The Knowledge Engineering Review, Vol. 29:4, 452–462.
doi:10.1017/S0269888914000162

An efficient route scheduling mechanism for WiMAX network

MING-HUANG GUO[1], DER-JIUNN DENG[2], HORNG-TWU LIAW[1] and JONG HYUK PARK[3]

[1]*Department of Information Management, Shih-Hsin University, #1, Lane 17, Mu-Cha Road, Sec. 1, Taipei 116, Taiwan; e-mail: mhguo@cc.shu.edu.tw;*
[2]*Department of Computer Science and Information Engineering, National Changhua University of Education, No. 2, Shi-Da Road, Changhua 500, Taiwan; e-mail: djdeng@cc.ncue.edu.tw;*
[3]*Department of Computer Science and Engineering, Seoul National University of Science and Technology, 1 Gwanak-ro, Gwanak-gu, Seoul 151-744, Korea; e-mail: jhpark1@snut.ac.kr*

Abstract

WiMAX was proposed as the new wireless network standard in recent years. In the wide cover area of WiMAX environment, there are many nodes and interference. In order to obtain the efficiency, the avoidance to the interference is important. In this paper, we propose a two-tier cluster-based route scheduling mechanism on WiMAX Mesh mode. It achieves both fairness and efficiency inside and outside the cluster. The simulation result shows that our mechanism provides great improvement in average delay time and throughput than other research.

1 Introduction

With the growth of network, WiMAX is the new-generation telecommunication technology. It is based on IEEE 802.16 and provides access to metropolitan area networks (MANs). Because WiMax covers wider radio range, it can service more users than other networks. However, if there are too many users delivering data at the same time, the starvation in network capacity and radio interference might cause heavy degradation in network performance. To avoid the problems, in this paper, we propose a scheduling mechanism on the WiMAX Mesh mode. The proposed mechanism is built up on two-tier cluster-based routing tree. It not only avoids the interference but also maintains the efficiency of transmission. It also achieves both fairness and efficiency inside and outside the cluster. The simulation result shows that our mechanism provides great improvement in average delay time and throughput than other research.

The rest of the paper is organized as follows. In Section 2, some related works are discussed, and the proposed mechanism is introduced in Section 3. Section 4 provides analysis and discussion between our proposal and some related works. Section 5 concludes this paper.

2 Related works

When a user requests to send data, the network should decide the routing path for the transmission first. To find the proper route for the different source and destination pairs, some issues are considered and are necessary. In the WiMAX network, there are wider radio range and more users. If the radio interference is not counted in routing on the WiMAX network, the data communication might be disturbed or interrupted very often when there are many users in the network. Moreover, if the network capacity is not considered in routing, the transmission starvation might occur due to out of network bandwidth. Therefore, it is very important to find a route with no interference and enough network bandwidth.

Table 1 The characteristics of some related works with centralized routing schemes

Year	Reference	Interference avoidance	Based on routing tree	Route scheduled
2005	Fu's scheme (Fu *et al.*, 2005)	No	Yes	No
2005	Kim's scheme (Kim & Ganz, 2005)	No	No	Yes
2005	Salem's scheme (Salem & Hubaux, 2005)	Yes	No	No
2005	Wei's scheme (Wei *et al.*, 2005)	Yes	Yes	Yes
2006	Chen's scheme (Chen *et al.*, 2006)	Yes	No	Yes
2007	Han's scheme (Han *et al.*, 2007)	No	No	Yes

In the IEEE 802.16 medium access control layer, there are Mesh mode and PMP (Point-to-MultiPoint) mode (Sayenko *et al.*, 2007), and the channel access are classified into single-channel accessing and multi-channel accessing schemes (Du *et al.*, 2007). To provide better network performance and fair accessing in data transmission, the route scheduling is necessary. In previous research, because the communication cost is higher and covering nodes are limited in PMP mode, most routing strategies are based on Mesh mode. Moreover, the centralized scheme is easy to deploy and more efficient, most studies are built up on it. Among the route scheduling studies, there are centralized route scheduling schemes (Fu *et al.*, 2005; Kim & Ganz, 2005; Salem & Hubaux, 2005; Wei *et al.*, 2005; Cao *et al.*, 2006; Chen *et al.*, 2006; Yang *et al.*, 2006; Han *et al.*, 2007; Wang & Mutka, 2008; Wang *et al.*, 2008; Xergias *et al.*, 2008), and the distributed ones (Cao *et al.*, 2005; Chieh & Wang, 2007). In Table 1, the characteristics of some related works based on centralized routing strategies are listed.

