Fair End-to-end Bandwidth Allocation (FEBA) algorithms observation and improvement

H.N. Truong, T.T.T. Do, D. Tran

Department of ECE, National University of Singapore
g0800446@nus.edu.sg, dothutrang@nus.edu.sg, tranduong@nus.edu.sg

Abstract—**IEEE 802.16 standard is specially designed for backhaul Wireless Mesh Networks (WMNs) to solve problems of scalability. However, bandwidth allocation problem is still unsolved. Fair End-to-end Bandwidth Allocation (FEBA) algorithm is one solution proposed for this problem. In this paper, analysis of FEBA for IEEE 802.16 mesh network is conducted. Some advantages and disadvantages in implementing FEBA are discussed in details as well. Two ways to improve FEBA are proposed: prioritizing the traffic flows and differentiating the QoS for each node in the network. One QoS differentiation scheme is also suggested for IEEE 802.16 Mesh Network.**

***Index Terms*— bandwidth balancing, fairness index, FEBA, IEEE 802.16, QoS, WMNs.**

I. Introduction

Wireless Mesh Networks (WMNs) are dynamically self-organized and self-configured networks, in which the nodes automatically form an ad hoc network and maintain the mesh connectivity. WMNs are undergoing rapid development progress and commercialization due to their several advantages such as low up-front cost, easy maintenance, robustness, etc. In WMNs, mesh routers form the mesh backbone for mesh clients and play the important role to improve the flexibility of mesh networking. Mesh routers also function as gateway/bridge to integrate WMNs with other networks. WMNs are very similar to ad-hoc networks because both networks use wireless multi-hop communication. Therefore, through multi-hopping like in ad-hoc networks, WMNs can give the same coverage using a mesh router with much lower transmission power. However, WMNs also have different characteristics compared to ad-hoc networks: mesh routers have minimal mobility and no energy constraint. Thus, WMN is not another type of ad-hoc networks but in fact it diversifies the capabilities of ad-hoc networks.

One research challenge on WMNs is scalability of the network. Recently, some research has been conducted on problems such that limited transmission range, low performance, poor fairness, and decreased network capacity for increasing number of nodes and traffic flows, or larger network. To solve these problems, IEEE 802.16 standard is one of industrial standards groups designed for WMNs. There are two tiers in WMN architecture: *backhaul* and *access*. Backhaul tier consists of mesh routers which create a multi-hop ad-hoc network. IEEE 802.16 standard focuses on the backhaul tier of WMNs and utilizes the TDMA MAC protocol to coordinate nodes in *mesh mode* operation. The MAC protocol in IEEE 802.16 distributed mode ensures the collision-free transmission of control messages. However, the bandwidth allocation problem is unsolved.

In [1], the Fair End-to-end Bandwidth Allocation (FEBA) algorithm for IEEE 802.16 is proposed. Negotiating bandwidth among nodes in a multi-channel network, FEBA provides the fair bandwidth allocation to end-to-end traffic flows in terms of throughput. Furthermore, utilizing control messages such as bandwidth requests and grants provided in IEEE 802.16, FEBA assigns the bandwidth at each neighbor in round robin fashion. The effectiveness of FEBA has been shown in performance evaluation through extensive simulations of different network configurations and system parameters. However, the results are mainly focused only in topology (chain/star) and dense population. Therefore, it is necessary to have further investigation on the impact of other system parameters. In this paper, we are going to investigate the influence of traffic flow priority, estimation weighting (α), and QoS end-to-end delay. Some improvement is also presented.

This paper is organized as follows. Section II presents the importance of bandwidth balancing in WMNs. Section III describes the FEBA algorithm. Section IV shows the observation of performance, advantages and disadvantages of FEBA. The improvement is presented in section V. And we conclude the paper in section VI.

