21

Holdoff Algorithms for IEEE 802.16 Mesh Mode in Multi-hop Wireless Mesh Networks

Bong Chan Kim[1] and Hwang Soo Lee[2]
[1]*Samsung Electronics*
[2]*KAIST*
Republic of Korea

1. Introduction

Multi-hop wireless mesh networks (M-WMNs) [Akyildiz, I. F., 2005] are one of the key features of beyond 3G systems because of their flexibility and low-cost deployment. So far, most of existing studies on multi-hop wireless mesh networks have been accomplished based on the IEEE 802.11 ad hoc mode. The IEEE 802.16 working group (WG) specified the IEEE 802.16-2004 standard [IEEE Std. 802.16-2004, 2004] in October 2004 and the standard defined two modes: the point-to-multi-point (PMP) mode and the mesh mode. The IEEE 802.16 mesh standard defines three mechanisms to schedule the data transmission: centralized scheduling (CSCH) [Morge, P. S., 2007], [Han, B., 2007], coordinated distributed scheduling (C-DSCH) [Morge, P. S., 2007], and uncoordinated distributed scheduling (Un-DSCH). In the IEEE 802.16 mesh mode with the CSCH, C-DSCH, and Un-DSCH, multi-hop communication is possible between nodes such as mesh base stations (MeshBSs) and mesh subscriber stations (MeshSSs) because all nodes are peers and each node can act as routers to support multi-hop packet forwarding. In particular, in the IEEE 802.16 mesh mode with the C-DSCH, every node competes for channel access using a distributed election algorithm (DEA) based on the scheduling information of the extended neighborhoods (one-hop and two-hop neighbors) in a completely distributed manner and reserves radio resource by a three-way handshaking mechanism in which nodes request, grant, and confirm available radio resource using mesh distributed scheduling (MSH-DSCH) message. Like this, because the IEEE 802.16 mesh mode with the C-DSCH has good flexibility and scalability, it is suitable as an alternative medium access control (MAC) protocol for establishing M-WMNs.
For M-WMNs to serve as a wireless network infrastructure, the protocol design for M-WMN should target a high network throughput. In the IEEE 802.16 mesh mode with the C-DSCH, after occupying radio resource, a node cannot transmit any MSH-DSCH message for a holdoff time in order to share radio resource with other nodes in M-WMN. If nodes get a short holdoff time in a heavily loaded network situation, the competition between nodes will happens severe and thus they will experience long contention times before reserving radio resource. On the other hand, if nodes get a long holdoff time in a lightly loaded

[1] This work was performed while the first author was a Ph.D. student.

network situation, radio resource will be wasted unnecessarily. Like this, network throughput has a close relationship with the performance of holdoff algorithm.

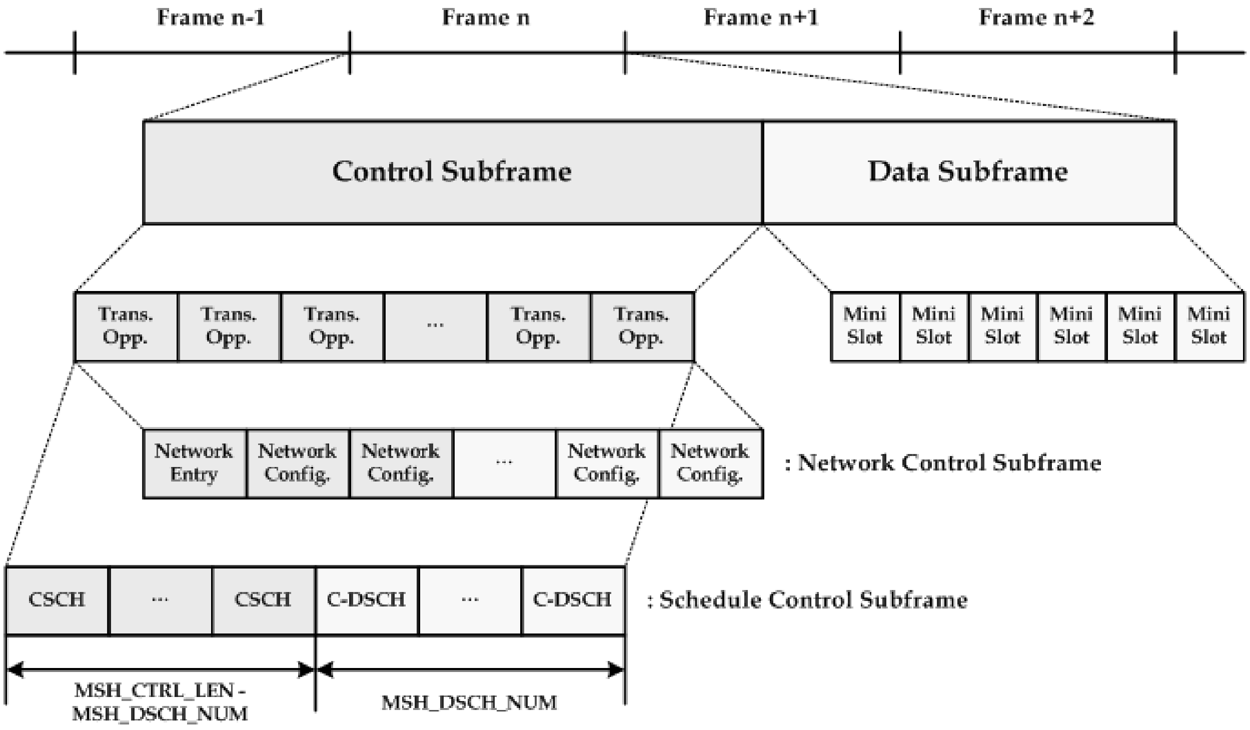


Fig. 1. Frame structure of IEEE 802.16 mesh mode

This chapter deals with holdoff algorithms for the IEEE 802.16 mesh mode with the C-DSCH in multi-hop wireless mesh networks and is structured as follows. In Section 2, we present the overview of the IEEE 802.16 mesh mode with the C-DSCH. In Section 3, we first explain a static holdoff algorithm, which is defined in the IEEE 802.16 mesh standard mesh standard, and introduce its limitations with the respect to network throughput. Next, we describe existing dynamic holdoff algorithm and introduce its advantages and disadvantages. Lastly, we propose an adaptive holdoff algorithm based on node state. In the adaptive holdoff algorithm, nodes collect neighborhood information in a distributed manner and calculate a metric to reflect the current network situation around them. And, nodes decide appropriate holdoff times on the basis of the calculated metric in order to improve network throughput. In Section 4, some simulation results are given. Simulation results show that it is required for nodes to adaptively adjust their holdoff times according to current network situations in order to improve the performance of M-WMN based on the IEEE 802.16 mesh mode with the C-DSCH. In Section 5, we make a conclusion.

2. Overview of IEEE 802.16 mesh mode with C-DSCH

In this section, the overview of the IEEE 802.16 mesh mode with the C-DSCH is given in four aspects: frame structure, MSH-DSCH message format, bandwidth reservation by three-way handshaking, and distributed election algorithm.

2.1 Frame structure of IEEE 802.16 mesh mode

Figure 1 shows the frame structure of the IEEE 802.16 mesh mode. As shown in the figure, the frame of the IEEE 802.16 mesh mode is based on the time division multiple access (TDMA) frame structure and consists of the control and data subframes.

The data subframe is used to transmit data packets and the control subframe is used to transmit MAC management messages. The control and data subframes are composed of multiple transmission opportunities (TOs) and minislots, respectively. As one of network parameters, the number of TOs consisting of the control subframe (MSH_CTRL_LEN) is set to one value between 0 and 15. Each TO in the control subframe consists of seven OFDM symbols and can carry only one MAC management message.

The control subframe consists of two types of subframes: the network control subframe and the schedule control subframe. The network control subframe enables new nodes to join mesh network. In a mesh network, nodes broadcast network entry and network configuration information and this enables a new node to get synchronization and initial network entry into the mesh network. The schedule control subframe is used to transmit MAC management messages in order to reserve minislots in the data subframe and it is divided into two parts, as shown in Fig. 1. The first part is used for the MSH-CSCH and MSH-CCFG messages transmissions in the CSCH mechanism and the second part is used for the MSH-DSCH message transmission in the C-DSCH mechanism.