In Table 1, the Wei's scheme (Wei *et al.*, 2005) provides all featuers: the scheme is built up on routing tree, and scheduling the route without interference. Han's scheme (Han *et al.*, 2007) proposed four scheduling schemes and found that the different sequence of nodes delivered would affect the transmission efficiency. In this paper, we propose a two-tier cluster-based route scheduling mechanism on Mesh mode. The proposal is a central routing algorithm with single-channel accessing. It not only avoids the interference but also maintains the efficiency of transmission. It also achieves both fairness and efficiency inside and outside the cluster. The proposed mechanism is introduced in the next section.

3 The cluster-based route scheduling algorithm for WiMAX network

In the proposal, we discuss the algorithms for two cases: the intra cluster and the extra cluster. The relationships between these two cases are depicted in Figure 1. In the coming sections, we will first introduce the routing tree creation for the intra cluster and extra cluster; then we explain the scheduling for the route in the intra cluster and extra cluster.

Let T be the uplink frame allocated by BS (Base Station), and all nodes in the network are separated into several different clusters. There is only one clusterhead as the root node in each cluster tree, and the clusterhead mananges all transmittion in the cluster. Let C represents a cluster and nodes in the cluster are numbered as $n = \{1, 2, 3, \ldots, m\}$. Therefore, the nodes in the cluster i is represented by $C_n^i = \{C_1^i, C_2^i, C_3^i, \ldots, C_m^i\}$. In this paper, because the proposal is based on IEEE 802.16d standard, every node in the cluster is fixed. The example topology for the proposal is shown in Figure 2, and the example topology with the single cluster and the cluster routing tree are given in Figure 3.

The symbols used in the proposal are defined in Table 2, and the time frame structure of the proposed mechanism is depicted in Figure 4.

Definition 1. *If* $C = \{C_n^i, t_n^i\}$ *is operable, if and only if there is at least one* $(C_n^j, t_n^j) \subseteq C, 1 \leqslant j \leqslant n$ *existed with any* $t \in [0, T]$.

Definition 2. *If* $U = \{(C_i, t_i)\}$ *is operable, if and only if there is at least one* $(C_i, t_i) \subseteq U, 1 \leqslant j \leqslant n$ *existed with any* $t \in [0, T]$.

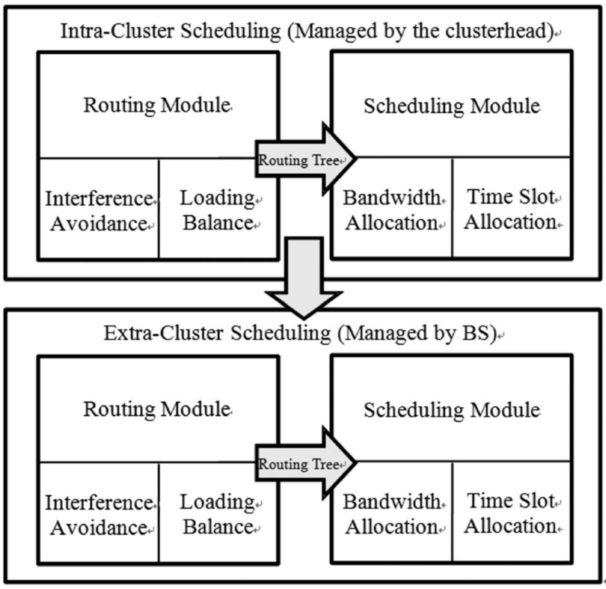

Figure 1 The relationships between the intra cluster and extra cluster in the proposal