II. Bandwidth balancing Role in WMNs

In networking, load balancing is a technique to spread work between two or more network nodes, links, gateways, or other resources, in order to get optimal throughput, response time, and optimal resource utilization. In wireless mesh networks (WMNs), load balancing is critical to obtain efficient utilization of the network capacity. Load imbalance in WMNs may be classified as gateway loading, center loading, and bottleneck node formation. Gateway loading imbalance occurs because of traffic aggregation at the gateway nodes and bandwidth constraint. It may result in congestion, packet loss, and buffer overflow. Center loading imbalance arises primarily due to shortest path routing. In the shortest path routing schemes, nodes which are close to the center of the network become points of contention, and usually exhaust their resources before the others, like bandwidth, processing power and memory storage in static WMNs, or battery energy reserve in portable WMNs. Similar to center loading, certain nodes at critical positions in the WMN also form network bottlenecks, that causes bottleneck node formation load imbalance. Load balancing, therefore, is essential to improve throughput and scalability of the WMN [3].

Load imbalance occurs even when the WMN has uniform data generation rate and uniform node distribution. There exist several issues for load imbalance, which include traffic stability arising out of fluctuations due to utilization of load balancing schemes, bandwidth stripping and effects to higher layer protocols, and so on.

In WMN load balancing, bandwidth plays an important role. It together with network capacity is the two main resources that limit the WMN. Bandwidth refers to the data rate through the radio interface, and can be understood simply as the throughput of end-to-end paths in the networks [1]. With more balanced bandwidth usage, higher throughput for nodes is achievable.

Bandwidth balancing implies the fairness of bandwidth allocation in the network [1]. Therefore, it is often measured by fairness index. One approach to calculate the fairness index is as following:

If a system allocates resources to n contending users, such that the i^{th} user receives an allocation x_i, then fairness index is:

$$\frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n} x_i^2} = \frac{E(x)^2}{E(x^2)} \qquad (1)$$

For the above fairness index formula, fairness index always lies between 0 and 1. For example, a distribution algorithm with a fairness of 0.10 means it is unfair to 90% of the users. Fairness does not necessarily mean equal distribution of resources. In some cases, it is justifiable to give more resources to some consumers than others. Therefore, fairness index is just one appropriate performance metric and would be the ratio of the resource allocated and the right for allocation [2]. For some load-balancing schemes, load is defined as a flow from one source node to one destination node, and the size of the load is defined as the bandwidth of the associated request. Therefore, load-balancing sometimes may be understood as bandwidth balancing. However, as aforementioned, other limited resources, such as network capacity may also be utilized. Furthermore, not only is fairness index utilized as degree of load-balance metric, other metrics such as bandwidth blocking rate may also be utilized in performance evaluation.

III. FEBA

Fair End-to-end Bandwidth Allocation (FEBA) algorithm is designed for IEEE 802.16 nodes to negotiate bandwidth in a multi-channel environment [1]. By assigning bandwidth requests and grants at each neighbor in a round-robin fashion, with an amount of service proportional to the number of traffic flows going to or coming from the neighbor, respectively, FEBA can provide a fair bandwidth allocation, in terms of throughput, to end-to-end traffic flows regardless of their path length.

A. *IEEE 802.16 Mesh – Channels, Frames and Subframes*

In IEEE 802.16 Mesh, time is divided into frames of fixed duration. Each frame comprises two sub-frames: a control sub-frame and a data sub-frame (Fig. 1). Control sub-frames are partitioned into slots of fixed duration (named control slots). Control messages can be sent in the control slots by nodes in a regular, though not periodical, manner. Although IEEE 802.16 mesh networks can utilize up to 16 multiple non-interfering channels for data transmission, control messages only can be sent in the same channel for all the nodes, e.g., channel Ch1 in Fig. 1. Data sub-frames comprise a fixed number of data mini-slots (named slots).

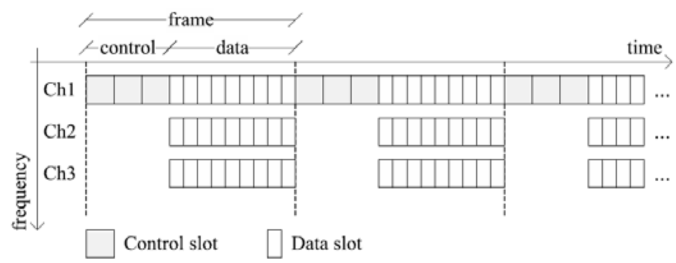

Fig. 1. Example of frame structure, with three channels.