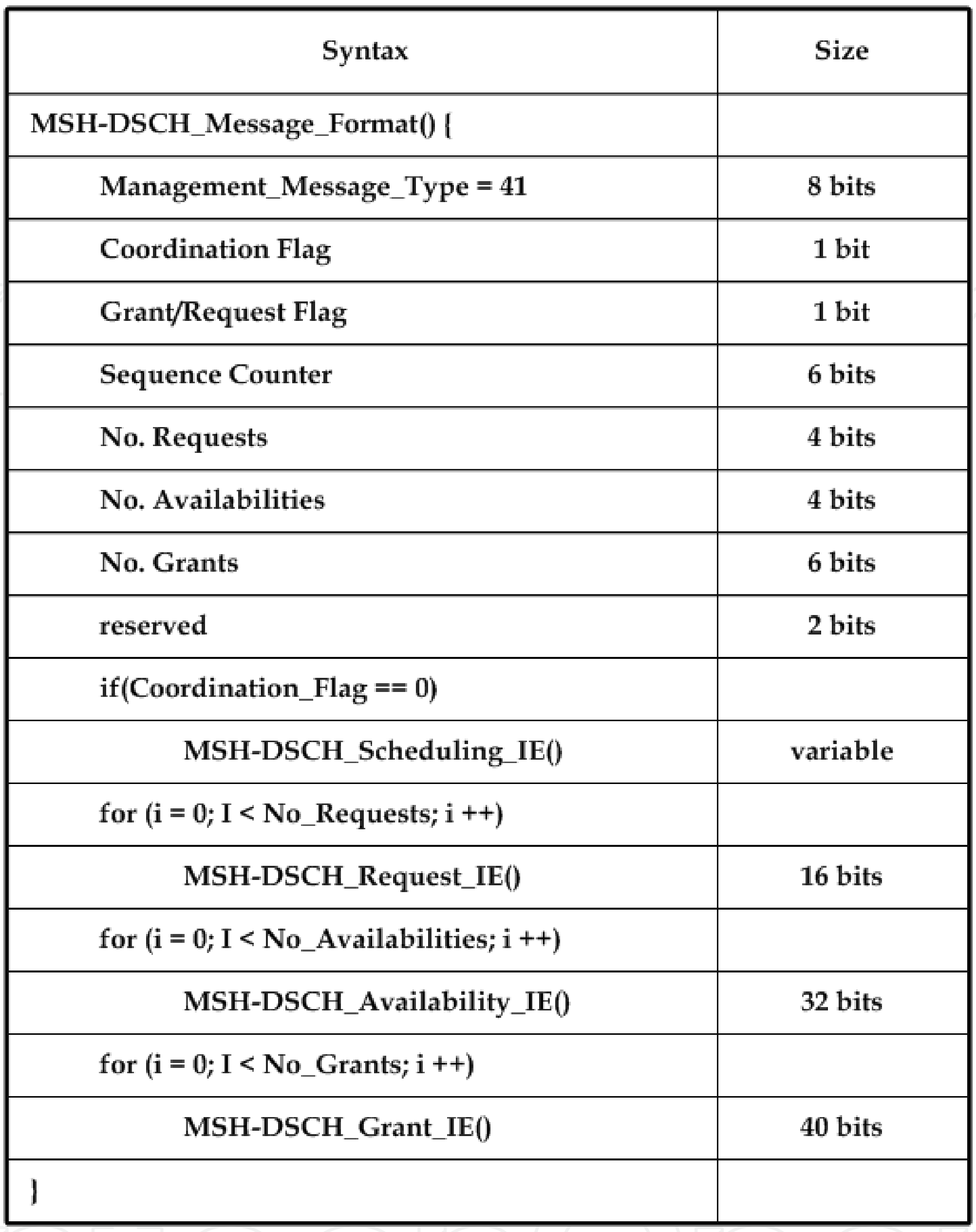
The number of TOs (MSH_DSCH_NUM) consisting of the C-DSCH part is selected in a range between 0 and 15 and thus, the length of the CSCH part is equal to MSH_CTRL_LEN – MSH_DSCH_NUM.

2.2 MSH-DSCH message format

In this chapter, because we focus on holdoff algorithms for the IEEE 802.16 mesh mode with the C-DSCH, only the MSH-DSCH message format related with the C-DSCH is given in detail. Every node sends its available resource information to neighbor nodes via MSH-DSCH messages. The request, grant, and confirmation of resource are also accomplished by exchanging the MSH-DSCH messages between a pair of nodes.

As shown in Fig. 2, in the C-DSCH, the MSH-DSCH message contains the following information elements (IE): [Morge, P. S., 2007]

- *MSH-DSCH_Scheduling_IE* includes the next MSH-DSCH transmission times and holdoff exponents of a node and its neighbor nodes.
- *MSH-DSCH_Request_IE* is used by a node to specify its bandwidth demand for a specific link.
- *MSH-DSCH_Availability_IE* is used by a node to convey its own status for individual minislots to its neighbors.
- *MSH-DSCH_Grant_IE* is used by a node to send bandwidth grant in response to a bandwidth request as well as to send a grant confirmation for a received bandwidth grant.



Syntax	Size
MSH-DSCH_Message_Format() {	
Management_Message_Type = 41	8 bits
Coordination Flag	1 bit
Grant/Request Flag	1 bit
Sequence Counter	6 bits
No. Requests	4 bits
No. Availabilities	4 bits
No. Grants	6 bits
reserved	2 bits
if(Coordination_Flag == 0)	
MSH-DSCH_Scheduling_IE()	variable
for (i = 0; I < No_Requests; i ++)	
MSH-DSCH_Request_IE()	16 bits
for (i = 0; I < No_Availabilities; i ++)	
MSH-DSCH_Availability_IE()	32 bits
for (i = 0; I < No_Grants; i ++)	
MSH-DSCH_Grant_IE()	40 bits
}	

Fig. 2. MSH-DSHC message format

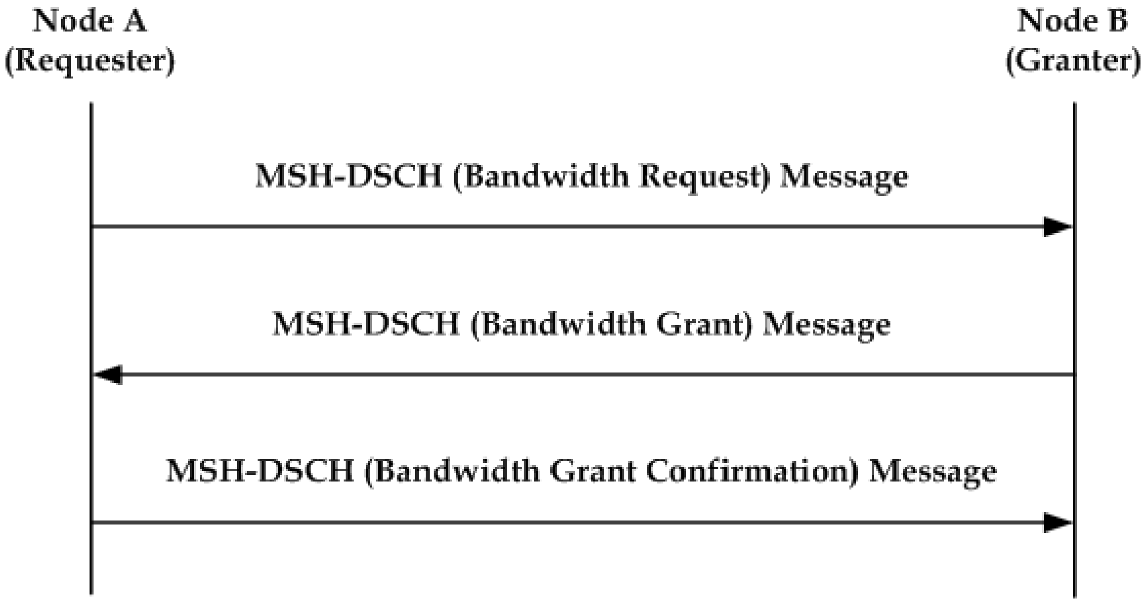

Node A
(Requester)
Node B
(Granter)
MSH-DSCH (Bandwidth Request) Message
MSH-DSCH (Bandwidth Grant) Message
MSH-DSCH (Bandwidth Grant Confirmation) Message

Fig. 3. Three-way handshaking

2.3 Bandwidth reservation by three-way handshaking

Based on the four IEs above, in the C-DSCH, nodes perform the bandwidth reservation process by three-way handshaking: bandwidth request, bandwidth grant, and bandwidth grant confirmation, as shown in Fig. 3.

