

Adjusting Holdoff Algorithm Dynamically to Network Conditions for Improving Performance of Wireless Mesh Networks

Santong Li^{1, a}, Xuejun Tian^{1, b}, and Takashi Okuda^{1, c}

¹ Aichi prefectural University, Nagoya, Japan;

^aist.nagoya@gmail.com, ^btan@ist.aichi-pu.ac.jp, ^cokuda@ist.aichi-pu.ac.jp

Keywords: IEEE802.16, WiMAX, wireless mesh networks, distributed scheduling.

Abstract. Unlike Wi-Fi, BWA (Broadband wireless access) technology provides high-speed communication in a wide area. The IEEE 802.16d (WiMAX) standard of wireless mesh networks is one of the widely used BWA standard. WiMAX mesh mode achieves data transmission in conflict-free manner in multihop networks by using the control messages (three way handshake messages or MSH-DSCH messages) to reserve channel for sending data. Concurrently, coordination of three way handshake messages depends on the mechanism named Election based Transmission Timing (EBTT). However, IEEE 802.16d mesh mode uses a static holdoff algorithm, which leads to a low performance in the majority of cases. In this paper, after analyzing the IEEE 802.16d mesh mode with coordinated distributed scheduling, we propose a novel method to improve the throughput by a dynamic holdoff algorithm. The simulation results show that our proposal gets a better throughput performance.

Introduction

The BWA was approved by the IEEE (Institute of Electrical and Electronics Engineers) in December 2001, fixed wireless communication standards. As additional specifications, IEEE802.16-2004 that using the frequency band of 2 ~ 11GHz has been approved from 2004, also referred as WiMAX. Because of its high compatibility and cost-effectiveness, it is becoming gradually one of the important network technologies.

WiMAX contains two common modes, PMP mode and Mesh mode. In the PMP mode, traffic only occurs between the BS (base station) and SSs (Subscriber stations). While in the Mesh mode traffic can be routed through other SSs and can occur directly between SSs with a coordinated distributed scheduling (MSH-DSCH).

In the WiMAX mesh mode, in order to achieve communications between SSs in conflict-free, every node competes for channel access using election based transmission timing (EBTT) before sending data. Each node notifies reservation status to other neighbors within one-hop and two-hop by MSH-DSCH. The three-way handshake mechanism is employed for requesting, granting, and confirming available data subframe. If the frame interval between two consecutive MSH-DSCH messages of one node is long, the time required to complete the reservation of channel becomes long. In other words, packet delay will get long.

In order to improve the capacity of mesh by changing the frame interval between two consecutive MSH-DSCH messages, some holdoff algorithms for IEEE 802.16d mesh mode have been proposed[2,5]. According to IEEE 802.16d mesh standard [1], the static holdoff algorithm is employed in most instances. It means all of the BS and SSs with the same transmission holdoff exponent (Xmt Holdoff Exponent). When the Xmt Holdoff Exponent is too small or equal to 0, competition of channel reservation will happen frequently, thus it will waste more unnecessary time before reserving successfully. Otherwise, as Xmt Holdoff Exponent increases, although it can reduce competition times, but also increases the frame interval between two consecutive MSH-DSCH messages that the three-way handshake takes very long.

As other policies for improving performance of IEEE 802.16d mesh networks, optimal data scheduling policy and cross-layer policy have been developed [9,5]. However, data subframe consists of 256 mini slots and easily becomes saturated under high-density network conditions. The

improvement of policy based on optimal data scheduling is not significant. The problem of cross-layer policy is that end-to-end delay will increase by the undue overhead.

In [3], the authors introduce a new scheduling scheme to improve scalability and resource utilization in wireless mesh networks where all nodes compete to gain access to the transmission time slot. In their proposed method, the transmission time of a node is modified according to the transmission status of its neighbors (WCN-DSCH), resulting in a shorter holdoff time and reducing the transmission delay and throughput. However, WCN-DSCH only depends on the number of competing neighbor nodes and does not consider the state of the node. Thus it has a poor performance in a small number of nodes.

