www.ietdl.org

Published in IET Communications
Received on 8th June 2010
Revised on 13th July 2011
doi: 10.1049/iet-com.2010.0450

ISSN 1751-8628

Peak avoidance and collision control for contention-based bandwidth requests in WiMAX systems

J.-Y. Pan[1,2] *S.-C. Shiu*[1] *W.-S. Lee*[1] *S.-W. Huang*[1]

[1]*Department of Communications Engineering, National Chung Cheng University, 168 University Road, Ming-Hsiung, Chia-Yi 621, Taiwan*
[2]*Center for Telecommunication Research, National Chung Cheng University, 168 University Road, Ming-Hsiung, Chia-Yi 621, Taiwan*
E-mail: jypan@comm.ccu.edu.tw

Abstract: There are two fundamental bandwidth-request mechanisms specified in the IEEE 802.16 standard: contention-based random access and contention-free polling. For example, non-real-time polling and best-effort services mainly rely on the contention-based mechanism to submit bandwidth requests to the base station (BS). However, the performance degrades considerably when the number of requests is high and they collide with each other. To avoid collision, the standard supports a truncated binary exponential backoff resolution. Nevertheless this resolution is inefficient in dispersing the requests when the network traffic is overloaded. In order to improve the efficiency of the random-access mechanism in the standard, the authors propose a new mechanism that aids mobile stations (MSs) in sending bandwidth requests more efficiently and also avoiding peak traffic. By using the mechanism, MSs send bandwidth requests under low loads, thus avoiding collisions.

1 Introduction

Fixed and mobile broadband wireless access networks are the most important key communication technologies in the near future. For this reason, the Institute of Electrical and Electronic Engineers (IEEE), world's leading professional association for the advancement of technology, has defined many network communication standards such as IEEE 802.11 for wireless networks and IEEE 802.3 for wired networks. The IEEE 802.16 standard is a novel wireless network solution that was developed in the last few years. It also has been commercialised under the name Worldwide Interoperability for Microwave Access (WiMAX). More and more applications in Internet are demanding a high quality of service (QoS). To support various multimedia applications, the IEEE 802.16 standard [1] defines four specific types of service flows that meet different user requirements. Each connection between a subscriber or a mobile station (SS or MS) and a base station (BS) is identified as one service flow by the connection ID. Unsolicited grant service (UGS) is defined to support constant bit rate flows such as voice over Internet protocol. Real-time polling service (rtPS) is designed for real-time applications that periodically send variable-size data packets, for example, video streams. Real-time applications require maximum delay and assurance of minimum bandwidth. To support non-real-time services, the IEEE 802.16 standard defines two other types of QoS: non-real-time polling service (nrtPS) and best-effort (BE) service. The nrtPS class is applied to services with no specific delay bound but requiring a minimum bandwidth, such as FTP. The latter QoS, that is, the BE service, is used to support traffic generated by most Internet applications (e.g. web surfing, email etc.).

The standard has also defined a contention-based bandwidth-request (BW-REQ) mechanism for nrtPS and BE services [1]. Without loss of generality, this study uses the BE service as an example for discussing contention-based BW-REQs. Before BE services require BS resources, the SS or MS has to send a BW-REQ and wait for the grant. In Section 2, we discuss the request and grant mechanisms in more detail. We review some related works about the performance of the request and grant mechanism as follows. Fallah *et al.* [2] analysed the BW-REQ mechanism by using a mathematical model. Vine *et al.* [3], Ni *et al.* [4], Staehle and Pries [5], Shehan [6] and He *et al.* [7] investigated the performance of the random-access scheme specified in IEEE 802.16. Vine *et al.* [3] analysed the mean delay for transmission requests from a fixed number of active stations under ideal channel conditions. They also investigated the performance of the random-access scheme specified in IEEE 802.16 by performing both simulation and mathematical analysis. Their study showed that the mean delay increases with the number of active stations. Ni *et al.* [4] first compared two fundamental BW-REQ mechanisms specified in the standard: random access and polling. Their results showed that random access outperforms polling when the request rate is low. However, its performance degrades considerably when the channel is congested. Staehle and Pries [5] studied the performance of the different random-access mechanisms specified in the IEEE 802.16 standards. Simulation results showed that the amount of resources that should be reserved for random access strongly depends on the desired performance. If long access delays can be tolerated, few resources are enough and a high throughput can be achieved. However, if short delays are required as some users produce delay critical traffic over BE connections, a considerable part of the uplink subframe should be devoted to random access. Shehan [6] presented an analytical model for the mandatory contention mechanism of IEEE 802.16d and analysed the performance of the BE service. He *et al.* [7] developed a unified analytical model to compare the BW-REQ schemes in fixed IEEE 802.16 networks. Sayenko *et al.* [8] tried to optimise the backoff window size and related parameters. Kim *et al.* [9] proposed an efficient BW-REQ scheme that enhances the performance by allocating more bandwidth to BE and nrtPS flows without affecting the UGS and rtPS flows. Ni *et al.* [10] proposed a new analytical model for the performance analysis of various contention-based BW-REQ mechanisms, including grouping and no-grouping schemes. Chou *et al.* [11] developed an analytical model for T16 timer setting and also derived analytical expressions for the average number of tries per BW-REQ and the average packet delay.

In these studies, traffic loading was shown to seriously affect the mean delay of the random-access BW-REQ mechanism. In the case of traffic overloads, the mean delay increases rapidly because of request collisions, which increases the number of retries. In this paper, we propose a 'BW-REQ congestion control' (BCC) scheme to mitigate the effect of bursts in requests and balance the loading of BW-REQs.

The remainder of this paper is organised as follows. Section 2 describes the request and grant mechanism and the frame structure of the IEEE 802.16 standard. Section 3 introduces our scheme. Section 4 presents the simulation environment. Simulation results are provided in Section 5. Finally, we present the conclusions in Section 6.

2 Overview of IEEE 802.16 standard

In this section, we introduce the frame structure and the request and grant mechanism of IEEE 802.16.

2.1 IEEE 802.16 frame structure

We consider the network to have a point-to-multipoint (PMP) architecture; this consists of one BS managing several MSs and SSs. Throughout this paper, we use the term 'MS' to refer to both 'MS and SS'. We now consider the time-division duplex mode in the PMP network. The frame structure is shown in Fig. 1. It consists of two subframes, where the subframe used for transmitting from the BS to the MS is called the downlink subframe and the one used for reverse transmission is called the uplink subframe. The downlink subframe carries the downlink map and uplink map; these are used to announce the downlink and uplink bandwidth allocations for data transmission. In the uplink subframe, contention time interval is allocated for MSs sending BW-REQs. The uplink channel descriptor (UCD) and downlink channel descriptor are also important information for bandwidth management; they consist of some essential parameters such as values of the minimum backoff window $W_{\min}$ and maximum backoff window $W_{\max}$ for contention-based BW-REQs and are expressed as power of 2. The Rx/Tx transmission gaps are located between the uplink subframe and the downlink subframe. The subframe duration can be dynamically sized to accommodate traffic in both directions. In addition, contention opportunities such as BW-REQ slots can also be flexibly arranged.