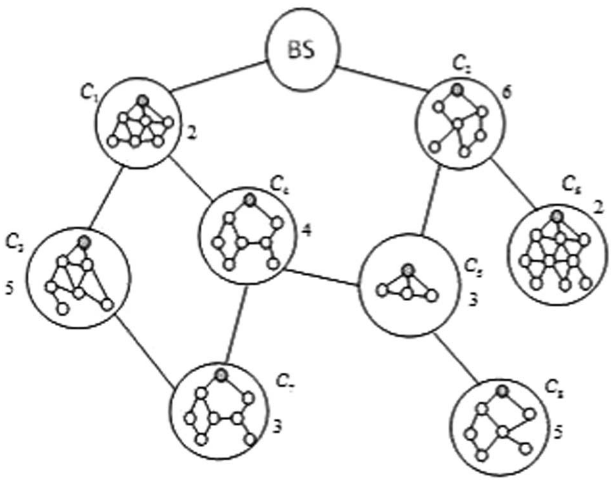

Figure 2 The network topology for the cluster-based structure

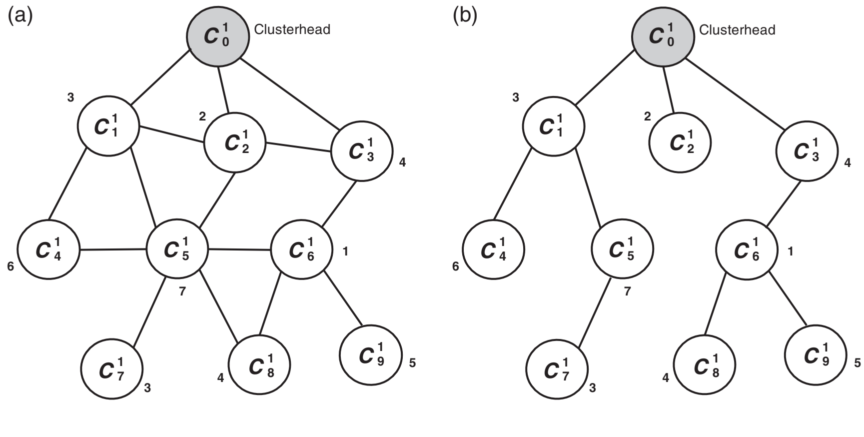

Figure 3 (a) The topology with single cluster. (b) The routing tree for the intra cluster

Table 2 Notations for the poposed algorithm

Notations	Statements
C_i	The cluster i in the network
T	The length of the frame
t_i	The summary of t_n^i for all n in C_i
t_n^i	The transmitting time for the node n in C_i
N	The number of nodes in the cluster
C_n^i	The node n in cluster i
hop_m^i	The distance in hops between C_i and the BS (Base Station)
I_p^i, k	The number of interfering nodes when C_p^i communicates with C_k^i
IC_j^i	The number of interfering nodes when C_i communicates with C_j
dl_p^i	The bandwidth requirements of C_p^i
$ICV_{p,k}^i$	The interference cumulation of C_p^i and C_k^i
DC_i	The bandwidth requirements of C_i
$CICV_j^i$	The interference cumulation of C_i and C_j
W_i	The weight of C_i
DW_i	The weight of data flow in C_i
r_{pk}^i	The transmission rate between C_p^i and C_k^i
rc_j^i	The transmission rate between C_i and C_j

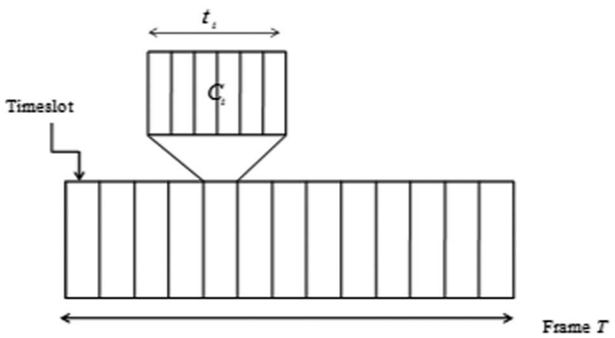

Figure 4 The time frame sructure for the poposed algorithm

By the definitions, it is not able to operate if any of the following conditions is not satisfied:

$$\begin{cases} \sum_{i=1}^{m} C_i \leqslant T \\ \sum_{i=1}^{n} C_n^i \leqslant t \end{cases}$$

3.1 The routing tree creation algorithm for the intra-cluster case

This algorithm is based on Wei's route construction algorithm for uplink (Wei *et al.*, 2005), and the routing tree is established with interference avoidance and bandwidth requirements. In the algorithm, $I_{p,k}^i$ is the number of interfering nodes when node p communicates with node k, and dl_p^i is the bandwidth requirements of node p. The algorithm is processed as follows:

1. The algorithm starts from the farthest node to the BS.
2. The number of interfering nodes for different route $I_{p,k}^i$ is computed.
3. The bandwidth requirements of interfering route $\sum dl_p^i$ is calculated.
4. $ICV_{p,k}^i = I_{p,k}^i \times \sum dl_p^i$
5. The smallest $ICV_{p,k}^i$ is selected for the route, and removing the node from the union.
6. If there is a clusterhead near a hop, the clusterhead is added in the route.
7. Repeat the previous steps until all nodes are in the routing tree.