Nodes use Mesh Distributed Schedule (MSH-DSCH) messages for bandwidth negotiation. MSH-DSCH messages contain a list of information elements (IEs), classified by the IEEE 802.16 standard into four types: *request IEs, grant IE*s, *confirmation IE*s and *availability IE*s. Negotiation procedure is a three-way handshake procedure: (i) requester node asks a neighbor node, granter, to allocate some bandwidth; (ii) the granter advertises a set of slots as 'granted' to the requester; (iii) the requester confirms that it will actually use that set of slots (or part thereof) to transmit data.

B. *Bandwidth request/grant in FEBA*

A simple network example in the Fig. 2 can be used to explain the request/grant data structures, in which neighbor nodes are connected by an edge; and active flows are represented by dashed arrows. The data structures used for bandwidth negotiation at node X also are illustrated. R and G are represented for Requesting and Granting queues respectively.

A weight ϕ_i is assigned for each active queue *i* (requesting and granting), which is computed so that the amount of service is proportional to the number of traffic flows under service:

$$\phi_i = \frac{\sum_{j \in A} I_i(j)}{|A|} \tag{2}$$

Where A is the set of all active traffic flows served by this node, *j* is an active flow, and $I_i(j)$ is an indicator function which equals 1 if j is under service at queue *i*, 0 otherwise. Since each traffic flow is under service at exactly one queue, $\sum_i \phi_i = 1$.

As can be seen from Fig. 2, X has 3 neighbors, i.e. A, B, C. Totally, there are 10 flows going through X. Therefore, C granting queue has $\phi_C^G = 2/10$, while the weight of C requesting queue is $\phi_C^R = 1/10$. This is because there are only 2 requesting flows coming from C to X, i.e. C-X and C-X-B, while there is only 1 requesting flow from X to C.

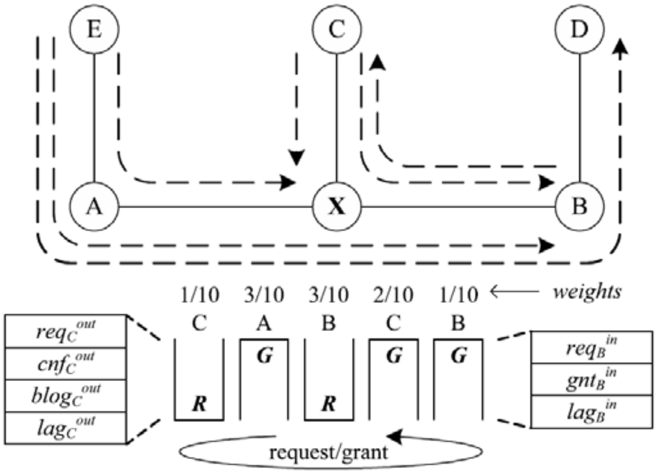

Fig. 2. Bandwidth request/grant data structure at node X.

Each queue *i* is entitled to grant/request up to $\phi_i F_{RR}$ bytes to/from neighbor *i*, where F_{RR} is target round duration system parameter. The granter should not schedule data slots too early, i.e. before the requester has the chance to confirm them, and not too late, which would entail longer response times of the handshaking procedure and greater per-hop packet transmission delays.

The bandwidth request/grant procedures are executed in a round-robin manner over all the queues in the active list until any of the following conditions becomes true: (i) the active list becomes empty; (ii) there is not enough space left in the control slot to add another IE to the MSH-DSCH message; (iii) the lag of the queue under service exceeds lag_{max}.

The procedure includes the following steps: (i) Randomly select a channel; (ii) Find the first available slot in the first frame of the grant horizon; (iii) If no slots are available, move to the next channel; (iv) If all channels have been searched, move to the next available frame of the grant horizon.