In the bandwidth reservation procedure, node A (requester) uses the Link ID to uniquely indentify a link for which node A needs bandwidth, and it sends an MSH-DSCH message that contains a set of MSH-DSCH_Request_IEs and a set of MSH-DSCH_Availability_IEs. The MSH-DSCH message with an MSH-DSCH_Request_IE is received by all the neighbors around node A. After receiving the bandwidth request message, node B (granter) looks up the set of available minislots in order to select a subset of minislots that is available for data receptions from node A. Node B chooses an available range of minislots for a bandwidth grant and sends the MSH-DSCH message with the MSH-DSCH_Grant_IE, which contains the set of minislots for the bandwidth grant. All the neighbors around node B receive the bandwidth grant message and they update their availability status by reflecting the scheduled data reception specified in the bandwidth grant message.

In the bandwidth grant confirmation, node A transmits an MSH-DSCH message with the MSH-DSCH_Grant_IE in order to inform all the neighbors around node A of the scheduled data transmission information. The neighbors update their availabilities by reflecting the newly scheduled data transmission. Data transmission is accomplished only over the reserved minislots, after the successful transmission of the bandwidth grant confirmation message.

2.4 Distributed election algorithm

In the three-way handshaking process, nodes independently select their own transmission times of MSH-DSCH messages using the distributed election algorithm. The DEA enables nodes to forward MSH-DSCH messages in an M-WMN in a completely distributed manner. A node can transmit an MSH-DSCH message without message collision at a selected transmission time by the DEA. In the C-DSCH, the transmission time corresponds to a specific TO in the schedule control subframe shown in Fig. 1. In the DEA, a node performs two functions: collecting the next transmission times (Next_Xmt_Times) of all nodes within its extended neighborhood (one-hop and two-hop neighbors) and selecting its Next_Xmt_Time.

1) Collecting the Next_Xmt_Times of all nodes within extended neighborhood: to select a collision-free TO for the MSH-DSCH transmission, every node should collect the Next_Xmt_Times of all nodes within its extended neighborhood. So, every node must inform its neighbors of the next MSH-DSCH transmission time of its neighbors as well as itself. In the DEA, nodes broadcast not an exact Next_Xmt_Time but the next transmission time interval (Next_Xmt_Time_Interval) information, which is expressed by two parameters of next transmission maximum (Next_Xmt_Mx) and transmission holdofff exponent (Xmt_Holdoff_Exp) as follows:

$$2^{Xmt\_Holdoff\_Exp} * Next\_Xmt\_Mx < Next\_Xmt\_Time < 2^{Xmt\_Holdoff\_Exp} * (Next\_Xmt\_Mx + 1) \qquad (1)$$

The Next_Xmt_Time_Interval is a series of one or more C-DSCH TOs. Every node broadcasts MSH-DSCH message containing the next transmission time interval information (Next_Xmt_Mx and Xmt_Holdoff_Exp) of its one-hop neighbors as well as itself. Using the next transmission interval information received from neighbors via MSH-DSCH messages, every node can calculate the Next_Xmt_Time_Intervals of all nodes within its extended neighborhood. In the IEEE 802.16 mesh standard, a node first selects its own Next_Xmt_Time at a current MSH-DSCH transmission time (Current_Xmt_Time) and then, it calculates a Next_Xmt_Mx value using equation (1) and a current Xmt_Holdoff_Exp value. Next, the node broadcasts an MSH-DSCH message containing the Next_Xmt_Mx and Xmt_Holdoff_Exp, which represents its Next_Xmt_Time_Interval, in order to inform neighbors of its next transmission time information.

2) Selecting a Next_Xmt_Time: in the IEEE 802.16 mesh standard, after an MSH-DSCH message transmission, a node is not eligible to transmit an MSH-DSCH message for transmission holdoff time (Xmt_Holdoff_Time) in order to share radio resource with other nodes in an M-WMN. Therefore, when a node selects a Next_Xmt_Time in the DEA, it first sets the temporary transmission time (Temp_Xmt_Time) as follows:

$$Temp\_Xmt\_Time = Current\_Xmt\_Time + Xmt\_Holdoff\_Time + 1 \qquad (2)$$

Next, the node should determine the set of eligible competing nodes related to the Temp_Xmt_Time from its neighbor table. This set will include neighbors that meet at least one of the following conditions: [Bayer, N., 2006]

- The Next_Xmt_Time_Interval of neighbor includes the Temp_Xmt_Time,
- The earliest subsequent transmission time (Earliest_Subsequent_Xmt_Time) of neighbor is equal to or smaller than Temp_Xmt_Time, or
- The Next_Xmt_Time of neighbor is not known.

In the conditions above, the Earliest_Subsequent_Xmt_Time is the earliest time that the node is eligible to transmit a MSH-DSCH message after the Next_Xmt_Time, as shown in equation (3):

$$Earliest\_Subsequent\_Xmt\_Time = Next\_Xmt\_Time + Xmt\_Holdoff\_Exp + 2^{Xmt\_Holdoff\_Exp} * Next\_Xmt\_Mx \tag{3}$$

The DEA is performed based on this set of eligible competing nodes. With this set, a node checks whether any competing nodes do not use a specific TO corresponding to the Temp_Xmt_Time, or not. If any competing nodes of the node do not use the specific TO, the node is the winner of the distributed election and the Next_Xmt_Time is set equal to the Temp_Xmt_Time. If a node does not win the distributed election, the Temp_Xmt_Time is set to Temp_Xmt_Time + 1 and the above process is performed again in order to select a collision-free Next_Xmt_Time.

3. Holdoff algorithms for IEEE 802.16 mesh mode

In the IEEE 802.16 mesh standard with the C-DSCH, a node is not eligible to transmit any MSH-DSCH messages for at least holdoff time after an MSH-DSCH message transmission. The reason to stop an MSH-DSCH message transmission for a holdoff time is for nodes to share radio resource with other nodes in M-WMN. However, a holdoff mechanism can result in resource waste by holding MSH-DSCH message transmission even in a lightly loaded network situation. Some holdoff algorithms have been proposed for the IEEE 802.16 mesh mode with the C-DSCH in order to improve the network throughput and to share the radio resource.

3.1 Static holdoff algorithm

In the IEEE 802.16 mesh standsard, the Xmt_Holdoff_Time is calculated based on the static Xmt_Holdoff_Exp and Xmt_Holdoff_Exp_Base values, as shown in equation (4):

$$Xmt\_Holdoff\_Time = 2^{(Xmt\_Holdoff\_Exp + Xmt\_Holdoff\_Exp\_Base)} \tag{4}$$

The IEEE 802.16 mesh standard defines static holdoff algorithm. In the static holdoff algorithm, the Xmt_Holdoff_Exp_Base is fixed to 4 for resource sharing between nodes in an M-WMN. The Xmt_Holdoff_Exp is also set to a specific value between 0 and 7 in initial node configuration procedure. That is, all nodes in M-WMN have an identical transmission holdoff time regardless of current network situation. Thus, when Xmt_Holdoff_Exp = 0, the competition between nodes happen severe and a node experiences long contention times before reserving resource. On the other hand, as the Xmt_Holdoff_Exp value increases, the contention between nodes becomes less competitive; however, nodes get a longer holdoff time and thus transmission interval becomes longer [Cao, M., 2005]. In particular, if the Xmt_Holdoff_Exp is set to an unnecessarily large value in a lightly loaded network situation, the waste of resource happens. Hence, the static holdoff algorithm can result in network throughput degradation regardless of an assigned Xmt_Holdoff_Exp value.

3.2 Dynamic holdoff algorithm

In [Cao, M., 2005], the authors performed the modelling and performance analysis of distributed scheduler in the IEEE 802.16 mesh mode, and they concluded that the capacity of IEEE 802.16 mesh network can be optimized by assigning appropriate Xmt_Holdoff_Exp values to nodes in the network.