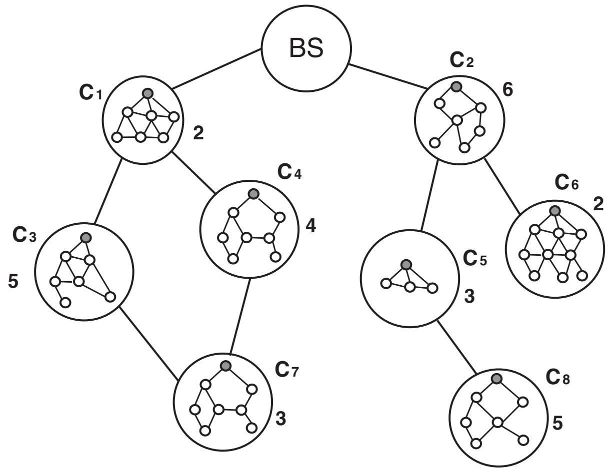

Figure 5 The results of the routing tree creation for the extra-cluster case

Let Figure 3(a) be the example network topology, the algorithm is run as below:

1. The farthest node to the clusterhead C_9^1 is selected.
2. According to the network topology, there is only one route for C_6^1 and C_9^1.
3. Then C_8^1 is selected to route, and there are two routes, C_5^1 and C_6^1, to select. The $I_{p,k}^i$ are 5 and 4, and $\sum dl_p^i$ is calculated to have $ICV_{5,8}^1$ as 58 and $ICV_{6,8}^1$ as 68. Therefore, C_6^1 is selected.
4. Repeat the previous steps to establish the complete route tree, as shown in Figure 3(b).

3.2 The routing tree creation algorithm for the extra-cluster case

This algorithm is similar to the one for the intra-cluster case. In the algorithm, IC_j^i is the number of interfering nodes when C_i communicates with C_j and dc_j^i is the bandwidth requirement between C_i and C_j. The algorithm is processed as follows:

1. The algorithm starts from the farthest node to the clusterhead.
2. The number of interfering nodes for different routes IC_j^i is computed.
3. The bandwidth requirements of interfering route $\sum dc_j^i$ is summaried.
4. $CICV_j^i = IC_j^i \times \sum dc_j^i$
5. The smallest value of cluster is selected for the route.
6. Repeat the previous steps until all nodes are in the route tree.

Let Figure 2 be the example network topology, the algorithm is run as below:

1. The farthest node to the clusterhead C_8^1 is selected.
2. According to the network topology, there is only route one for C_5^1 and C_8^1.
3. Then C_7^1 is selected. There are two routes C_3^1 andC_4^1. The IC_j^i values are 3 and 4, and $\sum dc_j^i$ is calculated to have $CICV_7^3$ as 30 and $CICV_7^4$ as 36. Therefore, the C_3^1 is moved to the route.
4. Repeat the previous steps until the complete routing tree is established, as shown in Figure 5.

Conbining the previous descriptions, Figure 6 lists the routing tree creation algorithm for the intra cluster and extra cluster. In the network, there are N nodes, and all nodes are processed in each round. Hence, the time complexity is $O(N^2)$.