2.2 Request and grant mechanism

The MS supports two mandatory contention-based BW-REQ mechanisms: contention-based BW-REQs and contention-based code division multiple access (CDMA) BW-REQs [1, 12]. Our paper focuses on the contention-based BW-REQ mechanism. When an MS has to request for bandwidth on a connection of BE scheduling service, it sends a message to the BS containing the immediate requirements of the demand assigned multiple access connection. When the BS receives the BW-REQ message from the MS, the BS allocates resources for the MS in the next frame. In other words, the MS should send the BW-REQ first before a connection is made with the BE scheduling service using the bandwidth resource. In the specification, a request may come as a standalone BW-REQ header or as a piggyback request. The piggyback request is an optional feature and we have not discussed it as it is beyond the scope of our paper. Two methods – contention-based random access and contention-free polling – are defined in the standard for determining which MS should be allowed to transmit its request. No acknowledgement message is sent back to the MS to confirm whether the BW-REQ message successfully arrived or not at the BS in both methods. For this reason, checking the timeout of timer T16 can help in determining whether the BW-REQ message collided with another message and got corrupted, which then triggers the contention resolution mechanism. In this paper, we only focus on the contention-based random-access method.

With contention-based random access, the MS transmits a BW-REQ message during a specific contention time interval and the random backoff mechanism may avoid collisions among different MSs. The standard specifies the truncated binary exponential backoff as the mandatory random-access contention resolution. Before MSs send a BW-REQ message, they randomly choose an integer number from $[0, 2^{W_{\min}} - 1]$ as a backoff counter, where $W_{\min}$ is the initial backoff window size described in the UCD. As contention slots pass and the backoff counter counts down to zero, the MS transmits the BW-REQ message allocating enough

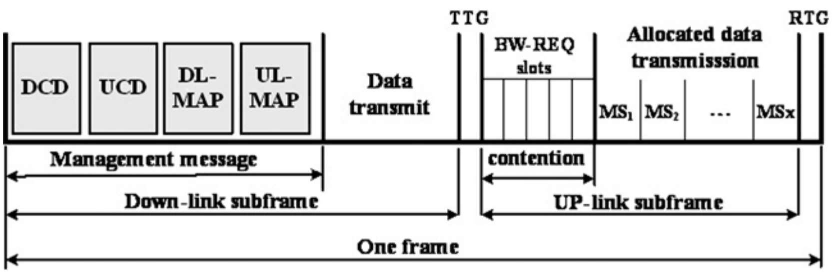

Fig. 1 *Frame structure of IEEE 802.16*

space for the current data in its buffers. The MS then starts the T16 timer and waits for the bandwidth grant from the BS. If the timer expires, the MS retries the process and selects a new backoff value in the interval of $[0, 2^{W_{\min}+1} - 1]$. In other words, the MS doubles the backoff window after each collision until the window size goes up to $2^{W_{\max}}$. The backoff method usually efficiently prevents collisions. In [3–6], the relationship between traffic loading and BW-REQ delay was investigated, and the results showed that when the traffic loading increases, the delay in the random-access method increases greatly.

3 Proposed algorithm

This section introduces the proposed random-access algorithm BCC in detail. The BCC sends a BW-REQ message efficiently in two steps: determine whether the traffic load is close to saturation and determine how the MS can avoid the peak traffic load. Without loss of generality, here, we use the BE service as an example for the random-access process of contention-based BW-REQs and do not involve other traffic types in the discussion of the request process.

3.1 Determine traffic loading peak

In BCC, we focus on the contention traffic loading of BW-REQ messages instead of data throughput, so we derive the traffic loading by collecting statistics of the number of received collisions during a time interval, where collisions occur when more than one BW-REQ arrive simultaneously at the BS. Here, ρ represents the loading of the request contention. The request contention loading index can be defined as

$$\rho = \frac{R_{\text{total}}}{S_{\text{bw}} \times T_{\text{sta}}} \tag{1}$$

where R_{total} means the number of detected collisions in the statistic time interval, S_{bw} is the number of contention slots per frame as defined by BS and for each T_{sta} frame we collect statistics of the BW-REQs.

Our paper discusses the forecasting of the peak traffic. We observed that the peak traffic contains considerable retransmissions of requests resulting from collisions of previous requests. Initially, an MS may send a request and wait for the corresponding grant from the BS. However, if the BS cannot successfully decode the request, mainly because of simultaneous requests colliding in one slot, then the MS waits for a response only until the T16 expires (which we set to 100 ms), which is a timer specified in the IEEE 802.16 standard. After that, the MS starts the retransmission scheme. If the BS senses many collisions in consecutive time slots, then the MSs involved in these collisions will continue to request and cause further collisions again in the foreseeable future (T16 timeout). In addition, a large number of collisions result from MSs that transmit just the first BW-REQ message for their bandwidth demand. Although the backoff window is doubled in the second transmission, according to the standard we still forecast that these BW-REQs will scatter in some interval. For example, the initial window range is 0–3, and then in the next retransmission, the window range will be 0–7. If one frame has four contention slots to accommodate these requests, we know the first requests of MSs demanding bandwidth simultaneously will fall in four slots of the same frame. If they fail to receive the grant from the BS, after T16 expires, their retransmission is scattered in consecutive two frames. Therefore we can forecast that many MSs will send BW-REQ messages in future frames; Fig. 2 shows the situation of further collisions in the future. Hence, when the value of ρ exceeds the threshold $T_{\max}$, after T16 expires, the retrying MSs are expected to send requests that collide again and cause a peak traffic load; we propose that the BS should alert all the MSs after T16 expires (100 ms) by a broadcast indication encoded in periodic UCD messages forecasting the impending peak. This indication flag can be encoded in a type–length–value format and simply be inserted in the bottom half of the UCD message with little overhead. This amendment is fully compatible with other non-peak-aware MSs. When MSs are aware of high loading, they execute the proposed algorithm for avoiding the peak.

3.2 Avoiding traffic loading peak

The core concept of this algorithm is to disperse the large number of BW-REQ in the peak load and rearrange them in low traffic loads. In our discussion, the BE service does not require any delay guarantee and real-time scheduling. For this reason, the algorithm may postpone sending BW-REQ messages to distribute the traffic load. The proposed algorithm deals with the above-mentioned problem in two steps: (i) the MS sends a new BW-REQ message and (ii) retransmits a BW-REQ owing to collision or channel error.

3.2.1 Sending new BW-REQ message: For sending a new BW-REQ message, the MS first determines whether the high loading flag of the broadcast message is true. If the result indicates that the traffic is at a loading peak, then the MS activates the power-saving class of Type III, which is one of three power-saving classes specified in the IEEE 802.16 standard [12], and enters the sleep mode. This type defines only a single sleep window, and the MS is reactivated after the sleep window expires. Without modifying the original exponential backoff mechanism in terms of slots, the proposed algorithm instructs the MS to simply skip the loading peak and sleep in terms of frames based on a BS's prediction with known collisions. We define $B_{\max}$ as the maximum number of frames for the sleep window; $B_{\min}$, the minimum. The MS selects a value in the interval $[B_{\min}, B_{\max}]$ as the sleeping window length. The MS then counts down the window by frame and retries the request method. We call this mechanism 'peak avoidance'. Fig. 2*b* shows the mechanism for peak avoidance. Using this mechanism, the BS notifies MSs that collisions may occur in the future. After informing about possible collisions, the BS asks the MSs to sleep for a certain period of time to prevent collisions between the new BW-REQ and resent BW-REQs.