In [Bayer, N., 2007], the dynamic exponent (DynExp) algorithm was proposed. In the DynExp algorithm, nodes that are currently not sending, receiving, or forwarding data packets use large Xmt_Holdoff_Exp values and thus get large Xmt_Holdoff_Time values. Nodes that transmit, receive, and forward data packets or nodes that have been selected by the routing protocol as potential forwarding nodes use small Xmt_Holdoff_Exp values and thus get small Xmt_Holdoff_Time values. For this purpose, nodes are classified as follows:

- Mesh base station (M-BS) is a normal mesh base station.
- Active node (ACT) is a node that is part of an active route and sends, receives, or forwards data packets.
- Sponsor node (SN) is a node that is not part of an active route but has been selected as a potential forwarding node by at least one of its neighbors.
- Inactive node (IN-ACT) is a node that is not part of an active route and does not send, receive or forward data packets.

According to the node types above, different Xmt_Holdoff_Exp values are defined as follows:

$$\begin{aligned} 0 < Xmt\_Holdoff\_Exp_{M\text{-}BS} &< Xmt\_Holdoff\_Exp_{ACT} \\ &< Xmt\_Holdoff\_Exp_{SN} \\ &< Xmt\_Holdoff\_Exp_{IN\text{-}ACT}' \end{aligned}$$

By using different Xmt_Holdoff_Exp values according to current node types, the DyExp algorithm improves network capacity. However, because the operation of DynExp algorithm depends on information from the routing layer, the DynExp algorithm cannot be operated independently without cooperation with routing layer.

3.3 Adaptive holdoff algorithm

In this section, we propose an adaptive holdoff algorithm. In the adaptive holdoff algorithm, nodes adaptively adjust the Xmt_Holdoff_Exp value according to current node state.

3.3.1 Definitions

Before proposing the adaptive holdoff algorithm, the neighbors of a node are classified into four types: tx-neighbor, rx-neighbor, tx/rx-neighbor, and null-neighbor. Each classification is defined as follows:

- A tx-neighbor of a node is a neighbor that has a data packet to send to the node.
- An rx-neighbor of a node is a neighbor to which the node has a data packet to send.
- A tx/rx-neighbor of a node is a neighbor that is both tx-neighbor and rx-neighbor of the node.

- A null-neighbor of a node is a neighbor that has no data packet to send to or receive from the node.

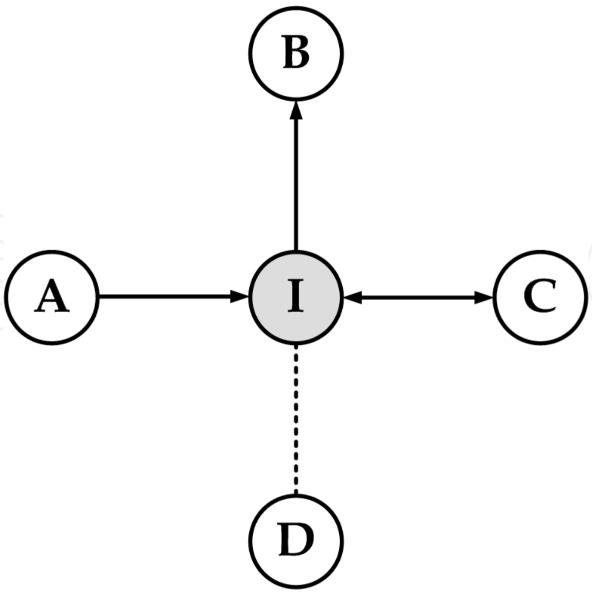


Fig. 4. Classification of neighbor nodes: A solid line between a pair of nodes indicates that there is at least one data packet to transmit between the two nodes, with an arrow showing the direction of the pending data transmission. A dotted line indicates that there is no data waiting for transmission between the pair of nodes

In Fig. 4, node I has four types of neighbors. Nodes A, B, C, and D correspond to the tx-neighbor, rx-neighbor, tx/rx-neighbor, and null-neighbor of node I, respectively.

3.3.2 Adaptive holdoff algorithms based on node state

In this section, an adaptive holdoff algorithm based on node state is presented. The adaptive holdoff algorithm does not depend on information from higher layer such as routing layer and it operates independently. In addition, it maintains backward compatibility with the IEEE 802.16 mesh standard.

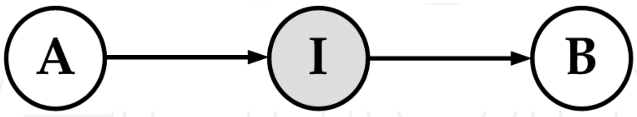
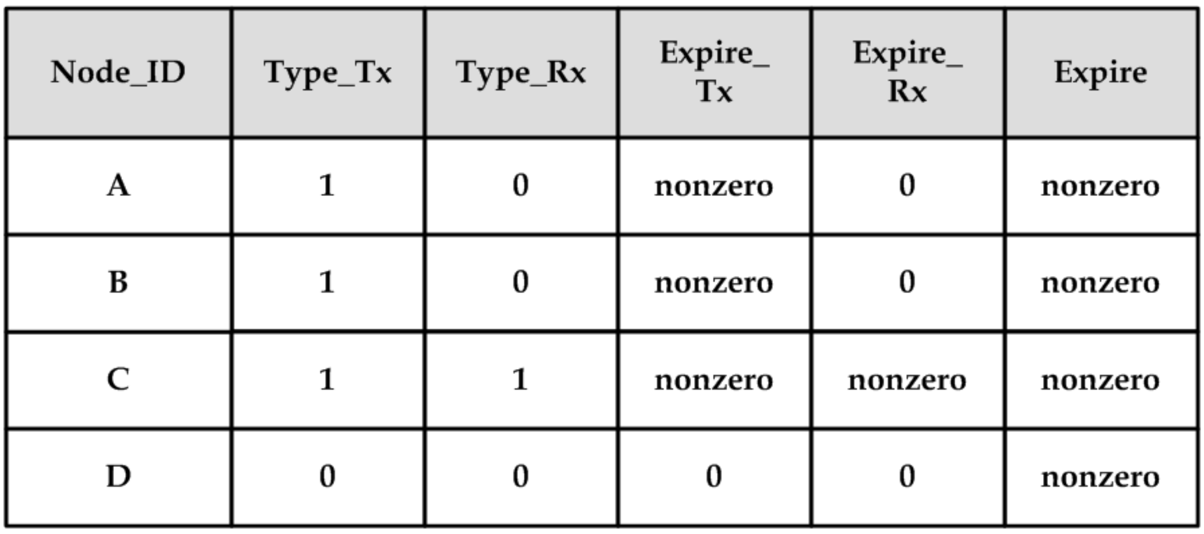


Fig. 5. Example topology: A solid line between a pair of nodes indicates that there is at least one data packet to transmit between the two nodes, with an arrow showing the direction of the pending data transmission

In Fig. 5, node I has two links (Links 1 and 2) for communication with nodes A and B, respectively. As shown in Fig. 5, node I should reserve each minislot for data reception from node A and data transmission to node B. On the other hand, node A needs to reserve only minislots for data transmission to node I, and node B needs to reserve only minislots for data reception from node I. Therefore, node I should be able to access schedule control subframe at a higher rate than nodes A and B in order to prevent node I from being a bottleneck.



Node_ID	Type_Tx	Type_Rx	Expire_ Tx	Expire_ Rx	Expire
A	1	0	nonzero	0	nonzero
B	1	0	nonzero	0	nonzero
C	1	1	nonzero	nonzero	nonzero
D	0	0	0	0	nonzero

Fig. 6. Neighbor table of node I in Fig. 4

Based on the features above, the *Weight* of a node is first defined for the adaptive holdoff algorithm as follows:

$$Weight = tx\_exist * 1 + rx\_exist * 2 \tag{5}$$

In equation (5), the tx_exist represents whether a node has at least one tx-neighbor or not. And, the rx_exist indicates whether a node has at least one rx-neighbor or not. Because a node transmits only one MSH-DSCH message (bandwidth grant message) to tx-neighbor over the TO of the schedule control subframe during the three-way handshaking, the tx_exist is multiplied by 1 in equation (5). On the other hand, because a node transmits two MSH-DSCH messages (bandwidth request and bandwidth grant confirmation messages) to rx-neighbor during the three-way handshaking, the rx_exist is multiplied by 2. In addition, as shown in Fig. 2, the MSH-DSCH message can contains multiple MSH-DSCH_Request_IE()s, MSH-DSCH-Availability_IE()s, and MSH-DSCH_Grants_IE()s at one time. Namely, a node can simultaneously request, grant, and confirm resources for multiple links via only one MSH-DSCH message. Therefore, the *Weight* depends on only the tx_exist and rx_exist in equation (5).