3.3 The route scheduling algorithm for the intra-cluster case

This algorithm is modified from Han's transmission-tree scheduling algorithm (Kim & Ganz, 2005). The algorithm starts from the nearest node to BS and schedules the route with bandwidth requirements. The process in the algorithm is described as follows:

1. The algorithm starts from the nearest node to BS.

```
S←{0}                 // Initial the set of selected nodes, and let the node 0 as the root node.
H←{}                  // Initial the set of selected links.
N←{1, 2, 3,...,n}     // Initial the set of unselected nodes.
L←{1, 2, 3,...,l}     // Initial the set of unselected links.
While N ≠ Ø           // Repeat the steps until the set of unselected nodes is empty.
  M←D( N )            // The algorithm starts from the farthest node to the BS.
    P(j)←Nei(j)       // Check out all the links of node j.
    V←R * P(j)        // Compute the interference cumulation for the link.
    j←min I(k)        // Select the link with the minimum interference cumulation value.
    Add j to H        // Move the selected link to the set of selected links.
    Remove j from L   // Remove the selected link from the set of unselect links.
  Add M to S          // Add the node to the selected nodes.
  Remove M from N     // Remove the node from the unselected nodes.
END                   // The algorithm is end.
```

Figure 6 The routing tree creation algorithm for the intra cluster and extra cluster

Table 3 The interference table for the transmitting in the intra cluster during the first time slot

Selected node	Interference nodes
C_2^1	C_1^1, C_3^1, C_5^1
C_6^1	$C_2^1, C_3^1, C_5^1, C_8^1, C_9^1$
C_4^1	$C_1^1, C_2^1, C_5^1, C_7^1$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C_1^1							1	1	1									
C_2^1	1	1																
C_3^1												1	1	1	1			
C_4^1	1	1	1	1	1	1												
C_5^1												1	1	1	1	1	1	1
C_6^1	1																	
C_7^1							1	1	1									
C_8^1												1	1	1	1			
C_9^1							1	1	1	1	1							

Figure 7 The results for the route scheduling algorithm executed in the intra cluster

2. The cluster with minimum bandwidth requirements is put into scheduling first.
3. According to the routing tree, the node without interference is added into the scheduling then.
4. Repeat the previous steps until all clusters are processed.

Let Figure 3(b) be the example topology, the algorithm is run as below:

1. The algorithm starts from the nearest node to the BS. There are C_2^1, C_3^1, and C_4^1 in one hop range, and their bandwidth requirements are $dl_{1,2}^1 = 2$, $dl_{1,3}^1 = 3$, and $dl_{1,4}^1 = 5$, respectively. Because $dl_{1,2}^1$ is the minimum among the tree values, C_2^1 is put into the route scheduling process first.
2. When C_2^1 is delivering data, the interference nodes are C_1^1, C_3^1, and C_5^1. In order to clearly represent the interference for each node, the interference table is created and shown in Table 3, and the intra-cluster route scheduling results is shown in Figure 7. According to the table, there are only C_2^1, C_6^1, and C_4^1 nodes able to transmit data in time slot 1, without interference.

The proposed route scheduling algorithm for intra cluster is described in Figure 8. There are *N* nodes, and the time slot for each node is *t*. Because there are at most $n - 1$ interference nodes for each node, and the time complexity for the algorithm is $O(N^2)$.

```
S<-{0}                                                             //Initial the set of selected nodes, and let the node 0 as the root node.
N<-{1,2,3,…,n}                                                     //Initial the set of unselected nodes.
While N ≠ Ø && (t < MAX_T)                                         //Repeat the steps until the set is empty or the bandwidth capacity is full.
        M<-D(N)
        H<-F(N)                                                    //The algorithm starts from the nearest node to the BS.
        if (H! = null)
        M<-min_demand(M,H)                                         //Select the node with minimum bandwidth requirement.
        for(ts=1;ts<=timeslot;ts++)                                //Check out that if there is any inferference node for the selected node.
                for each(Cnode in Inferference_Matrix[ts])
                        If(M=Cnode)
                                Inferference=true
                        If(Inferference=false)
                                break
                        timeslot++
                        END
                END
        END
        if(Interference=false) {                                   //If there is no interference node, add the selected node into the scheduling
                for(ts=1;ts<timeslot;ts++)
                        Add node to ScheduleMatrix[ts]             //Add the selected node into the rout scheduling process.
                        Add node_neighbor to Interference_Matrix[ts]    //Denote the neighbor of the selected nodes as the interference nodes.
                        Add node_extended_neighbor to Interference_Matrix[ts]}  //Denote the neighbor of neighbor nodes as the interference nodes.
                END
        else
                continue
        END
        Add M to S
        Remove M from N
END
```

Figure 8 The route scheduling algorithm for the intra-cluster case

Table 4 The interference table for the transmitting in the extra cluster during the first time slot