Fig. 3 is a simple example of this procedure. Assume that the granter needs to assign three slots for the transmission of its i^{th} neighbor. It first randomly selects one channel between the two available ones, say channel 1. Then, it visits the grant horizon from the earliest frame, i.e. x in Fig. 5. Slots 3 and 4 in channel 1 are available, thus a two-slot grant is issued. Since there is still one slot to be granted, the granter continues searching for available slots. Note that slot 4 in channel 2 cannot be granted anymore, since it overlaps in time with slot 4 in channel 1. Thus, slot 1 of frame y in channel 1 is granted instead, which completes the procedure. In the example the following grant IEs are thus added to MSH-DSCH: ⟨[3, 4], [x, x], 1⟩ and ⟨[1, 1], [y, y], 1⟩.

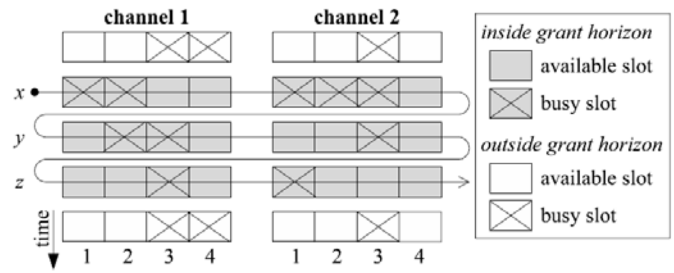

Fig. 3. Example of grant allocation of data slots within the grant horizon.

IV. FEBA OBSERVATION, ADVANTAGES AND DISADVANTAGES

A. Authors' observation

The author of FEBA compared FEBA with Greedy (which means that each node can assign as much bandwidth as possible) in term of end-to-end throughput, MAC throughput, fairness. When the number of nodes increases, end-to-end throughput for FEBA dramatically decreases, which is higher than that of Greedy. Meanwhile, FEBA achieves a lower MAC throughput than Greedy.

However, FEBA/DRR (Deficit Round Robin scheduler) attains almost perfect fairness among all the traffic flows in all cases, while the Greedy/DRR is only fair when the number of nodes is below seven.

In addition, because FEBA is designed for multi-channel used, the authors also observed the improvement of end-to-end throughput when the number of channel increases.

B. Advantages

In wireless mesh networks, the end-to-end throughput of traffic flows depends on the path length, i.e. the higher the number of hops, the lower becomes the throughput.

The main idea of FEBA is to assign bandwidth requests and grants at each neighbor: in a round-robin fashion; with an amount of service proportional to the number of end-to to-end flows going to or coming from the neighbor respectively; within a time window whose size depends on the frequency with which the neighbor transmits control messages.

FEBA gives a very high bandwidth fairness index, which is used to measure bandwidth balancing. The "Fairness" is a desirable property for any MAC protocol employed as the backhaul tier of a WMN.

C. Disadvantages

While fairness is the outstanding feature of FEBA, MAC throughput and average delay are the two drawbacks.

In the paper, the authors affirmed the MAC throughput weakness of FEBA, which is lower than Greedy. However, the authors did not compare FEBA and Greedy in terms of average end-to-end delay to verify whether FEBA can give a faster response.

In order to evaluate this, we setup a set of simulations.

1) Scenario setup

We use the same setup parameters as specified in the system profile profP3_10[8]:

- Channel bandwidth = 10Mhz
- Frame duration 4ms
- The nodes use 16-QAM-1/2 MCS unless stated otherwise
- Each traffic flow employs a 100kB buffer

We run each simulation 10 times independently and take the average. Each simulation lasts 120s with a 30s warm-up period as in [1]

2) Simulation and results

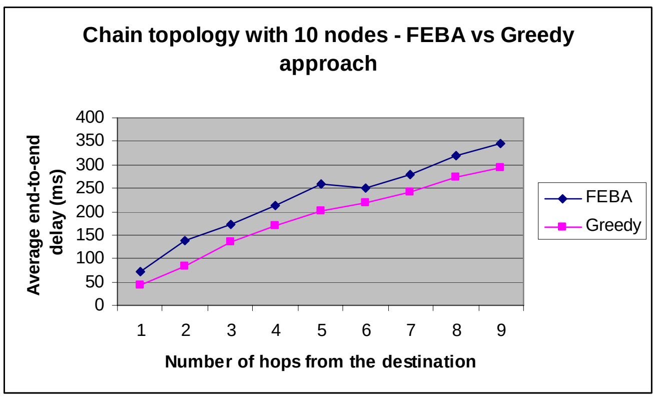

Fig. 4. Average end-to-end delay versus number of hops from destination for FEBA and Greedy

We use a chain topology with 10 nodes. Each node sends an Internet traffic flow (with 192-byte packet size) at the rate 100000b/s to the destination. Here, we use QPSK-1/2 MCS.

