On Contention Resolution Parameters for the IEEE 802.16 Base Station

Alexander Sayenko
Nokia Research Center
Helsinki, Finland
alexander.sayenko@nokia.com

Olli Alanen and Timo Hämäläinen
Telecommunication laboratory, MIT department,
University of Jyväskylä, Finland
{olli.alanen, timo.hamalainen}@jyu.fi

***Abstract*—In the IEEE 802.16 networks, the base station allocates resources to subscriber stations based on their QoS requirements and bandwidth request sizes. A subscriber station can send a bandwidth request when it has an uplink grant allocated by the base station or by taking part in the contention resolution mechanism. This paper presents analytical calculations for parameters that control the contention resolution process in the IEEE 802.16 networks. In particular, the backoff start/end values and the number of the request transmission opportunities are considered. Simulation results confirm the correctness of theoretical calculations. They also reveal that the adaptive parameter tuning results in a better throughput when compared to a static configuration. At the same time, all the timing requirements are met.**

***Index Terms*—IEEE 802.16 WiMAX, contention resolution, QoS, NS-2 simulator**

I. INTRODUCTION

IEEE 802.16 is a standard for the wireless broadband access network [1]. It provides a flexible fixed-wireless and mobile-wireless access between subscriber stations (SS) and a provider. The main advantages of 802.16 when compared to other wireless network access technologies are the longer range and a more sophisticated support for the Quality-of-Service (QoS) at the MAC level. Different application and service types can be used in the 802.16 networks and the MAC layer is designed to support this convergence.

The basic approach for allocating resources in the 802.16 network is that the base station (BS) does the scheduling for both the uplink and downlink directions. In other words, the BS translates the QoS requirements and information on the bandwidth request sizes of subscriber stations (SSs) into an appropriate number of slots within the MAC frame [2]. While the BS allocates some minimum number of slots for scheduling classes, such as Unsolicited Grant Service (UGS) and real-time Polling Service (rtPS), it may decide not to allocate slots at all for the non-real-time Polling Service (nrtPS) and Best Effort (BE) connections if their virtual uplink queue lengths equal zero. Later, when such an uplink connection needs slots to transmit data, it has to go through the contention resolution procedure to send the bandwidth request.[1]

[1]It is worth mentioning that it concerns only the ertPS, nrtPS and BE connections because the UGS and rtPS request/transmission policy does not allow to participate in the contention period.

The contention resolution mechanism in the 802.16 networks differs from the one used in the 802.11 standard [3]. While the 802.11 SS tries to *avoid* collisions by generating the random backoff timer and listening whether the medium is busy or not, the 802.16 SS just selects randomly the transmission opportunity. Though the transmission opportunity is chosen based on the similar truncated binary exponential backoff mechanism, an SS does not detect whether another SS is transmitting at the same time or not. If the BS receives successfully the bandwidth request, then it allocates resources for the SS. Otherwise, if a collision occurs or a packet is lost, no actions are taken and the SS retransmits the bandwidth request.

The 802.16 contention resolution mechanism is controlled by two sets of parameters: the number of the contention slots and the backoff start/end window values. While the number of the contention slots determines an overall number of the transmission opportunities during which an SS can send the bandwidth request, the backoff start/end window values control when an SS sends the bandwidth request. These parameters are set at the BS and transmitted to SSs in the uplink map (UL-MAP) and the uplink channel descriptor (UCD) messages. Thus, the BS can control flexibly SSs and the contention mechanism. The 802.16 specification left open concrete values for the backoff start/end window values, neither does it specify how a number of the transmission opportunities should be calculated. Thus, the task of a provider is to use some algorithm to set these parameters. On the one hand, knowing an average number of inactive connections, a provider can always choose an appropriate static configuration. On the other hand, a better resource allocation is achieved if these parameters are tuned adaptively. Besides, it eliminates a need to find an optimal static configuration, especially when the number of competing connections changes drastically.