Selected node	Interference nodes
C_2	C_1, C_5, C_6
C_3	C_1, C_7, C_4

3.4 The route scheduling algorithm for the extra-cluster case

This algorithm is similar to the intra-cluster scheduling one. The clusters are assigned with weights that is the main difference between the two algorithms. The process in the algorithm is introduced as follows:

1. The algorithm starts from the nearest node to BS, and the weight for evey cluster is calculated as $DW_i = hop^i_m \times dc^i_j \times W_i$.
2. The largest DW_i is seleced first, and the links without interference are added in the route scheduling process.
3. According to the routing tree, the node without interference is added into the scheduling then.
4. Repeat the previous steps until all clusters are processed.

Let Figure 5 be the example topology, and the algorithm is run as follows. In the examples, let the weights of $C_1 - C_8$ be 5, 3, 3, 6, 1, 4, 2, and 5, respectively.

1. The cluster C_1 and C_2 are in one hop range. Because their DW_i are 10 and 18, C_2 is scheduled first. When C_2 is delivering data, the interference nodes are C_1, C_5, and C_6.
2. The interference table in transmitting is created in Table 4, and the results for the algorithm are shown in Figure 9.

The proposed route scheduling algorithm for the extra cluster is described in Figure 10. There are N nodes, and the time slot for each node is t. Because there are at most $n-1$ interference nodes for each node, the time complexity is $O(N^2)$.

4 Perforamnce analysis

In this section, the performance for the single cluster and multi cluster are simulated and analyzed. The simulations are run by NS-2, and nodes are randomly distributed in the range of 100×100 units. Every

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C_1							1	1	1									
C_2	1	1																
C_3												1	1	1	1			
C_4	1	1	1	1	1	1												
C_5												1	1	1	1	1	1	1
C_6	1																	
C_7							1	1	1									
C_8												1	1	1	1			
C_9							1	1	1	1	1							

Figure 9 The results for the route scheduling algorithm executed in the extra cluster

```
S<-{0}                                                      //Initial the set of selected nodes, and let the node 0 as the root node.
N<-{1,2,3,…,n}                                              //Initial the set of unselected nodes.
While N ≠ Ø && (t < MAX_T)                                  //Repeat the steps until the set is empty or the bandwidth capacity is full.
        M<-D(N)
        H<-F(N)                                             //The algorithm starts from the nearest node to the BS.
        if (H! = null)
        DW<-min_dw(M,H)                                     //Select the node with minimum bandwidth requirement.
        for(ts=1;ts<=timeslot;ts++)                         //Check out that if there is any inferference node for the selected node.
                for each(Cnode in Inferference_Matrix[ts])
                        If(M=Cnode)
                                Inferference=true
                        If(Inferference=false)
                                break
                        timeslot++
                        END
                END
        END
        if(Interference=false) {                            //If there is no interference node, add the selected node into the scheduling
                for(ts=1;ts<timeslot;ts++)
                        Add node to ScheduleMatrix[ts]      //Add the selected node into the rout scheduling process.
                        Add node_neighbor to Interference_Matrix[ts]     //Denote the neighbor of the selected nodes as the interference nodes.
                        Add node_extended_neighbor to Interference_Matrix[ts]}  //Denote the neighbor of neighbor nodes as the interference nodes.
                END
        else
                continue
        END
        Add M to S
        Remove M from N
END
```

Figure 10 The route scheduling algorithm for the extra-cluster case

node in the network can send the data limited to five hops, and the transmission time for the intra-cluster case is 2 units, and the one between the clusters and BS is 8 units. In the single cluster, the simulations are examined with different number of nodes, 5, 10, 15, 20, and 25 nodes, in a cluster. In the multi cluster, there are five nodes in each cluster, and different number of clusters, two, four, and six clusters are assigned in the simulations. The following simulations compare the performance of proposed algorithm and the traditional IEEE 802.16d, and the results are showed in next sections.

4.1 The average delay time analysis for the single cluster

In this analysis, the average delay time is measured by seconds. When there are more nodes in the cluster, the network traffic will increase, and the average delay time will gradually increase too. In Figure 11, the simulation results show that our proposal does work and can provide lower average delay time than the traditional IEEE 802.16d.

