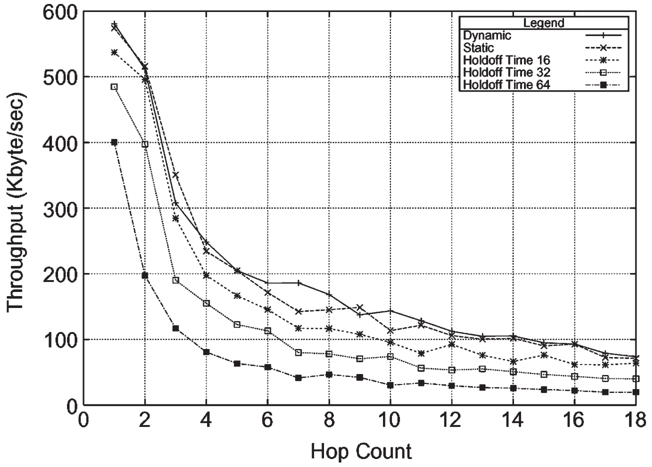


Fig. 15. TCP throughputs over different hop counts in chain networks.

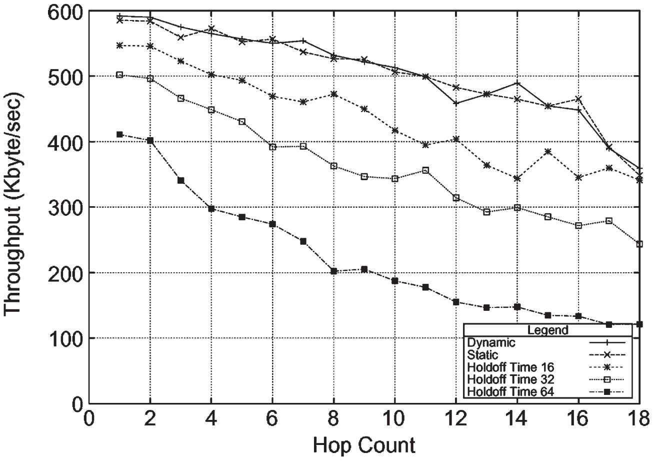


Fig. 16. UDP throughputs over different hop counts in chain networks.

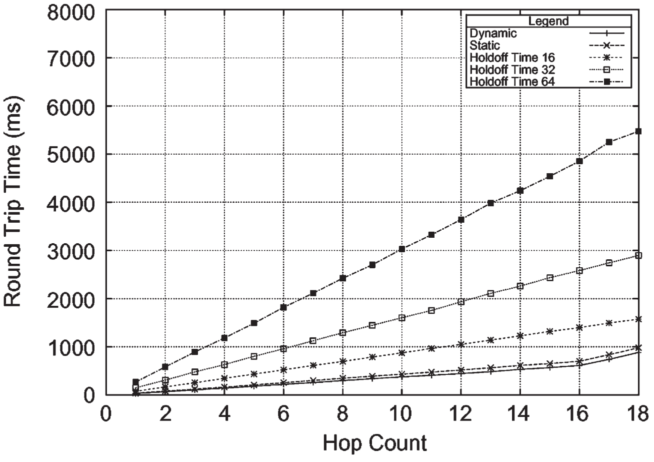


Fig. 17. RTT measured by the ping program in chain networks.

static and dynamic approaches significantly reduce the end-to-end round-trip packet delay, when compared with the three fixed-value holdoff time setting schemes. The results also show that the dynamic approach generates a smaller round-trip packet delay than the static approach. This is expected as the dynamic approach can reduce the time required to establish a data schedule further than the static approach.

Table III(a) shows the TCP and UDP throughputs and RTTs averaged across all different hop counts for each scheme. According to the average TCP and UDP throughput results, the static and dynamic approaches, on average, achieve higher TCP and UDP throughputs than the fixed-value schemes. For example, the dynamic approach outperforms the "holdoff time 16" scheme by a factor of 1.21 on the TCP throughput and by a factor of 1.16 on the UDP throughput, respectively. Regarding the RTT, the dynamic approach, on average, reduces the RTT of "ping" packets by a factor of 2.24 when compared to the "holdoff time 16" scheme.