From the simulation result, it is clear that the delay is affected by the distance from the source to the destination, as the average end-to-end delay increases when the number of hops increases. We can notice that FEBA has a longer delay compared to Greedy approach. This is because FEBA has to tradeoff between delay and fairness among traffic flows.

D. Influence of α on grant horizon and FEBA performance

1) Problem

One of the important ideas which the authors propose in [1] is the grant horizon. The grant horizon is defined as the range of frames where the granter is allowed to grant slots for data to be transmitted by the requester. In FEBA, each node uses a different grant horizon, depending on the neighbour's control latency – *h* (the interval between two consecutives turns to transmit an MSH-DSCH message). As *h* is not known *a priori*, Cicconetti et al. use the exponential weighted moving average (EWMA) technique to estimate it:

$$h_i^{+} = \alpha \cdot h_i^{-} + (1-\alpha) \cdot h_i^{sampled} \qquad (3)$$

The authors claimed that when α = 0.1, FEBA can produce the most accurate estimations for h. In this section we will try to evaluate the performance of FEBA on changing the value of α. From the simulation we want to know whether the author's claim is valid.

2) Simulation

Figure 5 shows us how the estimation error varies with different values of α. It is noteworthy that the optimized value of alpha is not exactly 0.1 as claimed in [1]. The optimized value for α should be in the range of 0.01 to 0.05. Although there is no formally correct procedure for choosing α, we can use the method of least squares to optimize its value.

However, in fact, the error estimation of *h* does not affect the overall performance of FEBA. It is very interesting that even when the error is around 60%, the fairness index is still nearly equal to 1. Or we can say that the performance of FEBA does not depend on how precisely we can estimate of its neighbours' latency. This can be explained by the fact that the higher is the estimated control latency, the larger the grant horizon becomes. Hence, if the nodes access control slots less frequently, they are allowed to assign grants over larger frame windows. By this way, FEBA still can guarantee the "fairness".

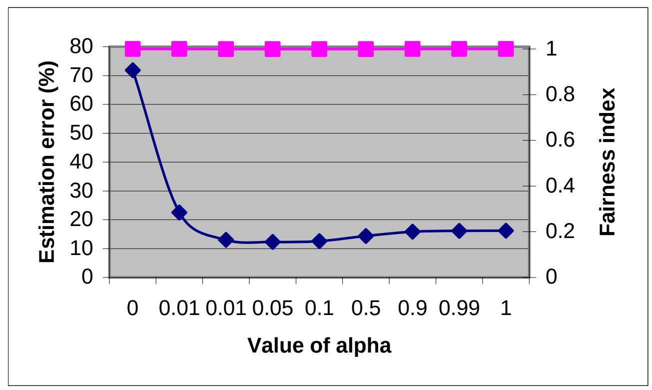

Fig. 5. Estimation error of control latency and their influence on fairness index with varying value of α

V. Improvement

A. Prioritizing traffic flows in FEBA

1) Motivation

As we discussed earlier the mechanism in requesting/ granting processes using a weight (*Φ*), in order to maintain a fair allocation to each flow, FEBA assigns $I_i(j)$ a value of 1 if j is under service at queue I, and 0 otherwise. That means the weight Φ_i of any queue i is proportional to the number of traffic flows under service according to expression (1). This will lead to a nearly perfect fairness index: every flow will have the same bandwidth regardless of the path length or characteristics of the flow. However, in a real network, each traffic flow has different level of priority. We cannot allocate the same amount of bandwidth to an Internet flow and a Video-streaming flow.