3.2.2 Retransmitting BW-REQ message: This policy is named 'collision control'. For an MS retransmitting the BW-REQ message, we define P_{sleep} as the probability that the MS suspends retransmission. Before an MS retransmits a BW-REQ with a high loading flag, it selects a random real value between 0 and 1 to determine whether to postpone the retry process. If the MS does not pass the test (i.e. the random value is less than the threshold P_{sleep}), it enters the sleep mode and the sleep window size is set to 25 ms (five frames). We define S_{re} as the sleep window in this paper. After entering the sleep mode, the MS again attempts the retransmission process. Fig. 2*c* shows the

doi: 10.1049/iet-com.2010.0450

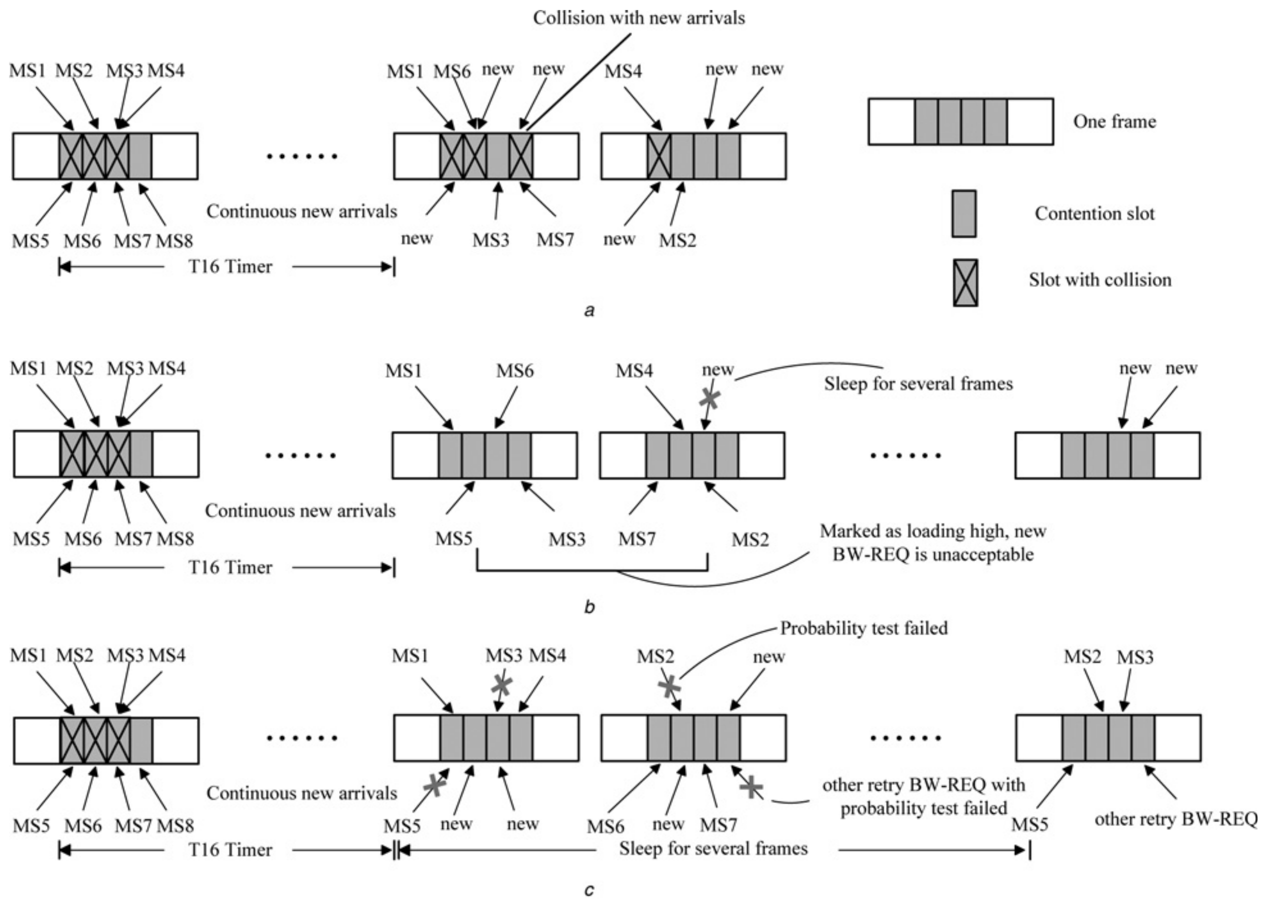

Fig. 2 *BW-REQ and collision control*

a Original
b Peak avoidance
c Collision control

mechanism for collision control. Using this mechanism, the BS allows the MS's retransmitted BW-REQ if it passes the probability test. The MS has to sleep for a while if it fails the probability test. This method can efficiently reduce the collisions of BW-REQs from different MSs.

Our algorithm not only decentralises new incoming BW-REQ traffic but also assures a better retransmission chance for contending MSs. The power-saving class of Type III may reduce power consumption, whose parameters such as start frame number and duration of sleep window can be determined using our algorithm. In this mechanism, the number of MSs in retransmission is reduced with a probability; in our simulation, we set the probability to 0.5. Fig. 3, pseudo code (Fig. 4) and flowchart (Fig. 5), shows our algorithm in detail.

Fig. 3 shows the framework of WiMAX with our mechanisms. The BS's collision detector detects collisions and counts the traffic loading index, informing the BS scheduler. Then the BS scheduler instructs the peak traffic indicator producing a peak traffic indication to MSs after T16 timeout. After the MS's peak traffic detector receives the indication, it triggers the collision control and the peak avoidance in the BW-REQ's generation, hence skipping possible collisions.

4 Simulation environment

In this section, we discuss the effect of our proposed scheme on the random-access mechanism performance using Network Simulator version 2 (NS2) [13] and use the IEEE 802.16 module of NIST [14] to simulate the random-access mechanism. In this paper, we do not consider collisions that occur because of channel error, that is, we assume the channel is perfect. In our simulation scenario, the data traffic for BE service is modelled as a web source. We use the traffic model named web exponential [15, 16] and set the parameters inter-arrival time λ as 5/s and average rate as 25 Kb/s. The IEEE 802.16 network parameters are shown in Table 1. In our simulation, nodes start to generate traffic one by one with 0.2 s intervals from the traffic start time (15 s) and stop sending traffic one by one at 0.2 s intervals after the traffic stop time (30 s).

5 Simulation results and performance analysis

5.1 Simulation 1: performance evaluation of different policies

We compared different mechanisms for our scheme: the original mechanism of the IEEE 802.16 standard; 'peak avoidance', using which MSs send new BW-REQ messages to lower the traffic load; 'collision control', which suspends ongoing request contentions to prevent collisions; and 'hybrid', where we use 'peak avoidance' and 'collision control' simultaneously. For these different situations, we compared the average delay and the number of collisions.