In [4], the authors proposed an adaptive holdoff algorithm (AHA) for IEEE 802.16d mesh mode with coordinated distributed scheduling. AHA improves network throughput by adaptively adjusting the transmission holdoff exponent according to current node state. In addition, it does not depend on information from other layer. However AHA does not consider the competition status with neighbor nodes, thus cannot be applied for other scenarios, such as full load.

In this paper, we propose a dynamic holdoff algorithm based on both of node state and neighbor (DHSN) for IEEE 802.16d mesh mode. Our proposal changes the holdoff exponent dynamically by the information related to the current node status and the number of competing neighbor nodes around it. Finally improves the network throughput.

The remainder of our paper is organized as follows. In section II, we introduce the IEEE 802.16d mesh mode. The DHSN is proposed in Section III. Section IV presents simulation results. Section V concludes our paper and presents further works.

The IEEE802. 16d Mesh Mode

In this section, we discuss the aspects of the IEEE 802.16d MAC protocol relevant to the mesh mode with distributed scheduling.

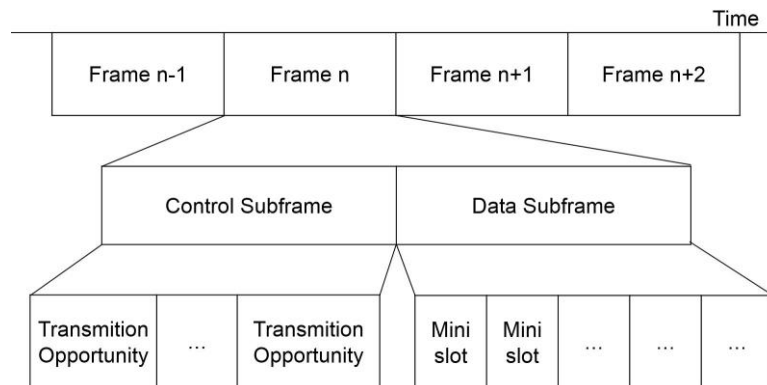


Figure 1: The structure of frame for IEEE 802.16d mesh mode

The Structure of Frame and of three Way Handshake Mechanism. In IEEE 802.16d mesh mode, the frame is based on time division multiple access (TDMA) frame structure, as shown in Fig.1. Each TDMA frame consists of a control sub-frame and a data sub-frame. Control sub-frames are partitioned into slots of fixed duration, which are accessed by nodes based on the distributed election procedure specified by the standard. The data subframe is situated at the end of the control subframe and it is divided into a fixed number of data minislots, up to 256. The number of transmission opportunities (TOs) in the control subframe can be a value between 0 and 15. Each transmission opportunity can carry one signaling message. There are two types of control subframes depending on their function as listed below: 1) Network Control Subframe, 2) Schedule Control Subframe. Network control subframe, as the name implies, help to distribute the network configuration parameters to neighbor nodes and also allow new nodes to synchronize themselves with the network, join an existing mesh network and establish logical links to neighbor nodes. The schedule control subframe is used to transmit signaling messages for scheduling the data sub-frame transmission

opportunities and is also split in two parts. The first part is for the messages that belong to the centralized scheduling mechanism and the second part is for MSH-DSCH messages that belong to the coordinated distributed scheduling mechanism.

To enable data transmission in a conflict-free manner, the three way handshake is employed by transmission of MSH-DSCH messages. As shown in the Table 1, MSH-DSCH messages are classified into three types: 1) Request, 2) Grant, 3) Confirm. The Request is used by a node for specifying its bandwidth demand for a specific link. The Grant is used by a node for sending available IE in connection with Request. The Confirm is used for confirming available IE and medium accesses of data subframe.