By assigning different values to indicator function $I_i(j)$, we can increase or decrease the weight of one queue. This allows some traffic flows to have higher priority than others.

2) Proposed solution

For each flow, we can add a 3-bit field in the header of its packet to contain the priority of the flow. So in total we can have $2^3 = 8$ levels of priority. At each node, we can maintain a table of priority level to help us calculate the weight for each queue. For example, we can use this table in differentiating flow priority:

TABLE I
PRIORITIY TABLE AND CORRESPONDING SERVICE

Index	Priority $I_i(j)$	Corresponding service
0	1	FTP file transfer
1	2	UDP tunnel
2	4	Internet web browsing
3	6	VoIP
4	8	Video conferencing
5	16	
6	32	
7	64	

For priority which is greater than 8, we can reserve for future use with higher-data-rate real time application.

When a node receives a packet from one flow, it will check the priority field and look up in its priority table to know which factor it has to use to calculate the weight.

3) Simulation result

In Fig. 6 we show the bandwidth allocation among the three flows with different levels of priority. Here we use a chain topology with 4 nodes. Node 0 and Node 1 send an asymptotic flow (constant bit-rate stream of 1000 bytes packets at a rate equal to the raw channel bandwidth) towards Node 3 with priority 1. Node 2 sends an asymptotic flow towards Node 3 with a varying priority from 1 to 8. As can be seen, when the three flows have the same priority, each flow have nearly the same amount of bandwidth (33%). When the priority of flow 2 to 3 increases, the amount of bandwidth it achieves also increases. For example, when its priority is eight times more than others, it gets around 80% of the total bandwidth while others get 10% each.

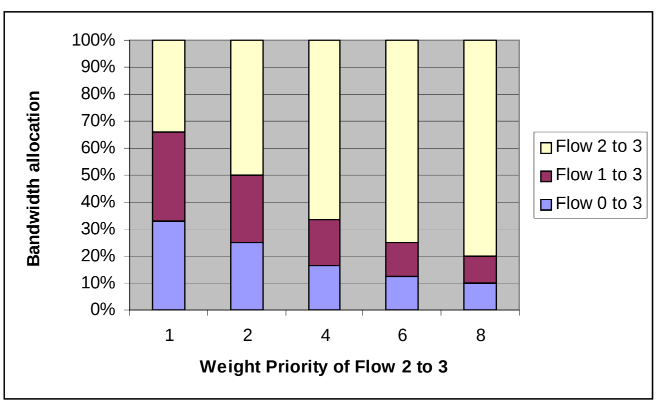

Fig. 6. Bandwidth allocation among three flows with different values of weight priority for flow 2 to 3

B. Modified FEBA for QoS differentiation in WiMAX

1) Observation

In the IEEE 802.16-2004 PMP mode (point-to-multipoint), we have four QoS classes of service: unsolicited grant service (UGS), real-time polling service (rtPS), non-realtime polling service (nrtPS) and best effort (BE). However, in Mesh Mode, we do not have a clear QoS differentiation scheme like that.

In addition, one of the reasons that affect the delay in FEBA is the system parameter *XmtHoldoffExponent* [1] which defines the MSH-DSCH transmission interval. The IEEE 802.16 standard [8] defines that after the current transmission time, a node is not allowed to transmit for the *XmtHoldoffTime*(H) which is given by the expression

$$H = 2^{XmtHoldoffExponent+4} \qquad (4)$$

Hence, to provide different QoS (in this case, we discuss only the average end-to-end delay) services to different applications, our idea is to differentiate the parameter *XmtHoldoffExponent* for each node.

2) Simulation

The result presented in this section is performed using a collocated scenario with varying number of nodes (a clique where all nodes are one-hop neighbors of each other) to demonstrate the effectiveness of our proposed idea. All N nodes are equally partitioned into three priority classes. Each node belongs to one priority class exclusively. We set the parameter *XmtHoldOffExponent* for 3 classes accordingly 1, 2 and 3. Figure 7 shows that class 1 has the smallest delay and the class 3 has the largest delay. This is because with greater exponent, the hold-off transmission time becomes longer. Thus, a node with lower priority can send requests at a lower frequency, leading to a larger delay in delivering all packets that are in its queue. We can also notice that if we increase the node density, the delay will increase due. This can be explained by the fact that nodes will lose more transmission opportunities as the number of competing neighbors increase.