4.2 The throughput analysis for the single cluster

In this analyais, the throughput is the packets received by a node in the unit time. When there are more nodes in the cluster, there will be more traffic in the limited network capacity, and the number of packets that every node can receive will be decreased. In Figure 12, the simulation results show that our proposal provides better route arrangement, and achieve better throughput than the traditional IEEE 802.16d.

4.3 The average delay time analysis for the multi cluster

When there are more clusters in the network, it is obvious that the network traffic will increase, and the average delay time will increase too. In Figure 13, the simulation results show that our proposal can still provide lower average delay time than the traditional IEEE 802.16d in the multi cluster network topology.

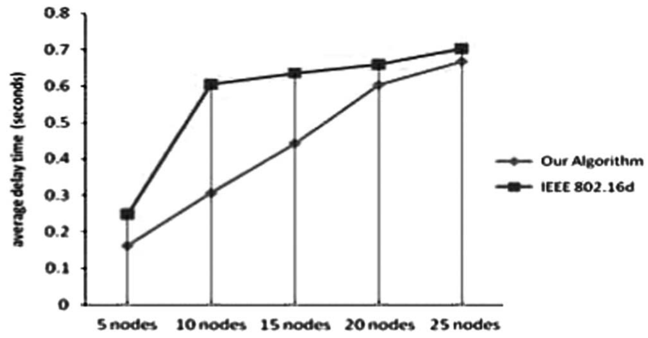

Figure 11 The average dealy time for different nodes in the single cluster

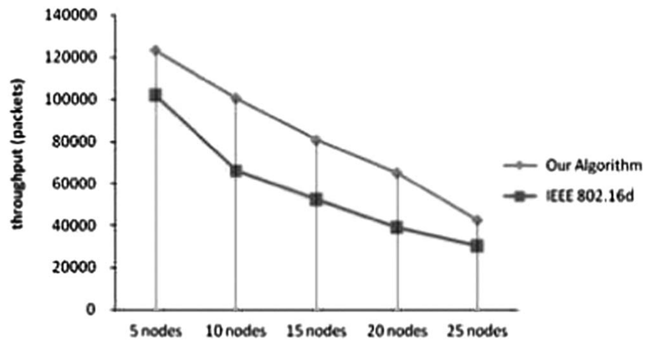

Figure 12 The throughput for differenet nodes in the single cluster

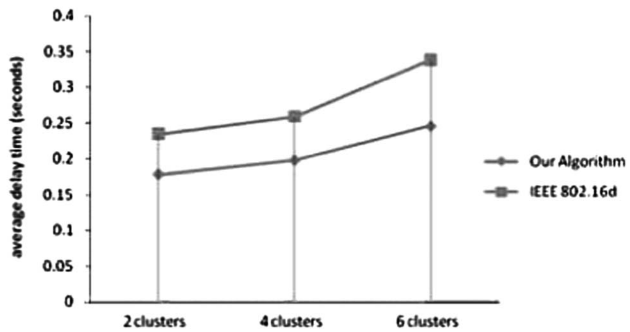

Figure 13 The average delay time for different clusters in the multi cluster

4.4 The throughput analysis for the multi cluster

When there are more clusters in the network, there will be more traffic in the limited network capacity, and the throughput of network will be decreased. In Figure 14, the simulation results show that our proposal provides better route arrangement, and achieve better throughput than the traditional IEEE 802.16d.

According to the previous simulation results, our proposal did provide better performance than the traditional IEEE 802.16d. Comparing with the related works, our algorithm and Wei's scheme (Wei *et al.*, 2005) are all developed with interference avoidance, route scheduling, and based on route tree. However, our proposal schedules the route with bandwidth requirements, but Wei's scheme (Wei *et al.*, 2005) did not. Therefore, our algorithm provides not only more considerations in route scheduling but also better performance.