B. Grid Network Topology

For grid network simulations, we construct a 10 × 10 grid network comprising 100 nodes, each of which is spaced 450 m apart from its vertical and horizontal neighbors. Each node runs a MAC-layer pseudo-data scheduler to periodically establish data schedules with its neighboring nodes in a round-robin manner. As previously explained, the frequency is chosen to be one data schedule every 3 s.

As shown in Table III(b), the ATOUN result shows that the dynamic approach achieves the highest utilization of transmission opportunity. The ATHPT result shows that the dynamic approach, on average, generates the shortest time required for establishing data schedules among all studied schemes. Regarding IICMT, the dynamic approach, on average, achieves 9.661, which is smaller than the "holdoff time 16" scheme by a factor of 7.278. The IICMT results show that the dynamic approach can both efficiently and fairly schedule transmission opportunities in the control plane. As for the static approach, it, on average, achieves a better IICMT value than the "holdoff time 16" scheme. However, its performances on ATOUN and ATHPT are close to those of the "holdoff time 16" scheme because in a grid network, a node's two-hop neighborhood node number is, on average, more than 16, and almost every node (except the nodes on the edges of the grid) has the same number. This condition allows the "holdoff time 16" scheme to perform almost equally well with the static approach.

C. Random Network Topology

For random network simulations, we use the ten random topologies generated in Section III-B to compare the performances of all studied holdoff time setting schemes in general networks. The simulation setting used here is the same as that used in Section III-B.

As shown in Table III(c), in random topologies, both the static and dynamic approaches generate better performances than the fixed-value holdoff time setting schemes. For the static approach, it, on average, increases the ATOUN value by a factor of 1.128 and decreases the IICMT value by a factor of 1.428, when compared with the "holdoff time 16" scheme. As for the dynamic approach, it, on average, increases the ATOUN value by 49.27% and decreases the IICMT value by 640.01%, as compared with the "holdoff time 16" scheme. These results show that both approaches generate fairer and more efficient

MAC-layer scheduling for the network than fixed-value holdoff time setting schemes.

One sees that the static approach's ATHPT value is close to that of the "holdoff time 16" scheme. This situation is similar to that in the grid network case. In these random topologies, the number of nodes in each node's two-hop neighborhood is, on average, more than 16. As such, in such a condition, setting all the nodes' holdoff time values to 16 is sufficient to decrease the time required for the three-way handshake procedure. As in the chain and grid networks, one also sees that the dynamic approach reduces the ATHPT value further than the static approach.

In summary, these simulation results show that, under most network topologies, the dynamic approach 1) utilizes the control-plane bandwidth efficiently; 2) reduces the time required to establish data schedules significantly; and 3) utilizes the control-plane bandwidth fairly.

VI. Conclusion

In an IEEE 802.16 mesh network, the holdoff time value setting design is very important to achieve good scheduling performances in the distributed coordinated scheduling mode. In this paper, we show that using a small value for this parameter can improve MAC-layer performances, which are quantified by three performance metrics: 1) ATOUN; 2) ATHPT; and 3) IICMT. However, we show that doing so can easily cause a node's network initialization process to fail in a dense network. In this paper, we further explain why using a fixed holdoff time value for all nodes, regardless of their node densities and dynamic bandwidth needs, can result in suboptimal performances.

To address these problems, we propose a two-phase holdoff time setting scheme to guarantee the success of network initialization and improve MAC-layer scheduling performances. Both a static approach and a dynamic approach of this scheme are proposed, and their performances are studied and compared in this paper.

The overall simulation results show that the dynamic approach significantly outperforms all fixed-value schemes. The ATOUN and ATHPT results show that it significantly increases the utilization of the control-plane bandwidth and decreases the time required for completing the three-way handshake procedure. The IICMT results show that it generates efficient and fair scheduling in the distributed coordinated scheduling mode. In addition, the throughput results show that it generates the highest TCP and UDP throughputs among all studied schemes. Finally, the round-trip time results show that the dynamic approach generates the shortest end-to-end round-trip packet delay among all studied schemes.