In this paper, we present analytical calculations to determine optimal values for the backoff start/end values and an optimal number of the request transmission opportunities. Our paper extends the previous research works that tackled the problem of the contention resolution mechanism in the 802.16 networks. In [4], the deep analysis of the performance of the exponential backoff algorithm has been presented. However, authors have considered 802.11 and 802.3 networks, in which an SS has to go through the contention resolution for *every*

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

packet it wants to send. As mentioned earlier, the 802.16 SS takes part in contention to send only the *first* packet from the data burst. In [5], authors reformulated some basic expressions taken from [4]. However, only the impact of the *backoff start* value was considered. In [6], [7], relatively simple algorithms on how to calculate the number of the *contention slots* were presented. In particular, [6] bases itself on a rule "the number of the contention slots should be equal to three times the number of data packets that can be transmitted". Absence of the simulation results gives no chance to test its efficiency. In [7], authors draw a simple conclusion that the number of the transmission opportunities should equal the number of SS. In our paper, we present calculations for both the backoff start/end values and the number of the transmission opportunities. It is also worth mentioning that when compared to [5], [6], [7], we run simulation scenarios in the NS-2 simulator that provides more realistic traffic models and protocol behaviours. We have implemented the 802.16 MAC level and the basic features of the OFDM PHY level.

The rest of this paper is organized as follows. Section II presents the details of the 802.16 contention resolution procedure and the analytical calculations for the backoff start/end values and the number of the transmission opportunities. Section III presents the simulation results. Finally, Section IV concludes the paper.

II. Analysis of the contention resolution procedure

A. 802.16 contention resolution

When an SS wants to enter the contention resolution process, it sets its internal backoff window equal to the backoff start value defined in the UCD message. Then, the SS selects randomly a number within its backoff window. This random value indicates a number of contention transmission opportunities that the SS will wait before transmitting the bandwidth request. The SS considers the contention transmission lost if no data grant is given within the required time interval after the bandwidth request has been sent. If so, the SS increases its backoff window by a factor of two and selects a number within its new backoff window.

As mentioned earlier, the contention resolution parameters are set by the BS. Thus, the BS can track the number of SSs, calculate, and announce a new configuration for the contention mechanism. Tracking SSs is not a complicated task because the BS has to keep a list of connections with the related parameters. According to [1], [8], only the ertPS, nrtPS, and BE connections can take part in the contention. Since an SS enters the contention resolution process only when it wants to send data and does not any uplink grant, the BS can calculate easily the number of SSs that may potentially send the bandwidth request. For these purposes, the BS just has to iterate through all the uplink ertPS, nrtPS, and BE connections and account for those ones for which the scheduler at the BS has not assigned uplink resources. The iteration will not take additional time since it can be combined with the resource allocation stage [2]. After that, the BS can calculate and announce new backoff start/end values and, possibly, recalculate the number of the contention slots.

B. Stations with identical backoff window values

The BS announces the backoff start/end values as the power-of-two numbers, where only the power value is transmitted. Suppose, S is the backoff start power value. Then, the range for the randomly generated backoff window is $[0..W-1]$, where W is determined as follows:

$$W = 2^S. \tag{1}$$

If W is the current window value, then the probability that a particular value is chosen is $1/W$. An SS can transmit successfully the bandwidth request if it generates some backoff value that is not generated by any other SS:

$$\frac{1}{W}\prod_{j=1}^{N-1}\left(1-\frac{1}{W}\right), \tag{2}$$

where N stands for the number of connections that take part in the contention resolution procedure. Since an SS can generate any value within the backoff window, then the probability for a successful transmission is

$$p = \sum_{i=1}^{W}\left(\frac{1}{W}\prod_{j=1}^{N-1}\left(1-\frac{1}{W}\right)\right) = \left(\frac{W-1}{W}\right)^{N-1}. \tag{3}$$

We can use (3) to find the backoff start value:

$$W = \frac{1}{1-\sqrt[N-1]{p}}. \tag{4}$$

The BS represents the backoff start value as a power-of-two number, where only the power value is transmitted. Since it is an integer number, we can calculate it as follows:

$$S = \min\left\{\lceil \log_2 W \rceil, 15\right\}. \tag{5}$$

The backoff start value occupies one byte in the UCD message, where only the four lower bits are used. Thus, the maximum power value is 15.