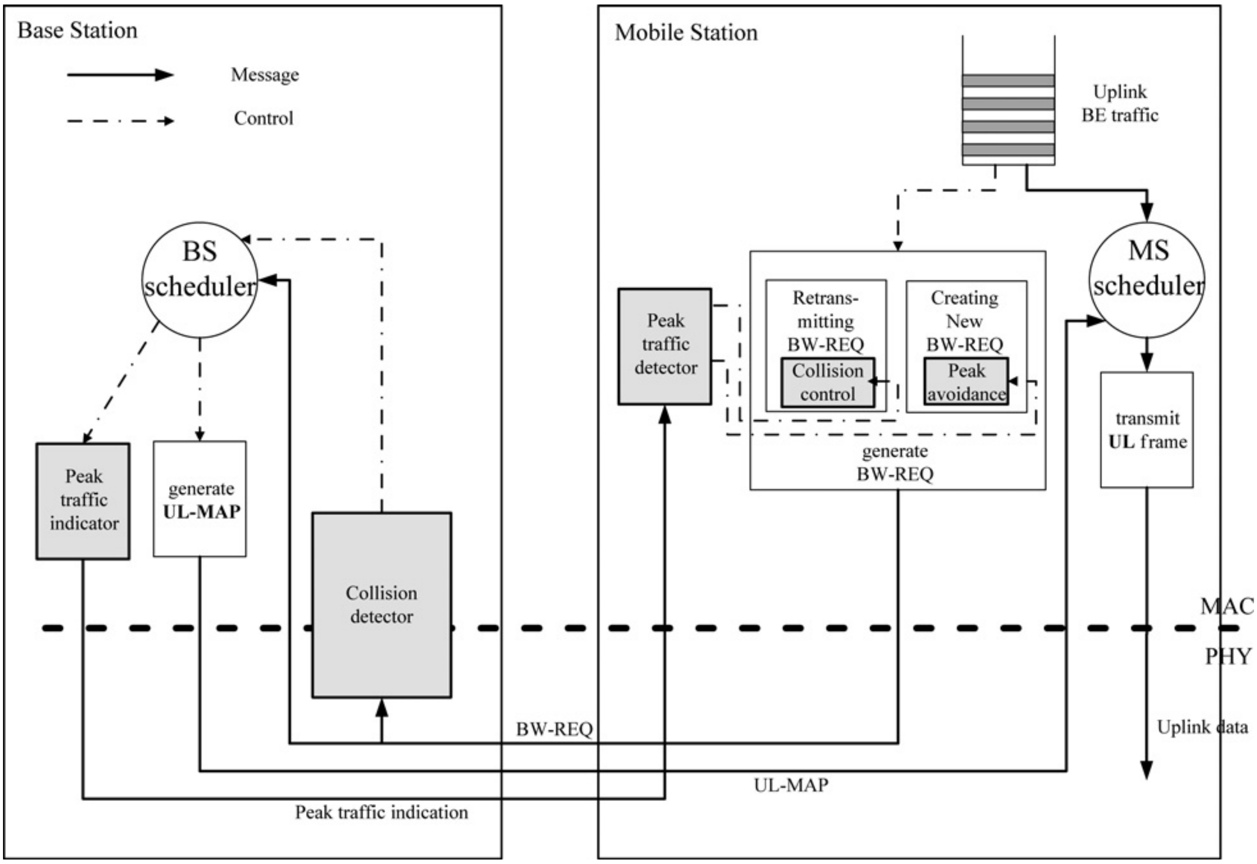

Fig. 3 *Framework of WiMAX architecture and proposed scheme*

```
Proposed Algorithm:
BW-REQ Congestion Control
{
    ρ = traffic loading index;
    T_max = threshold of loading index;
    P_sleep = threshold of sleeping probability;
    P_each-sleep = each node's sleeping probability;
    S_re = sleep window size for retry BW-REQ;
    B_random = randomly selected integer;
    B_min = minimum sleep window size;
    B_max = maximum sleep window size;
Start:
    if(BS's indication of {T16 timeout after the event (ρ ≧ Tmax)) ){

        if(Retransmission BW-REQ){ /* Collision control */
            /* execute probability test */
            P_each-sleep is a randomly selected value between 0 and 1

            if(P_each-sleep > P_sleep){
                continue BW-REQ transmission on primary path;
            }

            else{
                switch to Sleeping Mode for S_re frames ;
                go to Start;
            }
        {
        else{ /* For first transmission of BW-REQ */ /* Peak avoidance */
            B_random is a randomly selected value between B_min and B_max;
            switch to Sleeping Mode for B_random frames;
            go to Start;
        }
    }
    else{
        continue BW-REQ transmission on primary path;
    }
}
```

Fig. 4 *Pseudo code of proposed mechanism*

Table 2 shows the statistics of the simulation, including the number of BW-REQ, number of collisions and average delay, where the BW-REQ process is defined as a complete request process in that MSs notify the BS the need for uplink bandwidth allocation until they succeed in getting the allocation or cancel the request.

The average delay is the delay between an MS sending a BW-REQ message and the time when the BS receives this message and successfully decodes it. If request collisions occur, the MS waits until T16 expires (100 ms). The delay time (T16) accumulates up to the original BW-REQ delay. In other words, the delay means the time for a successful transmission (and possibly a few preceding failed transmissions included). The percentages shown in Table 2 are the ratios when compared with the original mechanism.

Fig. 6 shows the number of collisions in four situations. At the beginning, the network entry of MSs results in few collisions. From the traffic start time, MSs start to send BW-REQ to the BS, which increases the traffic load and collisions. Then, nodes stop generating traffic at about frame 7000, which decreases the collisions. After a short time, the retransmission of BW-REQ comes back, which continues to cause serious collisions, especially for the original mechanism. Using 'collision control', the involved MSs stop retrying according to a probability, which was set to 0.5 in this simulation, and the number of collisions is noticeably reduced. The probability is a parameter that we tried to optimise in the following simulation. 'Peak avoidance' and 'hybrid' also reduced collisions to less than that of the original mechanism. 'Peak avoidance' disperses the large number of BW-REQ in the peak load and rearranges them in the low traffic load, so the number of collisions is much less than the original mechanism, as