Table 1: Structure of MSH-DSCH message

Syntax	Size (bit)
MSH-DSCH - Message- Format()	-
Coordination Flag	8
Grant/Request Flag	1
Sequence counter	6
No.Requests	4
No.Availabilities	4
No.Grants	6
Reserved	2
if(Coordination Flag ==0)	-
MSH-DSCH Scheduling IE()	variable
for(i=0; i < No _ Requests; ++i)	
MSH-DSCH _ Request _ IE()	16
for(i=0; i < No _ Availabilities; ++i)	-
MSH-DSCH _ Availability _ IE()	32
for(i=0; i < No _ Grants; ++i)	-
MSH-DSCH _ Grant _ IE()	40
<i>g</i>	-

The EBTT Mechanism. The IEEE 802.16d mesh mode has not explicit schedule negotiation. The transmission timing is collision-free within the two-hop neighbourhood of each node. Furthermore, the mechanism is completely distributed and needs no central control.

In their MSH-DSCH messages, SSs only need to relay two parameters, Next Xmt Mx (*mx*) and Xmt Holdoff Exponent (*exp*). The XmtHoldoffTime (H) is the number of MSH-DSCH transmit opportunities after the last transmit time that a node is not eligible to transmit a MSH-DSCH message. The relation between H and *exp* can be presented as:

$$H = 2^{exp+4} \tag{1}$$

The Next Xmt Time (*nxmt*) is the next MSH-DSCH messages eligibility interval for this node and computed as the range:

$$2^{exp} * mx < nxmt \leq 2^{exp} * (mx + 1) \tag{2}$$

The Earliest Subsequent Xmt Time (*esxmt*) is the earliest time after the Next Xmt Time, that this node is eligible to transmit a MSH-DSCH message, and it can be calculated as:

$$esxmt = nxmt + H + 1 \tag{3}$$

As shown in Fig.2, an overview of medium access in the control subframe. Consider Node A which has a MSH-DSCH message for transmission and which has the Current Xmt Time (*cxmt*) within its eligibility interval. For determining its own *nxmt* during *cxmt*, the node sets the Temp Next Xmt Time (*tnxmt*) to *cxmt*+H+1. Then it has to determine the list of eligible competing nodes which is the set of all nodes in the physical-neighbor list by *tnxmt*. The eligible competing nodes have the following conditions:

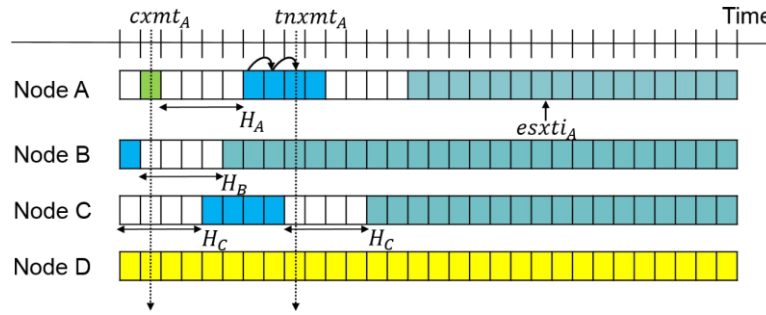


Figure 2: Medium access in the control subframe

A Mesh Election is held among this set of eligible competing nodes with three kinds of parameters: 1) $txmt$, 2) the local node ID, 3) node IDs of all eligible competing neighbor nodes. If the local node does not win the mesh election, system will set $txmt$ equal to the next MSH-DSCH transmission opportunity ($txmt = txmt + 1$). Otherwise, the local node is the winner of the mesh election and the $txmt$ is set to $txmt$. A detailed description of the Mesh Election algorithm can be found in [1].

The Problems of EBTT. IEEE 802.16d standard specifies a MAC protocol (EBTT) to coordinate the transmission of control messages in a collision-free manner.