Next, a neighbor table is presented in order to explain how to obtain the *Weight* of a node. In the adaptive holdoff algorithm, nodes store neighbor information in their neighbor table, as shown in Fig. 6. Each entry in the neighbor table contains Node_ID, Type_Tx/Rx, Expire_Tx/Rx, and Expire. Node_ID denotes the identifier of a neighbor. Type_Tx/Rx is used to distinguish the neighbor types: tx-neighbor, rx-neighbor, tx/rx-neighbor, or null-neighbor. Expire_Tx/Rx indicates the expiration time of a neighbor as a tx-neighbor or as an rx-neighbor. Expire represents the expiration time of a neighbor table entry.

In the adaptive holdoff algorithm, the neighbor table is managed in the following way:

1) Setting Type_Tx/Rx, Expire_Tx/Rx, and Expire: Whenever a node sends or receives a data packet to or from a neighbor over minislots of data subframe, it sets the values of Type_Tx/Rx, Expire_Tx/Rx, and Expire in the neighbor table entry for the neighbor. When a node receives a data packet from a neighbor, it finds the neighbor table entry for that neighbor in its neighbor table and sets the values of Type_Tx, Expire_Tx, and Expire as follows: the Type_Tx is set to 1. The Expire_Tx and Expire are set to EXPIRE_TIME, which is a pre-defined constant value. When a node sends a data packet to a neighbor, it finds the neighbor table entry for that neighbor in its neighbor table and sets Type_Rx, Expire_Rx, and Expire as follows: the Type_Rx is set to 1. The Expire_Rx and Expire are set to EXPIRE_TIME.

2) Maintaining/removing neighbor table entries: Maintenance and removal of a neighbor table entry is performed using the neighbor timer. The neighbor timer expires periodically; when this occurs, a node compares the current time with each values of Expire_Tx/Rx and Expire in a neighbor table entry and updates Type_Tx/Rx, Expire_Tx/Rx, and Expire in the following way: if the values of Expire_Tx/Rx are higher than the current time, the values of Type_Tx/Rx and Expire_Tx/Rx are not changed; otherwise, they are set to 0. If the value of Expire is less than the current time, the neighbor table entry is removed; otherwise, the value of Expire is not changed. This procedure is repeated for all the entries in the neighbor table.

According to the neighbor table management above, node I in Fig. 4 has the neighbor table as shown in Fig. 6. In the adaptive holdoff algorithm, a node can calculate its *Weight* using its neighbor table. For example, node I can calculate the *Weight* by checking the values of Type_Tx/Rx in its neighbor table entries as follows: if a node has at least one neighbor entry with the Type_Tx value of 1 in its neighbor table, the tx_exist is set to 1; otherwise, the tx_exist is set to 0. In a similar way, if a node has at least one neighbor entry with the Type_Rx value of 1 in its neighbor table, the rx_exist is set to 1; otherwise, the rx_exist is set to 0. Therefore, node I in Fig. 4 has the *Weight* value of 3 (= 1 * 1 + 1 * 2).

In the adaptive holdoff algorithm, the *Weight* of a node represents its access rate required to reserve a shared resource for data transmission or reception. Therefore, a node with a large *Weight* value should be able to access the TOs of the schedule control subframe at a high rate, and a node with a small *Weight* value should access the TOs of the schedule control subframe at a low rate.

To achieve this goal, in the adaptive holdoff algorithm, current transmission holdoff exponent (Cur_Xmt_Holdoff_Exp) is defined as follows:

$$Cur\_Xmt\_Holdoff\_Exp = Max\_Xmt\_Holdoff\_Exp * (1 - Weight / 3) \tag{6}$$

In equation (6), the Max_Xmt_Holdoff_Exp is the maximum value of transmission exponent. The Max_Xmt_Holdoff_Exp of a node is set to a constant in initial node configuration process.

And then, a node selects its Xmt_Holdoff_Time as follows:

$$Xmt\_Holdoff\_Time = 2^{(Xmt\_Holdoff\_Base + Cur\_Xmt\_Holdoff\_Exp)} \tag{7}$$

Algorithm 1 shows the operation of the adaptive holdoff algorithm. In Algorithm 1, it is assumed that the Xmt_Holdoff_Exp_Base is set to 4 and the Max_Xmt_Holdoff_Exp to 3.

As shown in Algorithm 1, if a node has a large *Weight* value, the small Xmt_Holdoff_Time value is obtained by setting the Cur_Xmt_Holdoff_Exp to a small value. Thus, the node can access the TOs of the schedule control subframe for resource reservation at a high rate. On the other hand, if a node has a small *Weight* value, the large Xmt_Holdoff_Time value is obtained by setting the Cur_Xmt_Holdoff_Exp to a large value. Thus, the node can access the TOs of the schedule control subframe for resource reservation at a low rate. In this way, nodes can adjust their access rate to the TOs of the schedule control subframe according to their current state.

Algorithm 1 *Adaptive Holdoff Algorithm based on Node State*

Notations Used:
tx_exist = indicate whether a node currently has at least one tx-neighbor
rx_exist = indicate whether a node currently has at least one rx-neighbor
Weight = access rate of a node
Current_Xmt_Holdoff_Exp = current transmission holdoff exponent
Xmt_Holdoff_Exp_Base = transmission holdoff exponent base (= 4)
Max_Xmt_Holdoff_Exp = maximum value of transmission holdoff exponent (= 3)
Xmt_Holdoff_Time = transmission holdoff time

```
Set the tx_exist by cheking the Type_Txs of neighbor table entries
Set the rx_exist by cheking the Type_Rxs of neighbor table entries
Compute Weight = tx_exist * 1 + rx_exist * 2
if (Weight == 0)
    Current_Xmt_Holdoff_Exp = Max_Xmt_Holdoff_Exp;
end if
else
    Current_Xmt_Holdoff_Exp = Max_Xmt_Holdoff_Exp *  (1 - Weight / 3);
end else
Compute Xmt_Holdoff_Time = $2^{(\text{Xmt\_Holdoff\_Exp\_Base} + \text{Current\_Xmt\_Holdoff\_Exp})}$
```

4. Simulation Evaluation

Computer simulations were performed to evaluate the performance of the holdoff algorithms. To show the importance of adjusting holdoff time according to current network situation from the perspective of performance improvement, the performance of the adaptive holdoff algorithm based on node state (AHA) is compared with that of the static holdoff algorithm (SHA), which is described in the IEEE 802.16 mesh standard.

4.1 Simulation environments

NCTUns-4.0 [Wang, S. Y., 2007-(a)], [Wang, S. Y., 2007-(b)], [Wang, S. Y., 2007-(c)] is used to evaluate the performance of the holdoff algorithms for the IEEE 802.16 mesh mode with the C-DSCH. Table 1 shows the parameters used in the simulation. For all other parameters, the default values provided in NCTUns-4.0 are used. Figure 7 shows the network topology used to evaluate the performance of the holdoff algorithms. The network consists of 16 nodes. Among 16 nodes, one node acts as MeshBS and the others as MeshSSs. Node 11 corresponds to the MeshBS and other nodes to MeshSSs. All nodes remain stationary for a simulation time of 300s. The number of traffic flows is varied from 1 to 6 to investigate the performance variation in different offered loads. Each traffic source generates user datagram protocol (UDP)-based data packets with the size of 512 bytes at a rate of 2Mbits/s.

Parameters	Value
Max. transmission range	MeshSS: 400m, MeshBS: 400m
Bytes per OFDM symbol	108 (64-QAM_3/4)
Xmt_Holdoff_Exp_Base	4
Xmt_Holdoff_Exp	3 (only static holdoff algorithm)
Distance between MeshSSs	300m

Table 1. Simulation parameters for holdoff algorithms

Simulations are performed in two scenarios: multi-hop peer-to-peer and multi-hop Internet access scenarios. Figure 8 shows traffic flows in the multi-hop peer-to-peer scenario. The multi-hop Internet access scenarios are classified into upload and download patterns. The traffic flows for upload and download patterns are shown in Figs. 9 and 10, respectively.

To evaluate the performances of two holdoff algorithms (SHA and AHA), the following performance metrics are used:

- The average per-flow throughput is the average packet throughput at a receiver.
- The total network throughput is the sum of the average packet throughputs at all receivers.