IET Commun., 2012, Vol. 6, Iss. 3, pp. 324–334
doi: 10.1049/iet-com.2010.0450

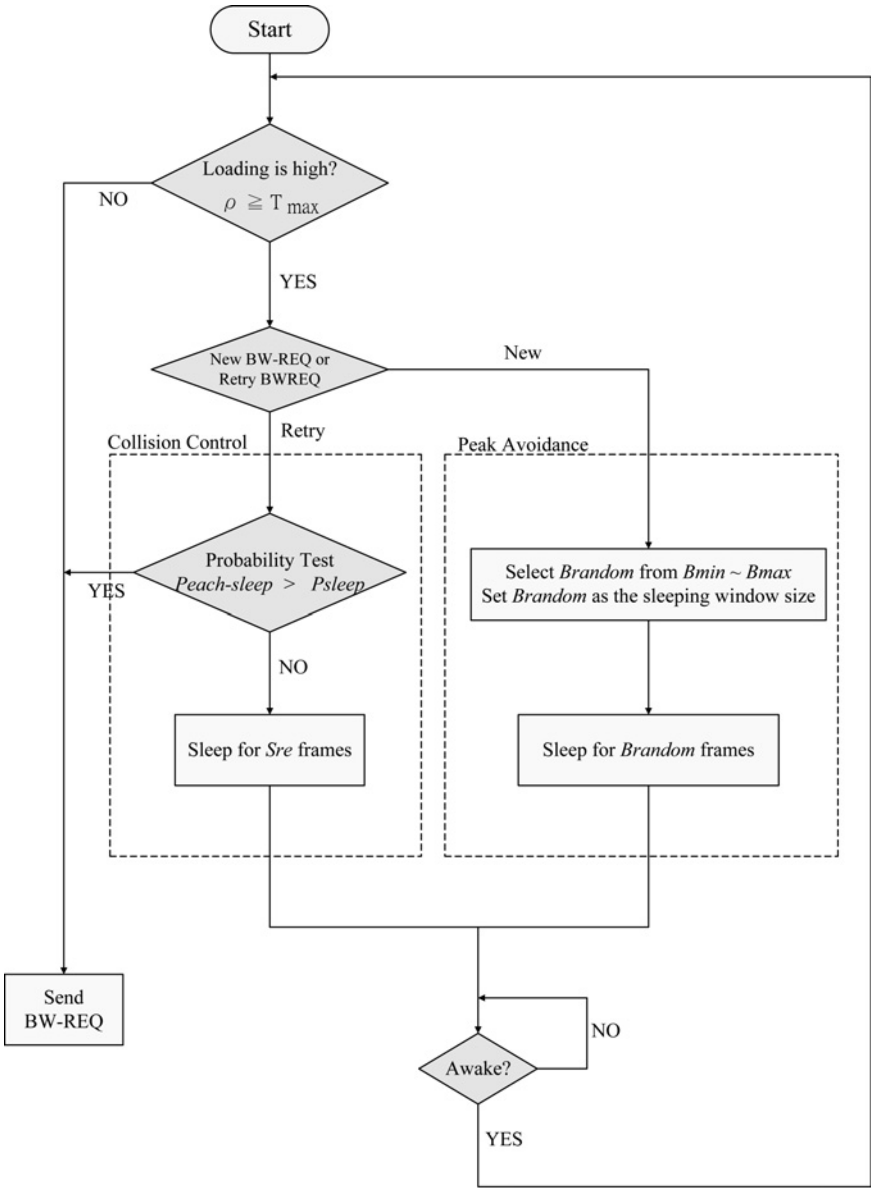

Fig. 5 *Flowchart of proposed mechanism*

Table 1 Simulation parameters

Simulator parameter	Value
channel bandwidth	11 MHz
frame duration	5 ms
request backoff start (W_{min})	2
request backoff stop (W_{max})	6
contention BW-REQ collision–detection timeout (T16)	100 ms
number of nodes	26
T_{max}	0.6
B_{min}, B_{max}	10, 20
P_{sleep}	0.5
simulation time	40 s
traffic start time	15 s
traffic stop time	30 s
S_{bw}	5
S_{re}	25 ms (5 frames)
T_{sta}	1

Table 2 Statistics on BW-REQ

	Average delay (in frames)	Number of completed BW-REQ	Number of received collisions
original	8.98 (100%)	2214	538 (100%)
peak avoidance	6.70 (75%)	1737	456 (85%)
collision control	5.25 (59%)	1724	399 (75%)
hybrid	6.64 (74%)	1731	443 (83%)

shown in Fig. 6. A comparison of 'peak avoidance' and 'hybrid' shows that the latter had a higher number of collisions, but this result is an outlier. We assumed that the non-optimised parameters B_{min}, B_{max} and P_{sleep} led to this unreasonable result.

Fig. 7 shows the average delay for the four cases, and the results are similar to those shown in Fig. 6. 'Collision control' reduces the delay efficiently. Surprisingly, in the

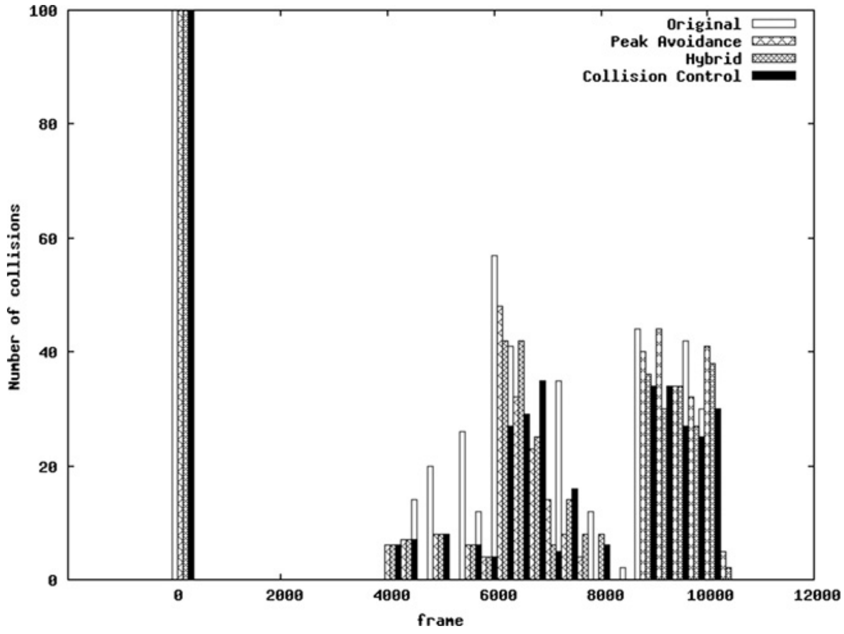

Fig. 6 *Number of collisions*

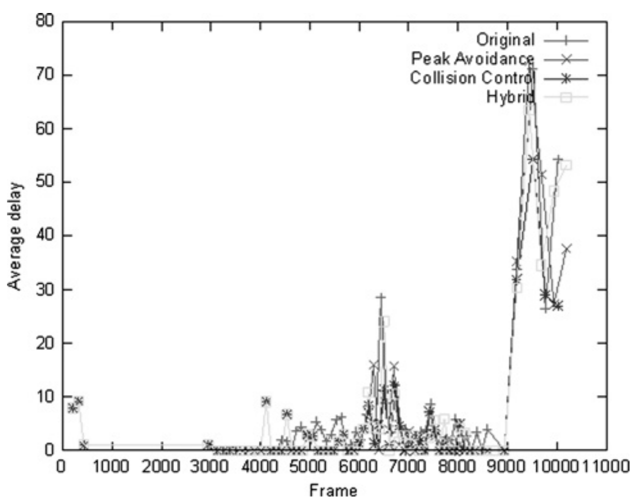

Fig. 7 *Average delay*

9300th frame, 'peak avoidance' produced a higher delay than the others. We conjecture that this is because the burst of new BW-REQs caused frequent collisions, and 'peak avoidance' cannot be expected to mitigate this.

The parameters we used in this simulation are listed in Table 1. Fig. 8 shows the average delay for the different mechanisms upon variation in the number of MSs. The overall traffic did not reach the peak load when there were less than 11 nodes, so it did not trigger our mechanism, and the average delay for the different mechanisms was almost the same. When the number of nodes was above 11, the average delay for our scheme was considerably reduced from the original mechanism.