As defined in the mesh mode of IEEE 802.16d specifications, due to a static holdoff algorithm, all of the Mesh Base Station (MS) and Subscribe Stations (SSs) use identical transmission exp regardless of the number of competing neighbor nodes and local node state. Therefore, when the number of competing neighbor nodes is small, lower utilization rate leads to lower performance. When the number of competing neighbor nodes is large, system cannot identify the busyness of nodes that also leads to a low performance.

The Dynamic Holdoff Algorithm Based on Competing Neighbors and Node State

In this section, we describe our motivation and neighbor management for introducing the proposed method at first. Then, we explain the approach of DHSN in IEEE 802.16d mesh networks.

The Parameters about Throughput. Throughput is an important indicator of network performance evaluation. In most instances, we can improve the overall performance of network by optimizing the algorithm. The throughput in IEEE 802.16d mesh mode can be calculated by the following formula:

$$\tau = \frac{D}{C+S+H} \quad (4)$$

In the formula (4), τ is the throughput, C is the competition time for getting the channel of sending data, S is the time that node transfers data and H is the holdoff time after sending data. C does not only depend on the number of competing neighbors, but also depends on the competitive frequency. We can calculate the H by formula (1). In addition, C will change with the H . Hence, they have an indirect relationship.

We can adjust dynamically the parameter of the denominator of the above formula according to the network status, e.g. the number of competitor neighbors or the number of messages that the node will send. Then, the performance of throughput can be improved.

Neighbor Classification and Busyness. In the IEEE 802.16d mesh mode with distributed scheduling, data transmission depends on MSH-DSCH messages that can reserve the minislots of the data subframe. Since all three kinds of control messages, request, grant and confirm, are encapsulated in the MSH-DSCH message, data transmission between nodes can occur directly and does not depend on central control manner.

Using the existing MSH-DSCH approach, we can find out the status of the node without the information from the other layer. As shown in Fig.3, a node can be classified by its number of request, grant and confirm messages. Then, busyness of a node can be calculated as a factor (Weight: ω).

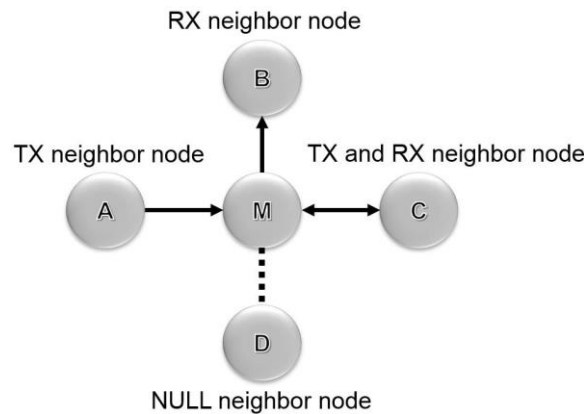


Figure 3: Classification of neighbor nodes

As shown in Fig.3, if a node, say node I, has a neighbor node A that tries to send data to it, called as a TX neighbor node, we know that the node I only needs to send a grant message to the TX neighbor node in three way handshake mechanism. If node I has a neighbor node B that want to receive date from it, called as a RX neighbor node, and according to the three way handshake mechanism, the node has to send two messages: request and grant. And so on, C is not only TX but also RX neighbor node and needs three messages to complete data transmission, D is called NULL neighbor node and needs to do nothing. Thereby, Weight can be calculated by the following formula:

$$\omega = RX_{exist} * 2 + TX_{exist} * 1 \quad (5)$$

If TX neighbor node or RX neighbor node exists, RX_{exist} or TX_{exist} is equal to 1. Otherwise, they are equal to 0. We can get a factor ω to determine roughly the busyness of a node and does not depend on the information from the other laye via the formula. Thereby, we propose a new algorithm according to the ω to improve throughput.

A Novel Method: DHSN. By discussing the formulas about throughput and Weight, we proposed a novel method that is called 'A Dynamic Hloff Algorithm based on Neighbor and Node state' or 'DHSN' in short. Considering the formula (4), we combine the two ideas in our proposal.