4.2 Simulation results

In the multi-hop peer-to-peer scenario where traffic is distributed throughout entire network, the performance of AHA is compared with that of SHA. The average per-flow throughput and total network throughput are shown in Figs. 11 and 12, respectively. In the SHA, because all nodes have an identical Xmt_Holdoff_Exp value (= 3), they have an identical

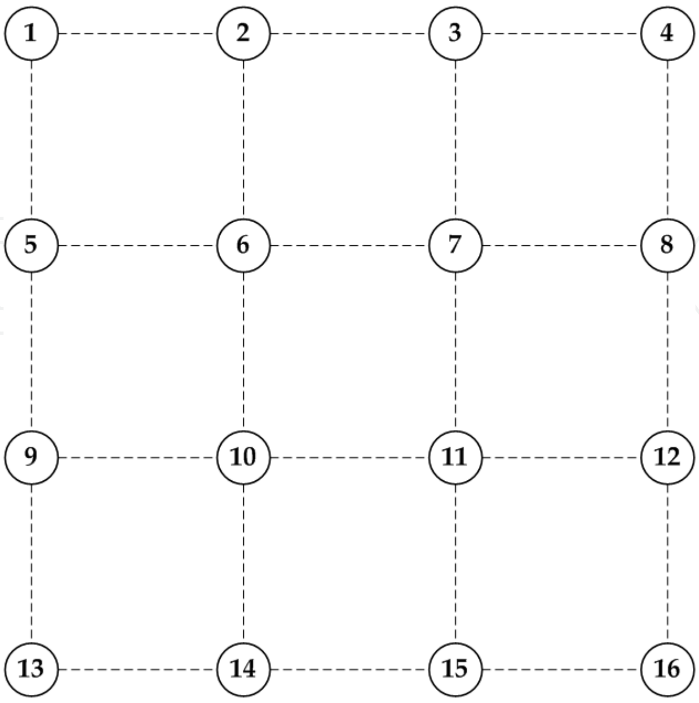


Fig. 7. Simulation network topology

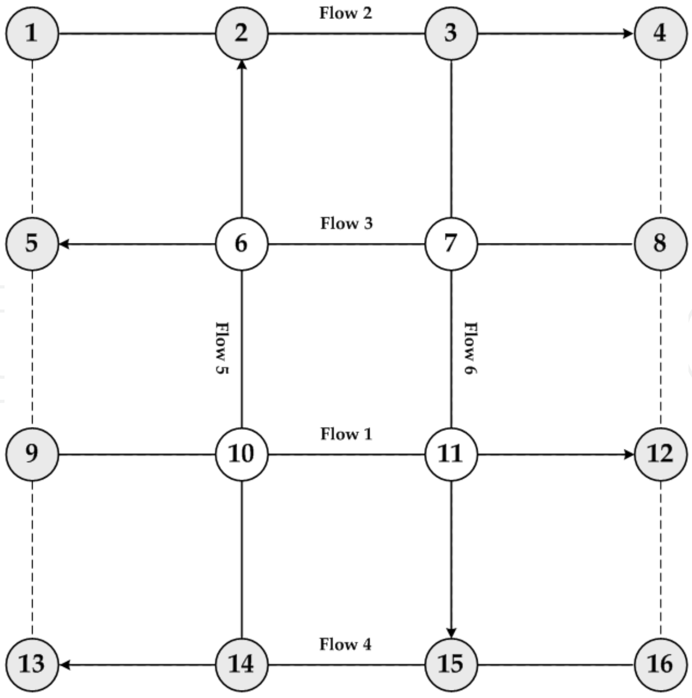


Fig. 8. Multi-hop peer-to-peer

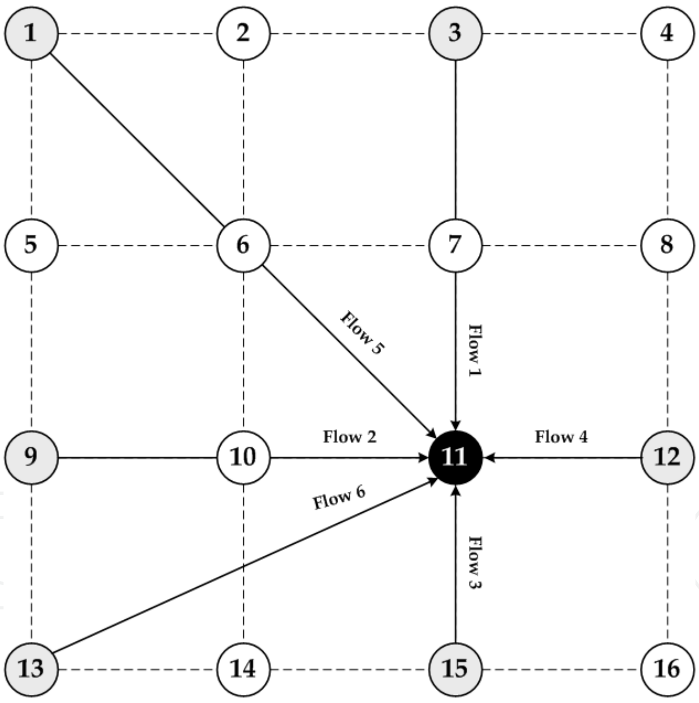


Fig. 9. Multi-hop Internet access with upload pattern

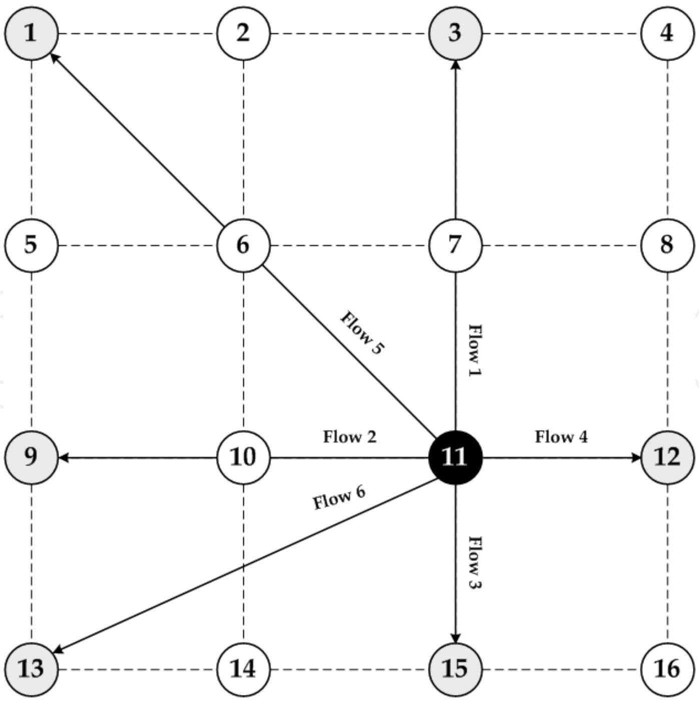


Fig. 10. Multi-hop Internet access with download pattern

holdoff time regardless of their current state. However, in the AHA, because nodes adjust their Xmt_Holdoff_Exp in a range between 0 and 3 according to their current state, the radio resource can be used efficiently.