To summarise Simulation 1, our scheme achieved better improvement in reducing both collisions and request delays. In our prediction, 'hybrid' should have a better result than only using 'peak avoidance' or 'collision control'. However, Fig. 8 shows 'hybrid' as more unstable, and Fig. 7 shows that 'peak avoidance' cannot reduce the average delay at peak load. We think that these anomalies result from the parameters having not yet been optimised. When we activate both 'peak avoidance' and 'collision control' without adjusting the related parameters, it may lead to collisions again after sleeping. Thus, we tried to optimise the related parameters through a heuristic approach in Simulation 2.

5.2 Simulation 2: heuristically optimising parameters of proposed schemes

In Simulation 2, we focused on adjusting the parameters of 'peak avoidance' and 'collision control', after which we analysed the performance.

The conceptual procedure of determining parameters is listed as follows.

Step 0: Initialing proper parameters as in Table 1.
Step 1: Deciding the optimal sleep window size according to the performance of 'peak avoidance'.
Step 2: Deciding the optimal value of Psleep according to the performance of 'collision control'.
Step 3: With adjusted sleep window size and Psleep, deciding the value of $T_{\max}$ according to the performance of 'hybrid'.

Firstly, we initialise the parameters in the scheme and observe the performance. The proposed schemes work in peak traffic. Then, we can determine the optimal parameters for two schemes 'peak avoidance' and 'collision control' while separately observing their average delay in heavy load (i.e. larger number of nodes). Finally, we adjust $T_{\max}$ to the optimal value which can minimise average delay of 'hybrid' with the two tuned parameters as mentioned above. In the hybrid mechanism, the two parameters 'sleep window size' and 'Psleep' would not affect each other scheme because they belong to different schemes and each scheme handles different types of BW-REQ, that is, new and retry BW-REQ. The discussion for tuning parameters in this simulation is as follows.

In 'peak avoidance', the most important parameter is the sleep window size. After the first retransmission, the retransmitted BW-REQ message is located at the contention slot between two frames. Hence, we consider that the sleep

doi: 10.1049/iet-com.2010.0450

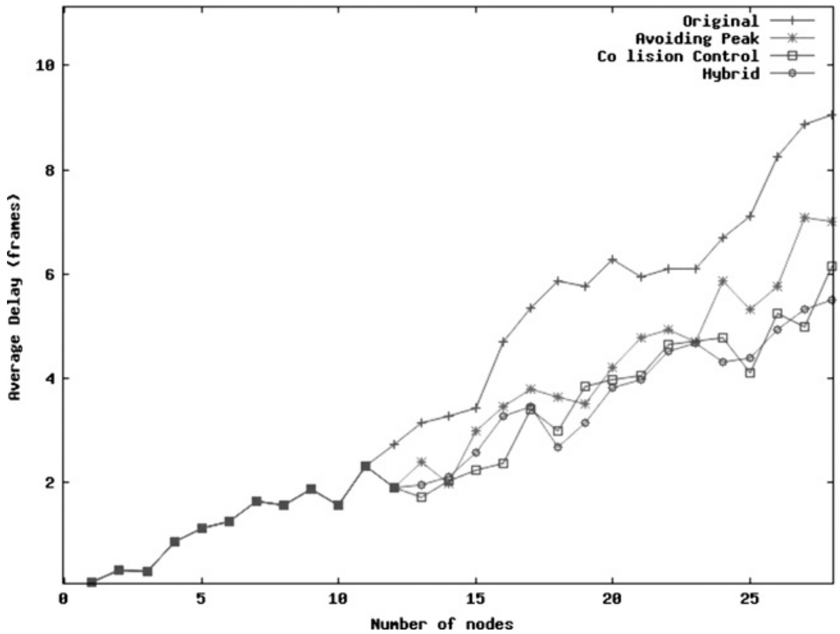

Fig. 8 *Average delay upon variation in number of MSs*

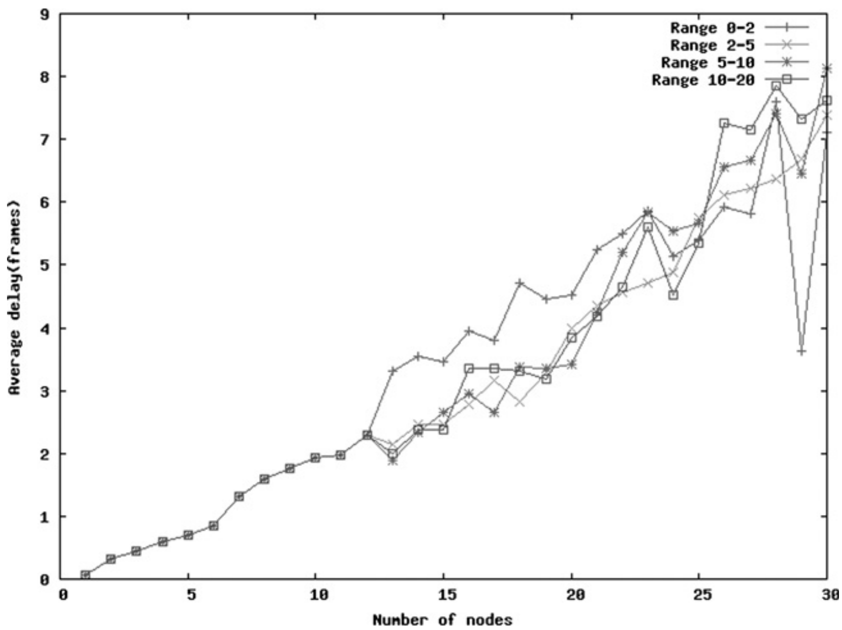

Fig. 9 *Comparison of average delay for different window sizes*

window size must be larger than two frames, and we set the sleep window size as 2–5, 5–10 and 10–20 frames to observe the simulation results.

Fig. 9 represents the relationship between the window size and delay. In Simulation 1, the sleep window size was 10–20 frames, and the results for Simulation 1 were quite unstable. If we used 5–10 frames as the sleep window size, the result was better than for 10–20 frames. If we set windows size parameters smaller, the average delay became lower. However when we used 0–2 frames as the sleep window size parameters, the average delay became unstable again. Finally, when we set the sleep window size as 2–5 frames, the result became more stable, and we obtained a smooth curve. Thus, in our opinion, the best choice for the sleep window size is 2–5 frames.

In ‘collision control’, we define P_{sleep} as the retransmission probability. A large P_{sleep} allows more MSs to stop retransmission and switch to the power-saving mode. We varied P_{sleep} as 0.25, 0.5 and 0.75 to determine a better choice for P_{sleep}.

Fig. 10 shows the average delay for different P_{sleep}. P_{sleep} represents the probability of sleeping. For $P_{\text{sleep}} = 0.25$, there are 75 MSs that execute the retransmission process per 100 retransmission MSs. As shown in Fig. 10, a small P_{sleep} results in better performance when the traffic is heavy but may cause more delay because of many nodes performing the retransmission process. We also observed that a large P_{sleep} results in too many nodes entering the sleep mode and increases the average delay as well.