Expression (4) is complicated in implementation due to the root expression. Since most hardware platforms with the floating-point accelerators support functions, such as $log_2(x)$ and 2^x, it is more efficient to calculate W as follows:

$$W = \frac{1}{1-2^{\frac{\log_2 p}{N-1}}}. \tag{6}$$

By combining (6) and (5), we obtain

$$S = \min\left\{\left\lceil -\log_2\left(1-2^{\frac{\log_2 p}{N-1}}\right)\right\rceil, 15\right\}. \tag{7}$$

As expression (7) provides valid results only when $N > 1$, the following expression calculates S for any number of competing connections N:

$$S = \begin{cases} \min\left\{\left\lceil -\log_2\left(1-2^{\frac{\log_2 p}{N-1}}\right)\right\rceil, 15\right\} N > 1, & \text{(8a)} \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad N \le 1. & \text{(8b)} \end{cases}$$

If there is only one connection taking part in the contention, or even none of them, then the backoff start value of zero suffices.

C. Stations with different backoff window values

In the previous subsection, we have presented a case when all the SSs have identical backoff windows. However, if SSs enter the contention resolution mechanism at different moments of time, they will have different backoff window values. In this subsection, we present a more general estimation that accounts for this fact.

The general considerations are exactly the same as for (3). The only difference is that now each SS has a different backoff window:

$$p_k = \sum_{i=1}^{W_k} \frac{1}{W_k} \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right). \tag{9}$$

It is easy to show that if all W_j have the same value, then (9) yields (3).

Unfortunately, one can apply (9) only for the theoretical analysis. The BS cannot use effectively (9) because it is not aware of the *concrete* backoff window values SSs use. As the BS announces the backoff start value in the UCD message, it does not receive any feedback about the backoff window each SS has at a particular moment of time. In other words, the BS knows the number of inactive connections, but it does not know whether they take part in the contention, and if they do, what the backoff window values are.

Lemma 1: Expression (3) is the worst-case estimation for a set of connections having different backoff window values.

Proof: Suppose, there are N connections taking part in the contention resolution mechanism. All the connections use the backoff window value of W_j, while the BS assumes that they are at the first stage of the contention resolution process, thus, having the backoff window value $W^*, W^* \le W_j, \forall j$. Taking into account (3) and (9), we will show that:

$$\prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) \le \sum_{i=1}^{W_k} \frac{1}{W_k} \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right). \tag{10}$$

It is possible to rewrite (10) in the following form:

$$\sum_{i=1}^{W_k} \prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) \le \sum_{i=1}^{W_k} \left(1 \times \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right)\right). \tag{11}$$

In turn, (11) can be written as a set of sums:

$$\sum_{i=1}^{W^*} \prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) + \sum_{i=W^*+1}^{W_k} \prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) \le \sum_{i=1}^{W^*} \left(1 \times \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right)\right) + \sum_{i=W^*+1}^{W_k} \left(1 \times \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right)\right). \tag{12}$$

For the sake of clarity, we will present separately that

$$\sum_{i=1}^{W^*} \prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) \le \sum_{i=1}^{W^*} \left(1 \times \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right)\right), \tag{13a}$$

$$\sum_{i=W^*+1}^{W_k} \prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) \le \sum_{i=W^*+1}^{W_k} \left(1 \times \prod_{\substack{j=1, j\ne k \\ W_j \ge i}}^{N} \left(1 - \frac{1}{W_j}\right)\right). \tag{13b}$$

First, consider expression (13a). Since $W^* \le W_j, \forall j$, (13a) simplifies to

$$\prod_{j=1}^{N-1} \left(1 - \frac{1}{W^*}\right) \le \prod_{j=1, j\ne k}^{N} \left(1 - \frac{1}{W_j}\right). \tag{14}$$