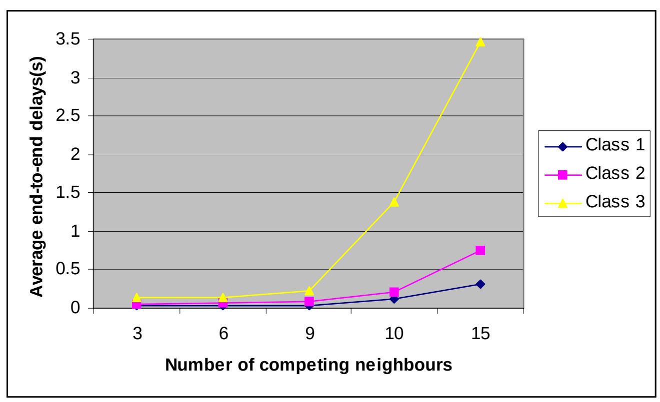

Fig. 7. Average delay in terms of the competing nodes for three classes of service

The obtained simulation result implies the effectiveness of our QoS differentiation scheme. One possible application is to use class 1 for real-time application with strict delay requirement such as VoIP or Video conference, class 2 for application with flexible delay (web browsing), and class 3 for best-effort application (email). Furthermore, as *XmtHoldoffExponent* has a size of three bits, we can provide up to eight different classes of services with the value of *XmtHoldoffExponent* between 0 and 7.

VI. Conclusion

This paper presents an analysis of Fair End-to-end Bandwidth Allocation (FEBA) algorithm, which is designed for IEEE 802.16 mesh network. We point out that FEBA not only provides bandwidth balancing among traffic flows but also makes the mesh network become load balanced. Some advantages and disadvantages in implementing FEBA are discussed in details as well.

We propose two ways to improve FEBA: prioritizing the traffic flows and differentiating the QoS for each node in the network. One QoS differentiation scheme is also suggested for IEEE 802.16 Mesh Network.

Acknowledgment

Author thanks Dr. Mehul Motani, National University of Singapore, with EE6902 – Computer Communication Networks who encourages them to read, think and write critically.

References

[1] Cicconetti, C.; Akyildiz, I.F.; Lenzini, L. Bandwidth Balancing in Multi-Channel IEEE 802.16 Wireless Mesh Networks. INFOCOM 2007. 26th IEEE International Conference on Computer Communications. Page(s):2108 – 2116.

[2] R. Jain, W. Hawe, D. Chiu. A Quantitative measure of fairness and discrimination for resource allocation in shared computer system. DEC-TR-301, September 26, 1984.

[3] B.S. Manoj; Ramesh R. Rao Wireless mesh networking - Architectures, protocols and standards. Load balancing in wireless mesh networks. Auerbach Publications

[4] Ian F. Akyildiz, Xudong Wang. A Survey on Wireless Mesh Networks. IEEE Radio Communications. September 2005

[5] Ian F. Akyildiz, Xudong Wang, Weilin Wang. Wireless mesh networks: a survey. Computer Networks 47 (2005) 445-487

[6] Parag S. Mogre, Matthias Hollick, Ralf Steinmetz. QOS in Wireless Mesh Networks: Challenges, Pitfalls, and Roadmap to its Realization. NOSSDAV , Urbana, Illinois USA.

[7] http://info.iet.unipi.it/

[8] *The Institute of Electrical and Electronics Engineer. IEEE Standard for Local and Metropolitan Are Networks Part 16: Air Interface for Fixed Broadband Access Systems, IEEE Std 802.16-2004. IEEE, 2004.*

[9] "An Integrated Framework for Enabling Effective Data Collection and Statistical Analysis with ns2" Claudio Cicconetti, Enzo Mingozzi, Giovanni Stea - Workshop on ns-2: the IP network simulator (WNS2) Pisa, Italy, October 10, 2006.