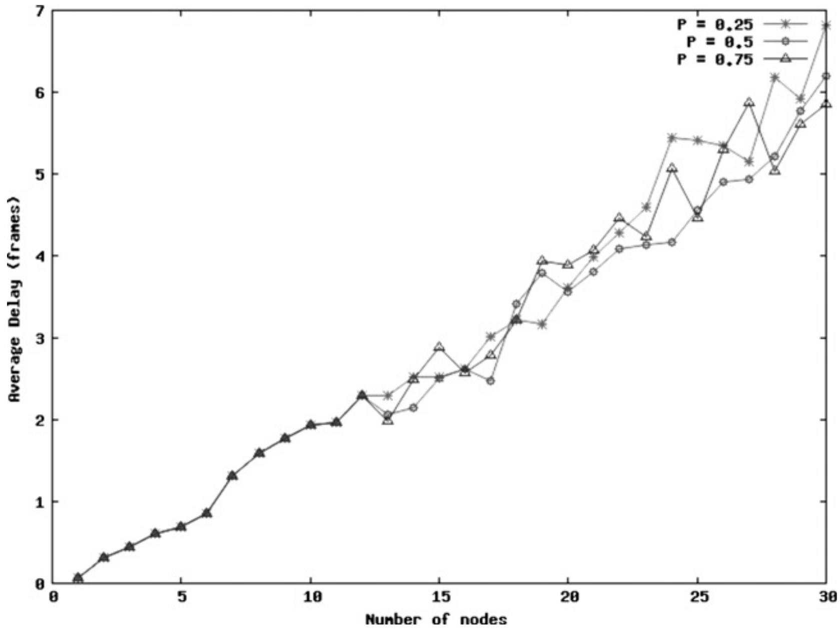

Fig. 10 *Comparison of average delay for different* P_{sleep}

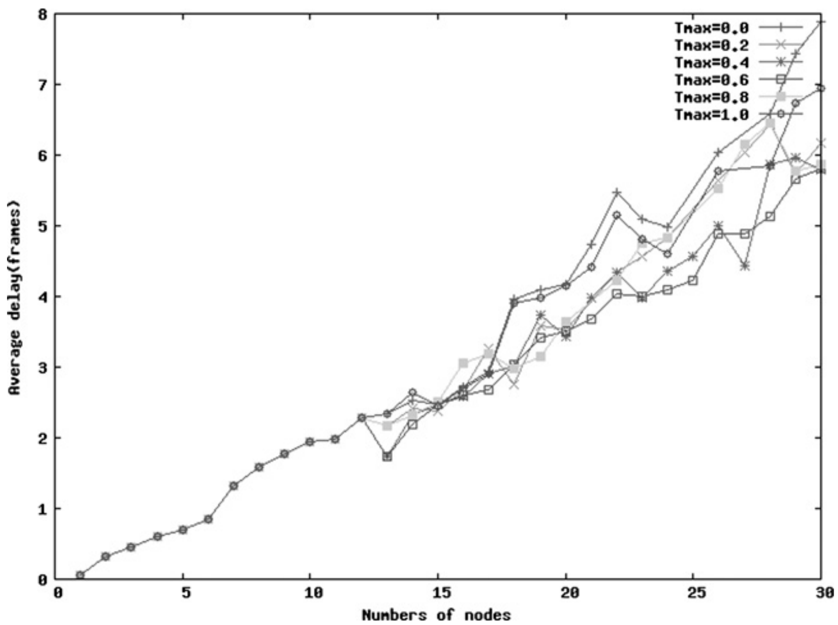

Fig. 11 *Comparison of average delay for different* T_{max}

On the basis of this result, we think that $P_{\text{sleep}} = 0.5$ is the best choice among the three, and we used this value in the following simulation.

Fig. 11 shows the average delay for different $T_{\max}$. $T_{\max}$ represents the threshold of loading index. The result of simulations shows that if we set $T_{\max}$ close to 0 or 1, average delay will increase in both cases. If we set $T_{\max}$ between 0.4 and 0.6, the proposed mechanism performs better in the average delay than other cases. The case of $T_{\max} = 0.6$ is the best in most cases, so we set $T_{\max}$ as 0.6 in this study.

5.3 Simulation 3: performance of proposed schemes after tuning

After adjusting the parameters in Simulation 2, we used the simulation parameters in Table 1 except that the sleep window size was set to 2–5 frames and then observed the results.

Fig. 12 shows the results for the four mechanisms with optimised parameters. Initially, the average delay for 'peak avoidance' and 'collision control' were almost the same, but under heavy traffic conditions, 'collision control' performed better than 'peak avoidance'. However, 'collision control' performed the best and had an average delay that was lower than the other three mechanisms. Compared with the original mechanism, Fig. 12 clearly shows that 'collision control' greatly reduced delay. Besides, Fig. 12 also shows slight instability of proposed schemes in heavy load.

In our proposed mechanisms, some requests waited for a certain period of time in order to ease the overall traffic when the traffic becomes heavy. However, the cost was not as high as expected.

IET Commun., 2012, Vol. 6, Iss. 3, pp. 324–334
doi: 10.1049/iet-com.2010.0450

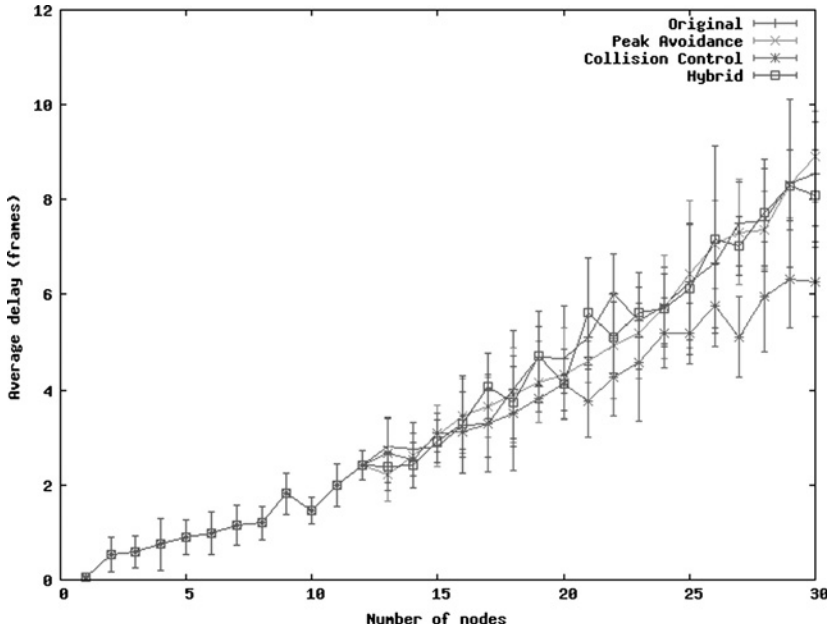

Fig. 12 *Average delay with confidence interval (95%) for four cases with optimised parameters*