It is easy to show that a product of values on the right hand side has a larger result when compared to a product of values on the left hand side. In fact, the same holds for expression (13b). A sum of products on the right hand side has always a larger result because $W^* \le W_j, \forall j$ and due to the fact that there are less colliding stations, as governed by $W_j > i$. Thus, the probability determined by (3) is less than the one determined by (9). As a result, (3) is the worst-case estimation. ■

D. Backoff end

Another important parameter of the contention resolution mechanism is the backoff end value E. When the backoff window reaches E, an SS drops an SDU and starts the contention resolution process from the beginning for a next SDU, if any. In fact, $E{-}S{+}1$ determines the number of attempts an SS makes while trying to send the bandwidth request. While p determines the initial probability that an SS transmits successfully the bandwidth request during the *first* attempt, we can also calculate the *final* probability that results from the fact that an SS can make several attempts. Denoted as $\mathbf{P}$, this worst-case probability can be calculated as follows:

$$\mathbf{P} = \sum_{i=1}^{E-S+1} p(1-p)^{i-1} = 1 - (1-p)^{E-S+1}. \tag{15}$$

By setting p and $\mathbf{P}$, one can determine the backoff end value:

$$E = \min\left\{\left\lceil \frac{\log_2(1-\mathbf{P})}{\log_2(1-p)} \right\rceil + S - 1, 15\right\}. \tag{16}$$

As in the case of the backoff start value (5), the backoff end value is encoded with four bits; its maximum value is 15.

E. Number of the contention slots

While the backoff parameters determine the probability that an SS sends successfully the bandwidth request, the number of the contention transmission opportunities has an impact on how *fast* the contention resolves. Indeed, a randomly generated backoff window instructs an SS to wait a certain number of transmissions before sending the request. The more transmission opportunities a frame has, the less time an SS would wait. Initially, an SS can wait 2^S transmission opportunities before it sends the bandwidth request. However, since there is a probability that a collision occurs, an SS can make subsequent attempts until the backoff window reaches the backoff end value E. Thus, the worst case estimation for the number of transmission opportunities an SS will wait is

$$\sum_{i=S}^{E} 2^i = -2^S(1 - 2^{E-S+1}) = 2^{E+1} - 2^S. \tag{17}$$

Knowing the worst-case number of the transmission opportunities an SS can potentially wait, it is possible to calculate the worst-case time by taking into account the number of frames per second and the number of transmission opportunities per a single frame. It is also necessary to account for the fact that an SS has to wait a specified number of frames before it starts to retransmit the bandwidth request. The worst-case time T, which we will refer to as the *medium access delay*, can be determined as follows:

$$T = \frac{2^{E+1} - 2^S}{N_T \mathrm{FPS}} + \frac{I}{\mathrm{FPS}}(E - S), \tag{18}$$

where FPS stands for the number of frames per second, N_T stands for the number of transmission opportunities per one frame, and I stands for the number of frames to receive before the the contention reservation is attempted again. Expression (18) can be used with 802.16d and 802.16e standards. The only difference is that 802.16d standard defines I/FPS as a constant value. For the sake of clarity we will treat it accordingly and denote as C.

Based on (18), the number of the transmission opportunities can be determined by the following expression:

$$N_T = \left\lceil \frac{2^{E+1} - 2^S}{\mathrm{FPS}(T - C(E - S))} \right\rceil. \tag{19}$$

To calculate the number of the contention transmission opportunities, a provider has to set the medium access delay. Fig. 1 presents the graphical interpretation of T. Suppose, there is one request transmission opportunity in each frame and an SS has to defer two transmission opportunities before sending the bandwidth request. As follows from the figure, T is the time to complete the contention resolution plus one frame duration. We should account for an additional frame because even though the BS receives successfully the bandwidth request in the i+2th frame, the uplink data grant will be allocated only in the next one. This is due to the fact that the BS does the scheduling decision in the beginning of a frame.