One is the number of competing neighbor nodes. If a node has many competitors, it needs a long holdoff time to avoid the failure for contention of channel reservation. Otherwise, the node needs a short holdoff time to reserve channel as much as possible for sending more data.

The other one is the state of the node. we only consider the busyness about the node and use a factor Weight to indicate clearly. If the node is busy, also means it has a lot of data to send, it needs a short holdoff time to complete the three way handshake. Otherwise, it has a long holdoff time to avoid the competition with the neighbor nodes.

For adapting to various network conditions, we try combining the two elements into the following formula:

$$\begin{cases} H = 2^{\log_{\omega+2} cn+2} & N \leq 9 \\ H = 2^{\log_{\omega+2} cn+1} & N > 9 \end{cases} \quad (6)$$

H is the holdoff time that after sending data. ω is the factor that represent the busyness of the local node. cn is the number of competitor neighbor nodes. N is the number of nodes. Our proposal considers three cases: a) if the nodes have the same busyness, the nodes that have more competitors will get a long holdoff time to avoid the competition with others. b) if the nodes have the same number of competitors, the busier nodes will get a short holdoff time to send more data. c) otherwise, the nodes will strike an optimal balance between the number of competing neighbors and statues.

Simulations

We use Network Simulator named NS2 to evaluate the proposed methods. The Computer Networking Group of the University of Pisa developed the patch [12] of WiMAX for NS2. This patch provides coordinated distributed mode of IEEE 802.16d mesh network. In some proposals, the voice over IP (VOIP) and video on demand (VOD) traffics are used for simulating the network as realistic as possible. Although the results of our proposal are better than the others with VOIP and VOD, the generation of traffic in other layers is not focused in this paper. Therefore, in order to evaluate our proposal easily, only CBR traffic is considered in simulations. In addition, we mix up NodeIDs randomly for getting a more real saturated condition. The packet size is set to 1000 bytes at first, and then the results are observed by changing packet size. The other settings of parameters are shown in Table 2.

Table 2: Simulation Parameters

Parameter	Value
Traffic rate	2Mbps
Frame Duration	10ms
Number of control slots per frame	4
Traffic	CBR
Warm up time	2s
Simulation duration	10s
Number of channels	1
Bandwidth	10MHz
Packet size	512, 1000, 2000 (bytes)
Topology	Grid-full
Number of nodes	4,9,16,25,36,49

Since our proposal combines the competing neighbors and node state, the appropriate condition is set to be saturated. Thereby, we use a dense network topology e.g. grid-full. As Fig.4 shows, the nodes have a logical connection between each other.

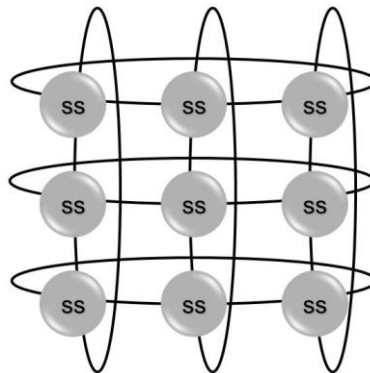


Figure 4: A scheme of the grid-full network topology with nine nodes

Owing to the structure of grid-full, every node has a large number of neighbors that affects the performance of the transmitting scheduling. The simulations with grid-full topology are executed by increasing the number of nodes. The simulation settings are listed in Table 2, the variation of nodes are 4, 9, 16, 25, 36, 49. Then the results are analyzed later.

The results of the simulations are shown from Fig.5 to Fig.9. The Fig.5 shows the frame interval between two consecutive MSH-DSCH messages. In IEEE 802.16d mesh mode, shorter frame interval means that node can return to competition and get the transmission opportunity (TO) sooner. Concurrently, shorter frame interval can improve the utilization ratio of control subframe, then enhance the performance indirectly. When the number of nodes is small, the control subframe cannot

be filled up due to that the static holdoff algorithm has a long holdoff time. However, the results of DHSN with an optimized holdoff algorithm are shorter than the static in all of situations.