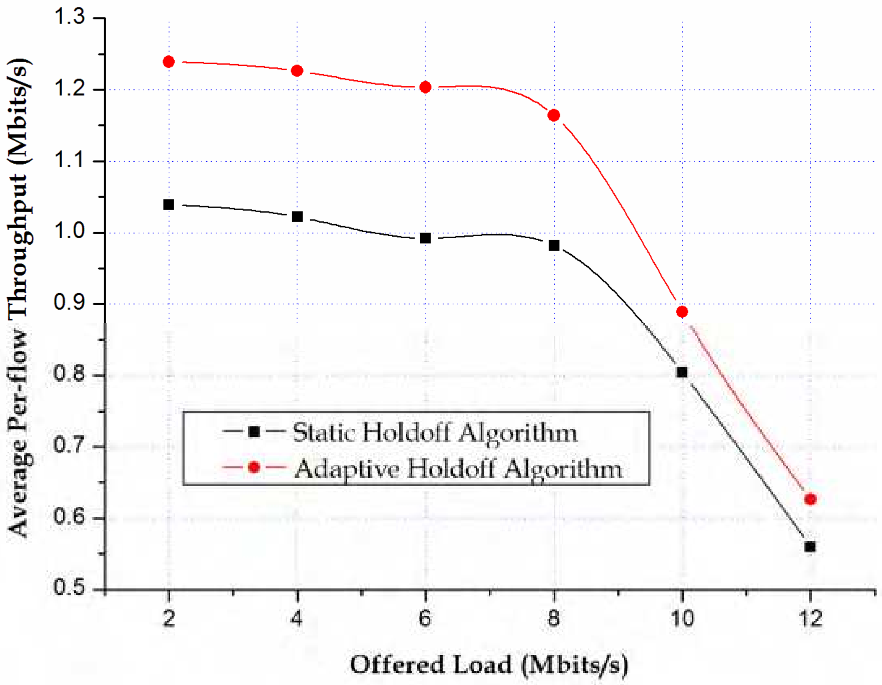


Fig. 11. Average per-flow throughput in multih-hop peer-to-peer

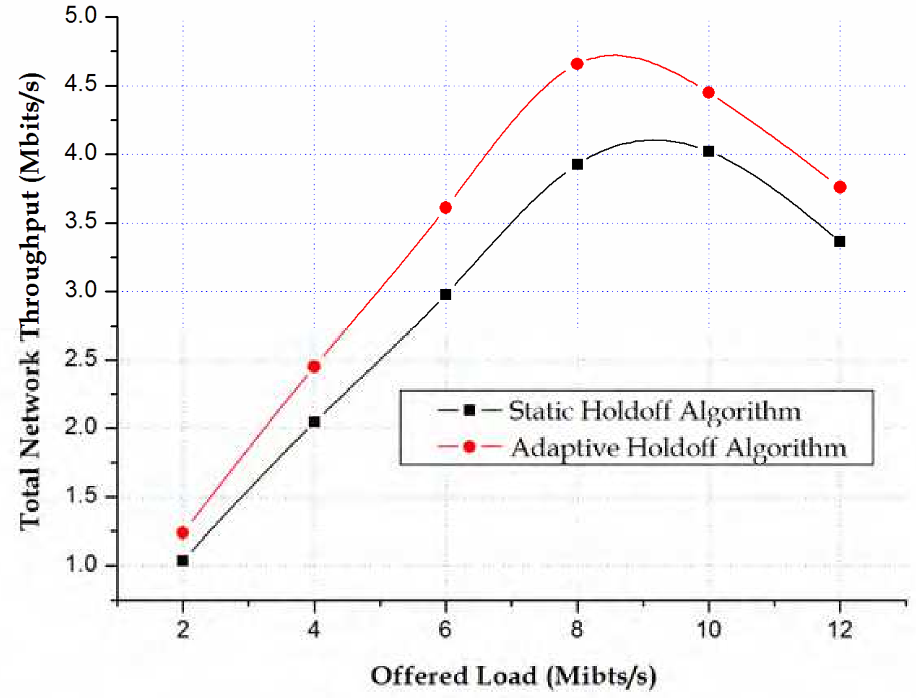


Fig. 12. Total network throughput in multi-hop peer-to-peer

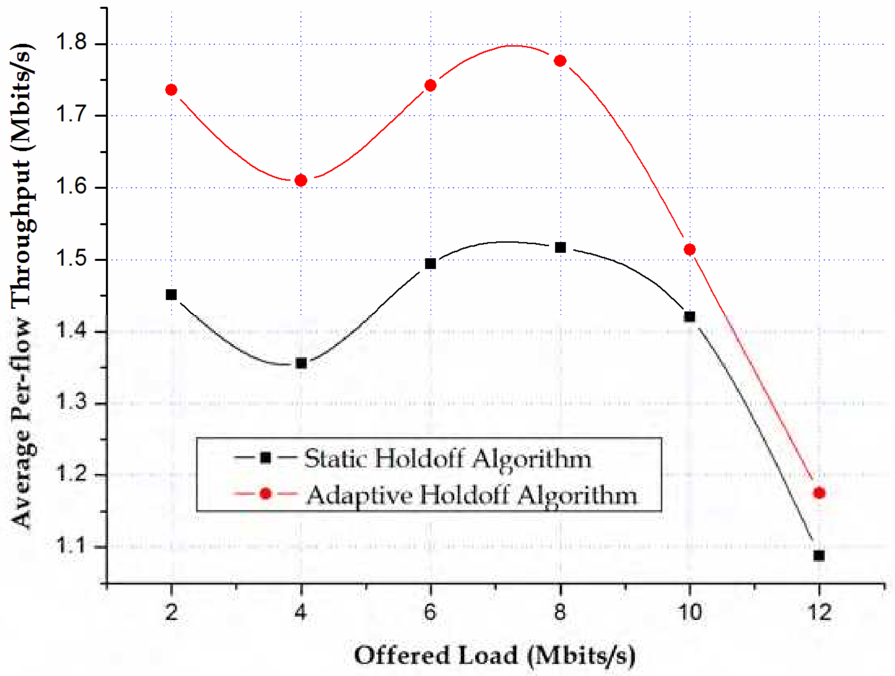


Fig. 13. Average per-flow throughput in multi-hop Internet access - upload

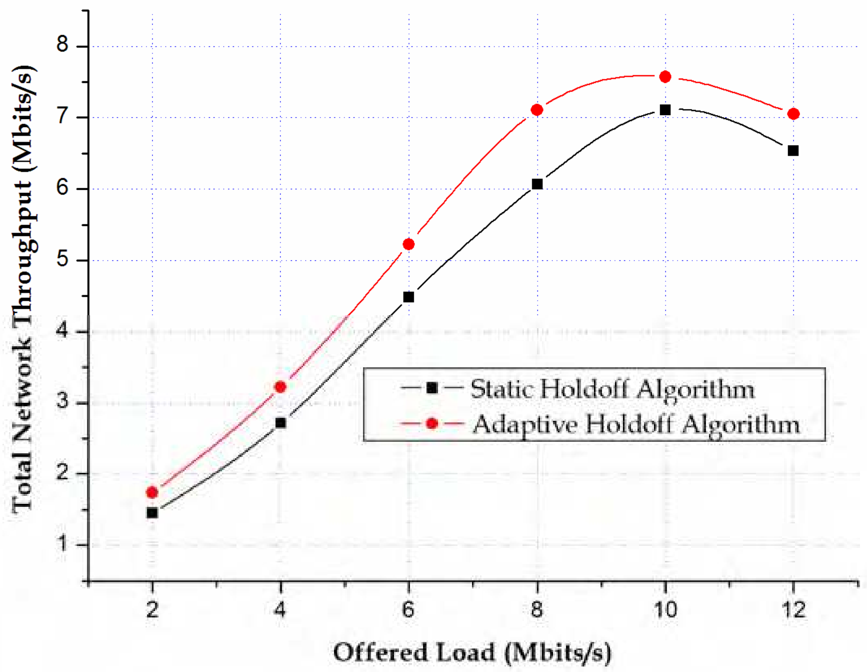


Fig. 14. Total network throughput in multi-hop Internet access - upload

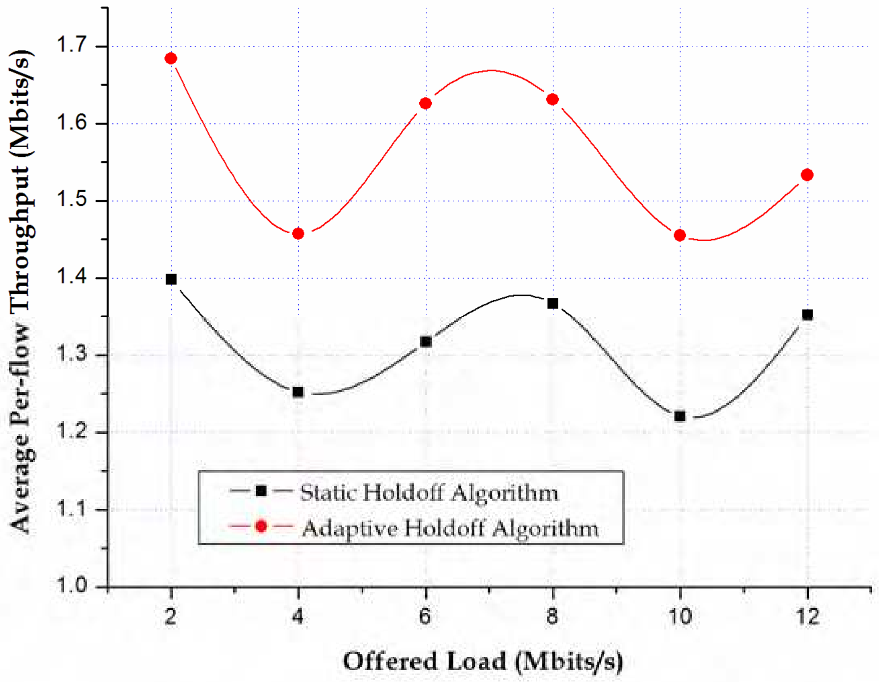


Fig. 15. Average per-flow throughput in multi-hop Internet access - download

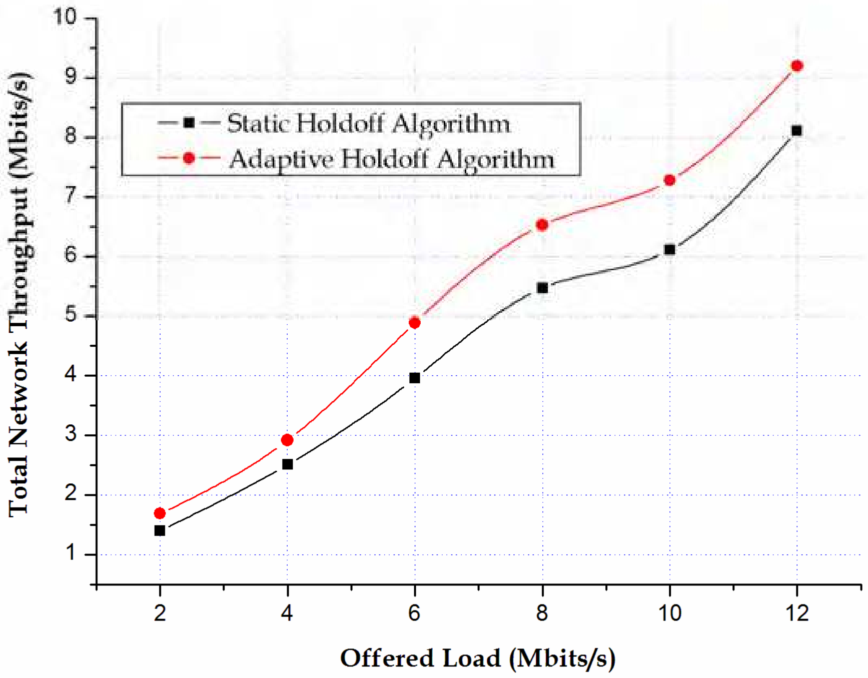


Fig. 16. Total Total network throughput in multi-hop Internet access - download

For example, because a node with the *Weight* = 0 currenlty has no data communication with neighbors, even though the node uses the Xmt_Holdoff_Exp of 3 in the holdoff process, the communication problem does not happen. Furthermore, that a node with a small *Weight* value reduces the access rate to the schedule control subframe enables other nodes with a large *Weight* value to access the schedule control subframe at a high rate.

A node with the *Weight* = 3 can access the schedule control subframe at a high rate by setting the Xmt_Holdoff_Exp to 0. By adaptively adjusting the Xmt_Holdoff_Exp value according to the current node state, the AHA performs better than the SHA in terms of average per-flow throughput and total network throughput, as shown in Figs. 11 and 12.

In general, access to the Internet through MeshBS is desirable for MeshSSs to obtain necessary service. In the multi-hop Internet access scenario, it frequently happens that MeshSSs upload files to Internet through MeshBS and MeshSSs download files provided in the Internet through MeshBS. As illustrated in Figs. 9 and 10, computer simulations are also performed in the multi-hop Internet access scenario with upload and download patterns. Figures 13 and 14 show the average per-flow throughput and total network throughput in the multi-hop Internet access scenario with upload pattern, respectively. In addition, Figs. 15 and 16 show the average per-flow throughput and total network throughput in the multi-hop Internet access scenario with download pattern, respectively. In the multi-hop Internet access scenario, it frequently happens that nodes act as one of sender or receiver. For example, node 12 serves only as sender in Fig. 9, and only as receiver in Fig. 10. In the AHA, because nodes set their Xmt_Holdoff_Exp to an appropriate value according to their current role, the AHA outperforms the SHA, as shown in Figs. 13, 14, 15, and 16.

5. Conclusion