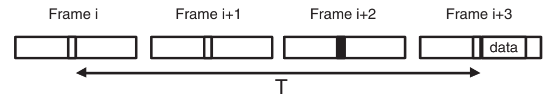


Fig. 1. Medium access delay

It is worth mentioning a particular case of (19) when the backoff start and backoff end values are the same. In this case (19) yields

$$N_T = \left\lceil \frac{2^S}{\mathrm{FPS} \cdot T} \right\rceil. \tag{20}$$

This form is suitable for the VoIP applications supported by the ertPS class. In this case, T should be less or equal to the packet inter-arrival time that can be calculated from the VoIP QoS profile. Since E equals S, there is only one attempt to send the bandwidth request. If an attempt fails, the VoIP SDU is dropped. It may affect the speech quality but, on the contrary, it preserves the timing requirements, which are very important for the real-time interactive applications. Of course, (20) is not suitable for the TCP applications performance of which depends significantly on packet drops. The simulations results for the adaptive backoff parameters and the VoIP connections are presented in [9].

III. Simulation

This section presents the simulation results for the analytical calculations. To run simulations, we have implemented the 802.16 MAC and PHY levels in the NS-2 simulator [10]. The MAC implementation contains the main features of the 802.16 standard, such as downlink and uplink transmission, packing and fragmentation, the bandwidth requests, the contention and ranging periods. We have also implemented the most important MAC signaling messages, such as UL-MAP, DL-MAP, UCD, DCD, ranging, dynamic service addition, etc. The implemented PHY layer is OFDM. However, since we focus on the contention resolution that is done on the MAC level, we do not simulate errors in the physical channel. The PHY layer affects only the frame size, slot duration, and size of some MAC level management messages.

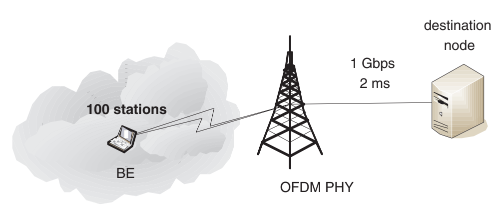


Fig. 2. Network structure.

Fig. 2 shows the network structure we use in the simulation scenarios. There are 100 SSs and one BS that controls the traffic of the WiMAX network, parameters of which are presented in Table I. Since we do not simulate errors at the PHY level, the ARQ mechanism is disabled. The BS shares resources between the BE connections according to the algorithm presented in [2]. In a few words, the BS allocates resources fairly between all the connections by taking their bandwidth request sizes into account.

TABLE I
WIMAX NETWORK PARAMETERS.

Parameter	Value
PHY	OFDM
Bandwidth	7 MHz
Cyclic prefix length	1/32
TTG+RTG	160 PS
Frames per second	50 (20 ms per frame)
Slots per frame	603
Duplexing mode	TDD
UL/DL ratio	dynamic
MCS	64-QAM3/4 (108 B/slot)
Packing/fragmentation	ON
ARQ/CRC	OFF
MAC PDU size	as large as possible

For the sake of clarity, each SS establishes one uplink and downlink transport BE connection to the BS (each SS also establishes the basic management connection to exchange the management messages with the BS). An SS hosts exactly one Pareto application that sends data over the TCP protocol to a wired node. The reason all the applications send data in the uplink direction is the fact that no contention occurs in the downlink direction. The BS always knows the amount of data it has to send downlink based on the state of its internal queues. The reason we choose the TCP protocol is that its throughput is sensitive to packet losses that may be caused by the wrong backoff parameters. Of course, there is also the downlink traffic because the TCP protocol sends acknowledgments to the transmitting applications. We choose the Pareto application because it sends data from time to time. When an application does not send data, the bandwidth request size equals zero and the BS does not allocate slots at all. Later, when an application wants to send data, it has to participate in the contention. It results in the varying number of the inactive connections (see Fig. 3) which, in turn, triggers the recalculation of the backoff parameters at the BS. New parameters are announced to SSs through the UL-MAP and UCD messages.

Before an application can start to send data, an SS has to register to the BS and establish the transport connections. Due to a significant number of SSs, we inject gradually all the SSs into the network during the first 20 seconds of a simulation run, as can be seen from Fig. 3. Since our simulator supports the initial-ranging and the connection setup procedures, an attempt to add all the SSs at the same time would result in a longer connection-setup delay.