Appendix

In this Appendix, we use an example to explain why the centralized scheduling mode has difficulty in exploiting spatial reuse of wireless communication. Suppose that a cluster has one BS node and two SS nodes, which are assigned node IDs 1, 2, and 3, respectively. In this cluster, a 20-Mb/s bandwidth is reserved for the centralized scheduling mode. For simplicity, assume that one minislot corresponds to a 1-Mb/s bandwidth under a specific modulation/coding scheme combination. As such, the bandwidth allocation problem is equivalent to the problem of allocating 20 minislots, numbered from 1 to 20, among the three nodes.

A bandwidth allocation message issued by the BS node have three entries—one for each node. Suppose that the three nodes are allocated a bandwidth of 10, 5, and 5 Mb/s, respectively. Each node can find its own entry using its node ID as the index and obtain the contents of other entries. As such, each node can know the (start minislot offset, end minislot offset) allocated for itself. For example, node 3 knows that nodes 1 and 2 are allocated 10 and 5 Mb/s, respectively, which are equivalent to 10 and 5 minislots, respectively. As such, it knows that the (start minislot offset, end minislot offset) for node 1, node 2, and itself are (1, 10), (11, 15), and (16, 20), respectively. Nodes 1 and 2 can also derive the same minislot offsets by themselves.

Allocating minislots using the preceding method is simple and guarantees collision-free transmissions, because there is no overlapping. This method, however, leads to a very low bandwidth utilization of a wireless network, because no spatial reuse is exploited. However, this problem cannot be helped. The message format defined in the centralized scheduling mode only specifies bandwidth and does not allow a BS node to inform each SS node of its start and end minislot offsets. Without such offset information, each SS node has to take a conservative (nonoverlapping) bandwidth allocation approach, making spatial reuse hard to exploit.

In the literature, several studies have been proposed to address this problem. In [10], Cheng *et al.* proposed a scheme that combines the centralized and distributed scheduling modes to increase the utilization of the control-plane bandwidth. Using this scheme, the distributed scheduling mode can use the control-plane bandwidth left by the centralized scheduling mode and thus utilizes the control-plane bandwidth more efficiently. In [11], Chen *et al.* assumed that each node is equipped with two directional antennas. With this assumption, they propose a scheduling algorithm to increase the utilization of the spectrum. In [12]–[14], interference-aware scheduling algorithms and routing-tree construction algorithms are proposed to achieve concurrent transmissions in the centralized scheduling mode.

Although these papers propose algorithms to increase the concurrence of transmissions in the centralized scheduling mode, they do not propose new methods for SS nodes to convert bandwidth setting into minislot allocations. As a result, to make concurrent transmissions possible, nodes that may interfere with each other will need to be suspended (be allocated a bandwidth of 0 Mb/s), except the one that is chosen to be active. Under such a condition, the period of issuing control messages containing scheduling information greatly influences the delays experienced by SS nodes. For example, if the issuing period is too long, SS nodes whose transmissions are suspended by the BS node will experience a very long delay before they can transmit data packets. So far, an efficient method to convert bandwidth setting into minislot allocations for the centralized scheduling mode still needs to be studied.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments, which significantly improved the quality of this paper.

REFERENCES

[1] *IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std. 802.16-2004 (Revision of IEEE Std. 802.16-2001).

[2] M. Cao, V. Raghunathan, and P. R. Kumar, "A tractable algorithm for fair and efficient uplink scheduling of multihop WiMax mesh networks," in *Proc. 2nd IEEE Workshop WiMesh*, Reston, VA, Sep. 25, 2006, pp. 101–108.

[3] D. Kim and A. Ganz, "Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems," in *Proc. 2nd Int. Conf. IEEE Broadnets*, Boston, MA, 2005, pp. 833–839.

[4] H. Shetiya and V. Sharma, "Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks," in *Proc. 1st ACM Workshop WMuNeP*, Montreal, QC, Canada, Oct. 2005, pp. 140–149.

[5] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu, "Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode," in *Proc. 6th ACM Int. Symp. MobiHoc*, Urbana-Champaign, IL, May 25–27, 2005, pp. 78–89.