Multi-hop wireless mesh networks are one of the key features of beyond 3G system because of their flexibility and low-cost deployment. The IEEE 802.16 mesh mode with the C-DSCH has recently emerged as an alternative MAC protocol for establishing M-WMNs. For the IEEE 802.16 mesh mode to serve as a MAC protocol for M-WMN, it should get a high network throughput. However, the holdoff algorithm of the IEEE 802.16 mesh standard has a limitation to the performance improvement of M-WMN. In this chapter, we dealt with existing holdoff algorithms such as static holdoff algorithm and dynamic holdoff algorithm and introduced their limitations. In addition, we proposed an adaptive holdoff algorithm based on node state for the IEEE 802.16 mesh mode with the C-DSCH. The adaptive holdoff algorithm assigns an appropriate Xmt_Holdoff_Exp value according to current network situation and it maintains the backward compatibility with the IEEE 802.16 mesh standard. Simulation results show that it is required for nodes to adaptively adjust their holdoff time according to current network situations in order to improve the performance of an IEEE 802.16 mesh system.

6. Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) and MMPC (Mobile Media Platform Center) support programs supervised by the IITA (Institute of Information Technology Advancement)

7. References

Han, B., Jia, W., and Lin, L. (2007). Performace Evaluation of Scheduling in IEEE 802.16 based Wireless Mesh Networks, Elsevier Computer Communications, Vol. 30, No. 4, Feb. 2007, pp. 782-792, ISSN 0140-3664

Akyildiz, I. F., Wang, X., and Wang W. (2005). Wireless Mesh Networks : A Suervey, Elsevier Computer Networks, Vol. 47, No. 4, Mar. 2005, pp. 445-487, ISSN 1389-1286

IEEE Std. 802.16-2004 (2004). IEEE Standard for Local and Metropolitan Area Networks: Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Oct. 2004

Cao, M.; Ma, W.; Zhang, Q.; Wang, X. & Zhu, W. (2005). Modeling and Performance Analysis of the Distributed Scheduler in IEEE 802.16 Mesh Mode, *Proceedings of ACM MobiHoc*, pp. 78-89, ISBN 1-59593-004-3, Urbana-Champaign, May 2005, ACM, Illinois, USA

Bayer, N.; Xu, B. ; Rakocevic, V. & Habermann, J. (2007). Improving the Performance of the Distributed Scheduler in IEEE 802.16 Mesh Networks, *Proceedings of IEEE VTC Spring*, pp. 1193-1197, ISBN 1-4244-0266-2, Burlington Hotel, Apr. 2007, IEEE, Dublin, Ireland

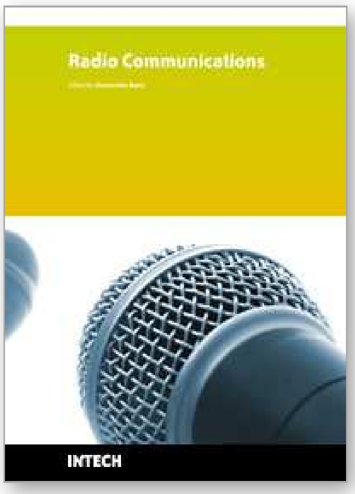
Bayer, N. ; Sivchenko, D.; Xu, B.; Rakocevic, V. & Habermann, J. (2006). Transmission Timing of Signaling Messages in IEEE 802.16 based Mesh Networks, *Proceedings of European Wireless*, ISBN 978-3-8007-2961-6, National Technical University of Athens, Apr. 2006, Athens, Greece

Morge, P. S.; Hollick, M. & Steinmetz, R. (2007). The IEEE 802.16-2004 MeSH Mode Explained Technical Report, KOM-TR-2006-08, Feb. 2007

Wang, S. Y.; Huang, C. H.; Lin, C. C.; Chou, C. L. & Liao, K. C. (2007-(a)). The Protocol Developer Manual for the NCTUns 4.0 Network Simulator and Emulator, July 2007.

Wang, S. Y.; Chou, C. L. & Lin, C. C. (2007-(b)). NCTUns Hompage. [Online]. Available: http://nsl10.csie.nctu.edu.tw/

Wang, S. Y.; Chou, C. L. & Lin, C. C. (2007-(c)). The GUI User Manual for the NCTUns 4.0 Network Simulator and Emulator, July 2007



InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821