To analyze the parameters of the contention resolution mechanism, we compare simulation results when the BS relies upon the static configurations and when the BS updates the backoff parameters adaptively. Regardless of a simulation case,

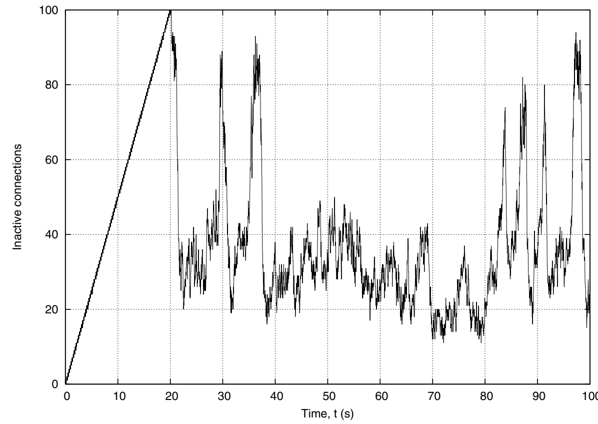


Fig. 3. Number of inactive uplink connections.

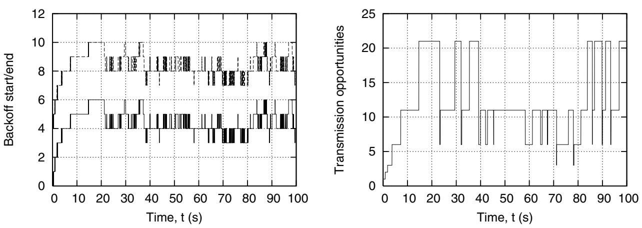
all the applications start to send data at the 20th second of the simulation run and application activity pattern is the same. For each case, Table II presents the amount of data transferred by all SS in the uplink direction, the maximum medium access delay, and the loss ratio. As follows from the results, a small backoff start value and a few request contention opportunities cannot ensure an efficient connection resolution due to a huge number of collisions (case I). Some SSs were not capable even of starting to send data. Even a bigger backoff start value results in a slightly better contention resolution process (case II). However, regardless of the backoff start value, failing to set up a proper backoff end value results in an unacceptably large medium access delay. There is no need to say that the medium access delay of more than 60 s is not suitable for any service. At the same time, it is interesting to note that the large backoff end value results in a low loss ratio – no packets were dropped in our simulation case. This fact is explained by expression (18). If we substitute the PHY parameters and the backoff parameters from cases I and II into that expression, we receive a big worst-case medium access delay. Thus, an SDU can reside for a long period of time in the SS output queues without being dropped. Not surprisingly, increasing the number of the transmission opportunities per frame results in a larger amount of data the SSs can transfer and a smaller medium access delay (case III). Though a larger number of the request transmission opportunities results in even a smaller medium access delay, the SSs send less data because fewer slots remains for the user transport connections (case V). The best results are achieved when the BS updates parameters

TABLE II
SIMULATION RESULTS.

	Backoff start	Backoff end	Transm. opp.	Data (MB)	Max.access delay (s)	Loss (%)
I	0	15	1	58.610	68.52	0
II	6	15	1	82.566	72.90	0
III	6	10	21	165.352	1.58	0.004
IV	p=0.1, $\mathbf{P}$=0.35, T=2			181.905	1.69	0.017
V	6	10	30	156.503	1.05	0.001

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

adaptively based on the number of the inactive connections (case IV): the medium access delay is within the required boundaries and the amount of the transferred data is larger when compared to other static cases. Fig. 4(a) and Fig. 4(b) show the calculated backoff start/end values and the number of the request transmission opportunities. It is not complicated to correlate the number of the inactive connections presented in Fig. 3 with the calculated values. As a number of inactive uplink connections increases, the BS announces bigger backoff start/end values (to increase the probability for a successful transmission) and starts to allocate more request contention slots (to ensure the required medium access delay). Referring back to Table II, the packet loss ratio is very low and can be tolerated by many services.



(a) Backoff start/end (b) Transmission opportunities

Fig. 4. Dynamically calculated parameters.