The results of end-to-end (e2e) one-way delay is shown in Fig.6. It is also the one of important metrics for mesh network. The DHSN is shorter than the static because the unused slots of control subframe are reduced. However, e2e one-way delay grows with the number of nodes increase, DHSN makes the busier nodes can get transmission opportunity earlier than the idle nodes, thus the delay of DHSN grows slowly.

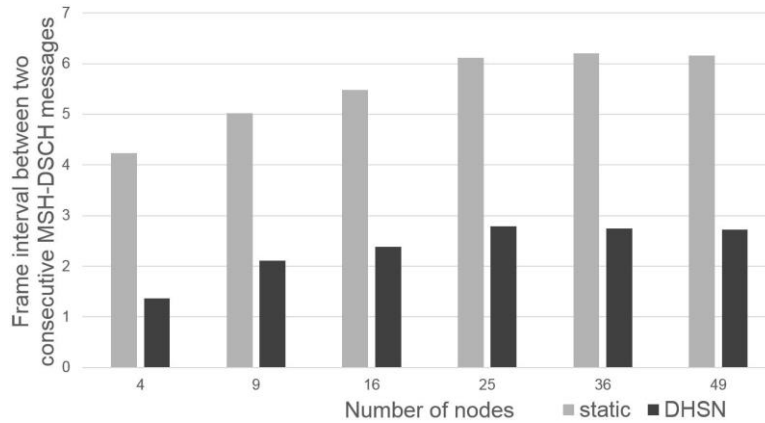


Figure 5: Frame interval between two consecutive MSH-DSCH messages

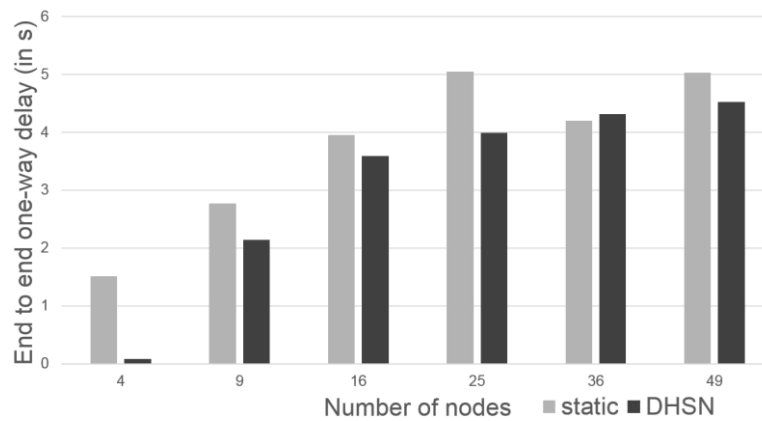


Figure 6: End to end one-way delay

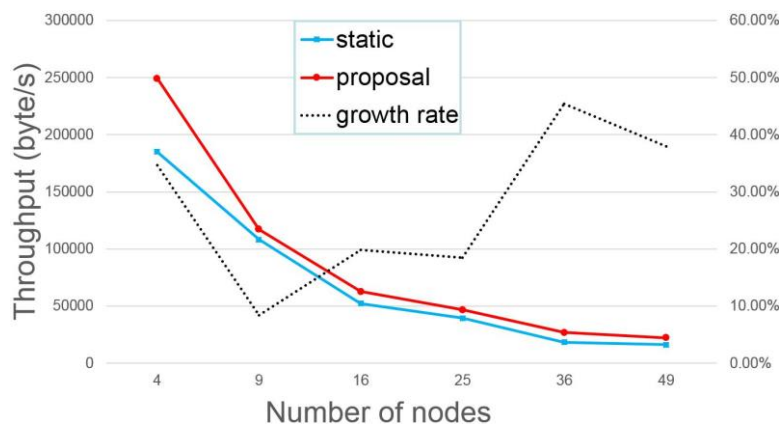


Figure 7: The average throughput with packet size 1000 bytes

Figure 7 shows the comparison of average throughputs with 1000 bytes packet size. As we predicted, our proposal which combines two ideas has a significant improvement than the static. DHSN does not only consider the competing neighbors, but also the node state. The dotted line