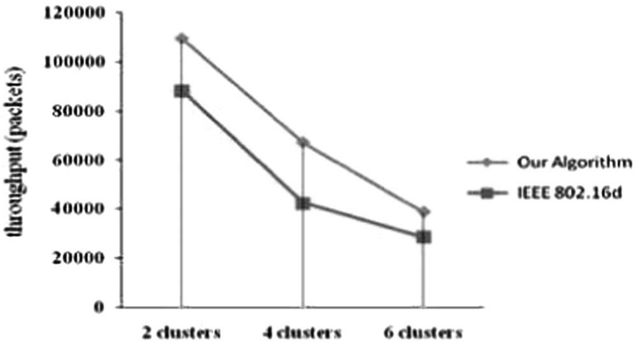

Figure 14 The throughput for different clusters in the multi cluster

5 Conclusion

WiMAX is the new-generation telecommunication technology to provide access to MANs. As it covers wider radio range, it can service more users than other networks. To provide better network performance, in this paper, we propose a scheduling mechanism on the WiMAX Mesh mode. The proposed mechanism is built up on two-tier cluster-based routing tree. It not only avoids the interference but also maintains the efficiency of transmission. It also achieves both fairness and efficiency inside and outside the cluster. The simulation result shows that our mechanism provides great improvement in average delay time and throughput than other research.

In the future, in order to provide the pervasive networking, the proposal should be modified to be deployed on WiMAX PMP mode. Moreover, additional performance evaluations should be done to compare with other research or improve the mechanism.

Acknowledgements

This research was supported in part by the MKE (The Ministry of Knowledge Economy), Korea under the ITRC (Information Technology Research Center) support programme supervised by the NIPA (National IT Industry Promotion Agency; NIPA-2010-C1090-1031-0004), and the ROC NSC under contract numbers NSC97-2221-E-128-005-MY3.

References

Cao, M., Ma, W. & Wang, X. 2005. Modeling and performance analysis of the distributed scheduler in IEEE 802.16 Mesh mode. In *The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 78–89.

Cao, M., Raghunathan, V. & Kumar, P. R. 2006. A tractable algorithm for fair and efficient uplink scheduling of multi-hop WiMAX Mesh networks. In *The IEEE Workshop on Wireless Mesh Networks*, 93–100.

Chen, J., Chi, C. & Guo, Q. 2006. An odd-even alternation mechanism for centralized scheduling in WiMAX Mesh network. In *The IEEE Global Telecommunications Conference*, 1–6.

Chieh, T. T. & Wang, C. Y. 2007. Routing and admission control in IEEE 802.16 distributed Mesh networks. In *The International Conference on Wireless and Optical Communications Networks*, 1–5.

Du, P., Jia, W., Huang, L. & Lu, W. 2007. Centralized scheduling and channel assignment in multi-channel single-transceiver WiMAX Mesh network. In *Wireless Communications and Networking Conference*, 1734–1739.

Fu, L., Cao, Z. & Fan, P. 2005. Spatial reuse in IEEE 802.16 based wireless Mesh networks. In *The IEEE International Symposium on Communications and Information Technology*, 1358–1361.

Han, B., Jia, W. & Lin, L. 2007. Performance evaluation of scheduling in IEEE 802.16 based wireless Mesh networks. In *Computer Communication*, 782–792.

Kim, D. & Ganz, A. 2005. Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems. In *International Conference on Broadband Communications, Networks, and Systems*, 895–901.

Salem, N. B. & Hubaux, J. P. 2005. A fair scheduling for wireless Mesh networks. In *The IEEE Workshop on Wireless Mesh Networks, Section III*.

Sayenko, A., Alanen, O. & Hämäläinen, T. 2007. Scheduling solution for the IEEE 802.16 Base Station. In *Computer Networks*.

Wang, B. & Mutka, M. 2008. Path selection for mobile stations in IEEE 802.16 multi-hop relay networks. In *World of Wireless, Mobile and Multi-Media Networks*, 1–8.

Wang, J., Jia, W. & Huang, L. 2008. An efficient centralized scheduling algorithm for IEEE 802.16 multi-radio Mesh networks. In *The 2nd International Conference on Ubiquitous Information Management and Communication*, 1–5.

Wei, H., Ganguly, S., Izmailov, R. & Haas, Z. 2005. Interference-aware IEEE 802.16 WiMAX Mesh networks. In *The IEEE Vehicular Technology Conference*, 3102–3106.

Xergias, S., Passas, N. & Salkintzis, A. K. 2008. Centralized resource allocation for multimedia traffic in IEEE 802.16 Mesh networks. *Proceedings of the IEEE* **96**, 1, 54–63.

Yang, C., Liu, Z. & Yang, Y. 2006. A centralized scheduling algorithm based on multi-path routing in WiMAX Mesh network. In *International Conference on Wireless Communications Networking and Mobile Computing*, 1–4.