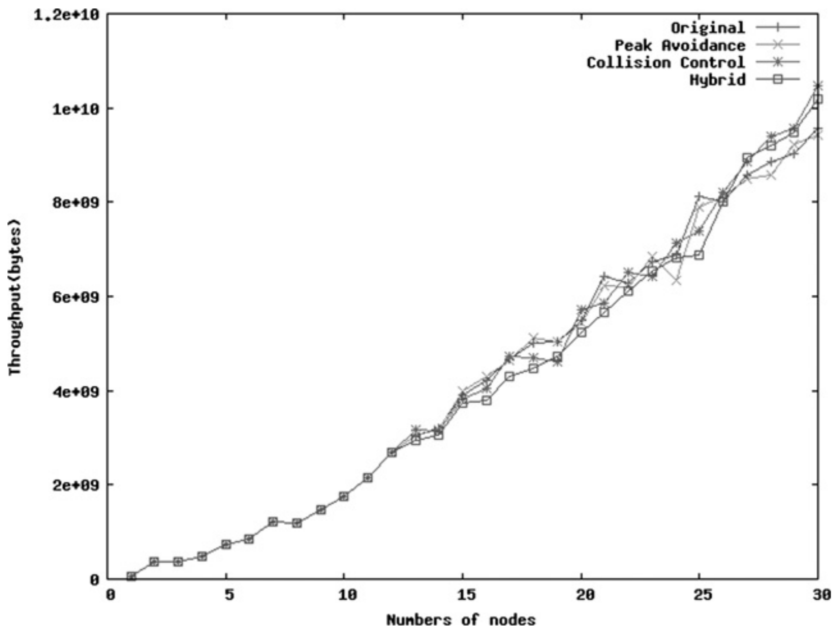

Fig. 13 *Comparison of throughput for four cases*

Fig. 13 shows how the throughput is affected by four mechanisms. The throughput in our article is defined as the overall traffic in the simulation time. When the traffic load is light, that is, the number of MSs is small, the original mechanism performs slightly better in terms of throughput. When the traffic load gets higher, the overall throughput of our mechanisms increases to more than that of the original mechanism. Surprisingly, the mechanism reduces the collision frequency to enhance the overall efficiency, thus increasing the throughput.

6 Conclusions

In this paper, we discuss our attempt to improve the efficiency of the random-access mechanism in the IEEE 802.16 standard. The proposed algorithm balances the traffic load and decreases the number of feckless BW-REQ messages. In our algorithm, the MS sends BW-REQ messages in the BE service more efficiently by controlling congestion.

Our simulation first focused on the performance of the proposed mechanism and then tuned the parameters. Before adjusting the parameters, the simulation result showed that our proposed mechanism performed better than the original mechanism but was not very stable. In addition, the 'hybrid' mechanism did not perform as well as we expected; the 'hybrid' mechanism was supposed to show the best performance. Therefore we tuned the related parameters by using a heuristic approach and analysed the performance. The proposed mechanisms not only balanced the traffic load but also effectively reduced the collision frequency.

This paper discusses three types of mechanisms: peak avoidance, collision control and hybrid. The simulation results also verified the effectiveness of our mechanisms. Although 'hybrid' was a combination of 'peak avoidance' and 'collision control', it did not perform better than 'collision control' only. Surprisingly, 'collision control' was the most effective among the three mechanisms for reducing the average delay. This mechanism does not guarantee fairness and stability if the system stays under high load conditions for a long time, so this issue will be addressed in our future work.

7 Acknowledgments

The authors thank the National Science Council, Taiwan, for partially supporting this research under contract nos. NSC 99-2221-E-194-053 and NSC 99-2221-E-324-012, as well as the Hsin-Chu Science Park, Taiwan, under the contract no. 99IA03. The authors would also like to thank the anonymous reviewers of the paper for their very helpful and constructive criticism. Their suggestions have enabled us to improve the quality of this work. Editage is appreciated for their editorial assistance.

This study was partially published in the proceedings of APCC-2008, the 14th Asia–Pacific Conference on Communications.

8 References

1 IEEE Std. 802.16-2004.: 'Local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems', October 2004
2 Fallah, Y.P., Agharebparast, F., Alnuweiri, M.M.H., Leung, V.C.M.: 'Analytical modeling of contention-based bandwidth request mechanism in IEEE 802.16 wireless networks', *IEEE Trans. Veh. Technol.*, 2009, **57**, (5), pp. 3094–3107
3 Vine, A., Zhang, Y., Lott, M., Tiurlikov, A.: 'Performance analysis of the random access in IEEE 802.16'. IEEE 16th Int. Symp. on Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005, 11–14 September 2005, vol. 3, pp. 1596–1600
4 Ni, Q., Vinel, A., Xiao, Y., Turlikov, A., Jiang, T.: 'Wireless broadband access: WiMAX and beyond – investigation of bandwidth request mechanisms under point-to-multipoint mode of WiMAX networks', *IEEE Commun. Mag.*, 2007, **45**, (5), pp. 132–138
5 Staehle, D., Pries, R.: 'Comparative study of the IEEE 802.16 random access mechanisms'. Int. Conf. on Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST'07, 12–14 September 2007, pp. 334–339
6 Shehan, P.: 'Analysis of contention based access for best effort traffic on fixed WiMAX'. Fourth Int. Conf. on Broadband Communications, Networks and Systems, 2007, 10–14 September 2007, pp. 552–558
7 He, J., Tang, Z., Chen, H.H.: 'Performance comparison of OFDM bandwidth request schemes in fixed IEEE 802.16 networks', *IEEE Commun. Lett.*, 2008, **12**, (4), pp. 283–285
8 Sayenko, A., Alanen, O., Hamalainen, T.: 'On contention resolution parameters for the IEEE 802.16 base station'. IEEE Global Telecommunications Conf., 2007. GLOBECOM'07, 26–30 November 2007, pp. 4957–4962
9 Kim, S.J., Kim, W.J., Suh, Y.J.: 'An efficient bandwidth request mechanism for non-real-time services in IEEE 802.16 systems'. COMSWARE 2007. Second Int. Conf. on Communication Systems Software and Middleware, 7–12 January 2007, pp. 1–9
10 Ni, Q., Hu, L., Vinel, A., Xiao, Y., Hadjinicolaou, M.: 'Performance analysis of contention based bandwidth request mechanisms in WiMAX', *IEEE Netw. Syst. J.*, 2010, **4**, (4), pp. 477–486
11 Chou, S.-F., Liu, J.-H., Chao, H.-L., Guo, T.-C., Liu, C.-L., Tsai, F.-J.: 'Performance enhancement of contention-based bandwidth request mechanism in IEEE 802.16 WiMAX networks'. IEEE 21st Int. Symp. on Personal Indoor and Mobile Radio Communications, September 2010, pp. 1287–1292
12 IEEE Std. 802.16e-2005.: 'Local and metropolitan area networks part 16: air interface for fixed and mobile broadband wireless access systems amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1', February 2006
13 'The Network Simulator – ns-2' [Online]. Information Sciences Institute (ISI)/University of Southern California (USC) Available: http://www.isi.edu/nsnam/ns/index.html
14 'Seamless and secure mobility' [Online]. National Institute of Standards and Technology (NIST) Available: http://www.antd.nist.gov/seamlessandsecure/toolsuite.html
15 Cicconetti, C., Erta, A., Lenzini, L., Mingozzi, E.: 'Performance evaluation of the IEEE 802.16 MAC for QoS support', *IEEE Trans. Mob. Comput.*, 2007, **6**, (1), pp. 26–38
16 Molina, M., Catelli, P., Foddis, G.: 'Web traffic modeling exploiting TCP connections' temporal clustering through HTML-REDUCE', *IEEE Netw.*, 2000, **14**, (3), pp. 46–55

doi: 10.1049/iet-com.2010.0450