indicate the growth rate of throughput. Because of the improved utilization ratio of control subframe, the system gets a high throughput in less number of nodes. Concurrently, since the holdoff time is adjusted dynamically with the busyness of node, the improvement is more obvious as the number of nodes increases from 9.

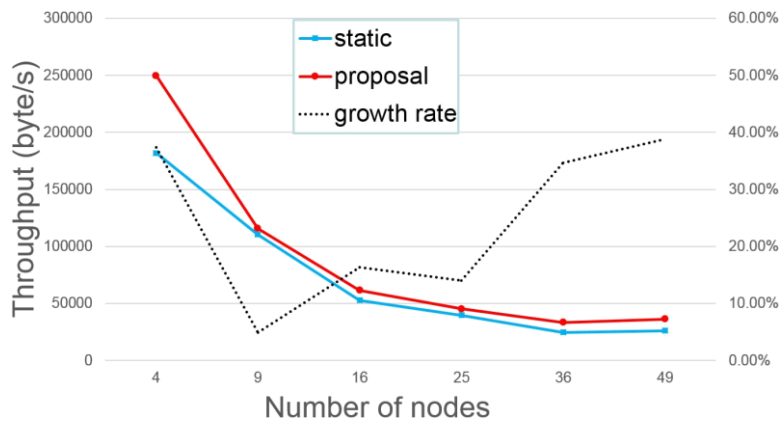


Figure 8: The average throughput with packet size 512 bytes

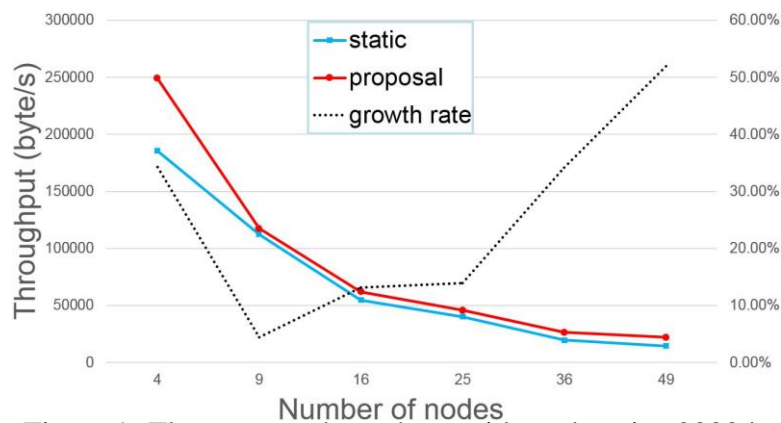


Figure 9: The average throughput with packet size 2000 bytes

The effects of packet size on throughput are indicated in Fig.8 (512 bytes) and Fig.9 (2000 bytes). The results show that, the performances of DHSN are better than static holdoff algorithm, which have the similar trends of growth in other cases. When the packet size is small, medium accesses in control subframe occur more frequently, which leads to more competitions. Otherwise less competitions. Thereby, as packet size increases, the growth rate shows a considerable increase. Within 512 bytes packet size, the growth rate reaches a maximum 38.88% in the case of 49 nodes. Within 1000 bytes packet size, the growth rate reaches a maximum 45.44% in the case of 36 nodes. Within 2000 bytes packet size, the growth rate reaches a maximum 51.97% in the case of 49 nodes. According to the results, DHSN can be suit for various conditions, such as different number of nodes or packet sizes.