[6] N. Bayer, D. Sivchenko, B. Xu, V. Rakocevic, and J. Habermann, "Transmission timing of signaling messages in IEEE 802.16 based mesh networks," in *Proc. Eur. Wireless*, Athens, Greece, Apr. 2006.

[7] N. Bayer, B. Xu, J. Habermann, and V. Rakocevic, "Improving the performance of the distributed scheduler in IEEE 802.16 mesh networks," in *Proc. IEEE VTC—Spring*, Dublin, Ireland, Apr. 2007, pp. 1193–1197.

[8] S. Y. Wang, C. C. Lin, K. H. Fang, and T. W. Hsu, "Facilitating the network entry and link establishment processes of IEEE 802.16 mesh networks," in *Proc. IEEE WCNC*, Hong Kong, Mar. 11–15, 2007, pp. 1842–1847.

[9] S. Y. Wang, C. L. Chou, C. H. Huang, C. C. Hwang, Z. M. Yang, C. C. Chiou, and C. C. Lin, "The design and implementation of the NCTUns 1.0 network simulator," *Comput. Netw.*, vol. 42, no. 2, pp. 175–197, Jun 2003 (available at http://NSL.csie.nctu.edu.tw/nctuns.html).

[10] S.-M. Cheng, P. Lin, D.-W. Huang, and S.-R. Yang, "A study on distributed/centralized scheduling for wireless mesh network," in *Proc. IWCMC*, Vancouver, BC, Canada, Jul. 3–6, 2006, pp. 599–604.

[11] J. Chen, C. Chi, and Q. Guo, "A bandwidth allocation model with high concurrence rate in IEEE 802.16 mesh mode," in *Proc. Asia-Pacific Conf. Commun.*, Perth, Western Australia, Oct. 3–5, 2005, pp. 750–754.

[12] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, "Interference-aware IEEE 802.16 WiMax mesh networks," in *Proc. IEEE 61st VTC—Spring*, May 30–Jun. 1, 2005, vol. 5, pp. 3102–3106.

[13] J. Tao, F. Liu, Z. Zeng, and Z. Lin, "Throughput enhancement in WiMax mesh networks using concurrent transmission," in *Proc. Int. Conf. WCNM*, Wuhan, China, Sep. 23–26, 2005, vol. 2, pp. 871–874.

[14] L. Fu, Z. Cao, and P. Fan, "Spatial reuse in IEEE 802.16 based wireless mesh networks," in *Proc. IEEE ISCIT*, BUPT, Beijing, China, Oct. 12–14, 2005, vol. 2, pp. 1358–1361.

Shie-Yuan Wang (SM'06) received the M.S. and Ph.D. degrees in computer science from Harvard University, Cambridge, MA, in 1997 and 1999, respectively.

He is currently an Associate Professor with the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C. He was the author of a network simulator called NCTUns, which is now a famous tool widely used all over the world. His research interests include wireless networks, Internet technologies, network simulations, and operating systems.

Chih-Che Lin (S'06) received the B.S. degree in computer science and information engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 2002. He is a currently working toward the Ph.D. degree with the Department of Computer Science, NCTU.

He is a core team member of the NCTUns network simulator project. His current research interests include wireless networking, wireless mesh networks, vehicular networks, intelligent transportation systems, and network simulation.

Han-Wei Chu received the B.S. and M.S. degrees in computer science from National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 2004 and 2006, respectively.

During 2004 and 2006, he was a core team member of the NCTUns network simulator project. He is currently with the Department of Computer Science, NCTU. His research interests include computer networking and network simulation.

Teng-Wei Hsu received the B.S. degree in electrical engineering from Chang Gung University, Taoyuan, Taiwan, R.O.C., in 2005 and the M.S. degree in network engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2007.

During 2005 and 2007, he was a core team member of the NCTUns network simulator project. He is currently with the Department of Computer Science, NCTU. His research interests include wireless mesh networks and network simulation.

Ku-Han Fang received the M.S. degree in network engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 2007.

During 2005 and 2007, he was a core team member of the NCTUns network simulator project. In January 2008, he joined Cyberlink Corporation, Taipei, Taiwan. His current research interests include wireless mesh networks, satellite network protocols, and network simulation.