Our proposal solved the problems of EBTT as mentioned in section II, thereby improved the throughput obviously. When the number of competing neighbor nodes is small, DHSN raises the utilization of control channel to a higher level by reducing the frame interval between two consecutive MSH-DSCH messages, thereby an obvious improvement can be observed with 4 nodes. With increase of the number of nodes, competitions for getting transmission opportunity increase accordingly. When the number of nodes is larger than 9, the utilization of control channel is saturated and more failure competitions lead to lower performance. Hence, reducing the frame interval between two consecutive MSH-DSCH messages blindly is unwise. DHSN can classify nodes by their busyness and plot out holdoff time properly, especially under high-density network conditions. Thereby, the performance of DHSN is better than static holdoff algorithm. In addition, the change of packet size will influence traffic capacities coordinated by each MSH-DSCH message. An efficient holdoff algorithm DHSN will enhance the improvement with increase of packet size.

Conclusion

In this paper, we proposed a dynamic holdoff algorithm based on neighbor and node state (DHSN) for IEEE 802.16d mesh mode. Through analyzing the EBTT mechanism, we combine two elements which influence the performance of networks to solve existing problem. The difference from the static holdoff algorithm is that, our proposal DHSN can adjust the holdoff exponent to network condition dynamically and accurately by the busyness of local node and the number of competing neighbor nodes. The simulation results show that, the throughput increases significantly by improving utilization ratio with less nodes, and the increases of throughput are obvious by discriminate busyness among nodes accurately with a large number of nodes. In addition, as packet size increases, the performance of DHSN becomes more effective.

As the future work, we will focus on further simulation, such as other types of topologies, utilisation of mini-slots of data subframe.

References

- [1] [1]IEEE 802.16-2007, DRAFT Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems (2007).
- [2] Min Cao, Wenchao Ma, Qian Zhang, Xiaodong Wang, Wenwu Zhu, Modelling and Performance Analysis of the Distributed Scheduler in IEEE 802.16 Mesh Mode (2007).
- [3] Bong Chan Kim, Dong Gu Kwak, Heecheol Song, Hwang Soo Lee, Joong Soo Ma Personal, An adaptive holdoff algorithm based on node state for IEEE 802.16 mesh mode with coordinated distributed scheduling (2008).
- [4] Sara Lakani, Hossein Ghaffarian, Mahmood Fathy, Bijan Raahemi, A Neighbor-based Holdoff Reduction Scheme for Distributed Scheduling in Wireless Mesh Networks (2010).
- [5] Nico Bayer, Bangnan Xu, Veselin Rakocevic and Joachim Habermann, Improving the Performance of the Distributed Scheduler in IEEE 802.16 Mesh Networks (2007).
- [6] Chee-Wei Ang, Multihop Interference and Multihop Propagation of Control Signaling in IEEE 802.16d Mesh Networks (2010).
- [7] Parag S.Mogre, Matthias Hollick, Ralf Steinmetz, The IEEE 802.16-2004 MeSH Mode Explained Technical Report KOM-TR-2006-08 (2006).
- [8] C. Cicconetti, I. F. Akyildiz, L. Lenzini, Bandwidth Balancing in Multi-Channel IEEE 802.16 Wireless Mesh Networks (2007).
- [9] Limin Peng, Suyun Sun, Coordinated Distributed Data Scheduling Scheme in IEEE 802.16 Mesh Networks (2011).
- [10] S.Lakani, M.Fathy, H. Ghaffarian, A New Approach to Improve the Efficiency of Distributed Scheduling in IEEE 802.16 Mesh Networks (2009).
- [11] You Lei, Yifan Wang, Yong Zhang, Anfeng MAO, Functional model and simulation of IEEE 802.16 mesh mode based on NS2 (2008).
- [12] Information on http://cng1.iet.unipi.it/wiki/index.php/Main_Page.
- [13] Lawrence Harte, Tomas Pazderka, Introduction to 802.16 Wimax, Wireless Broadband Technology, Operation and Services, Altheos, Inc, South San Francisco, 2006.