Fig. 5 presents the medium access delay of all the SSs for the adaptive case. X-axis is the time when an SS starts the contention resolution and Y-axis is the medium access delay. It is quite easy to correlate spikes in Fig. 5 with moments of time in Fig. 3, when the number of inactive connections decreases drastically, i.e., when a significant number of connections take part in the contention resolution. Despite it, as follows from the figure, all the timing requirements are ensured; the medium access delay is less than 2 s. In [9] we showed that the adaptive approach is capable of ensuring very tight delay requirements of the VoIP services. In particular, we ensured the target medium access delay of 30 ms.

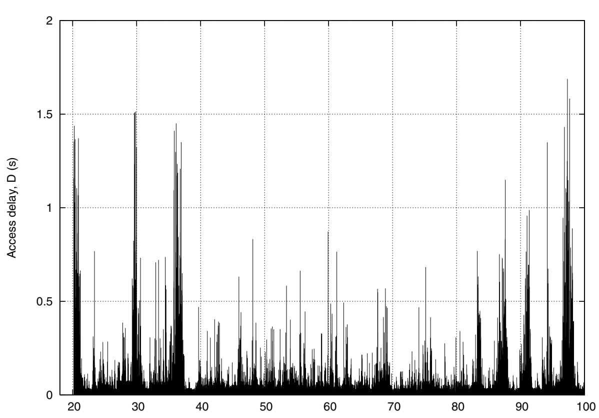


Fig. 5. Medium access delay (adaptive backoff case).

IV. Conclusions

In this paper, we have presented the analytical calculations to find the backoff start/end values and the number of the transmission opportunities for the 802.16 contention resolution mechanism. The input parameters are the initial and final probabilities, and the required time for the contention resolution. The simulation results have confirmed the correctness of the proposed calculations. The adaptive parameter tuning provides better results when compared to a pure static case: SSs send more data and the medium access delay is within the required boundaries. Furthemore, adaptive parameters eliminate a need to find the optimal static parameters.

The analytical expressions presented for the bandwidth request contention resolution can also be applied to the ranging contention period because it relies upon exactly the same mechanism. Finding the optimal ranging parameters may be a crucial task for the mobile 802.16 networks in which mobile stations enter and leave cells intensively.

Our future research works will aim at extending our calculations for the 802.16 OFDMa PHY. Though it has the same contention resolution mechanism, an SS sends a special CDMA code instead of the bandwidth request. Since the BS can detect orthogonal CDMA codes sent at the same time, OFDMa PHY provides a better performance. Some analytical results are presented in [11].

References

[1] "Air interface for fixed broadband wireless access systems," IEEE Standard 802.16, Jun 2004.

[2] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," in *The 9th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Oct 2006, pp. 108–117.

[3] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard 802.11, 1999.

[4] B. Kwak, N. Song, and L. Miller, "Performance analysis of exponential backoff," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, Apr 2005.

[5] S.-M. Oh and J.-H. Kim, "The optimization of the collision resolution algorithm for broadband wireless access networks," in *The 8th International Conference on Advanaced Comunication Technology*, vol. 3, Feb 2006, pp. 1944–1948.

[6] M. Hawa and D. Petr, "Quality of service scheduling in cable and broadband wireless access systems," in *IEEE International Workshop on Quality of Service*, May 2002, pp. 247–255.

[7] D.-H. Cho, J.-H. Song, M.-S. Kim, and K.-J. Han, "Performance analysis of the IEEE 802.16 wireless metropolitan area network," in *International Conference on Distributed Frameworks for Multimedia Applications*, Feb 2005, pp. 130–136.

[8] "Air interface for fixed broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands," IEEE Standard 802.16e, Dec 2005.

[9] A. Sayenko, O. Alanen, and T. Hämäläinen, "Adaptive contention resolution for VoIP services in IEEE 802.16 networks," in *The 8th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Jun 2007.

[10] UCB/LBNL/VINT, "Network simulator ns-2," 1997, http://www.isi.edu/nsnam/ns.

[11] J.-B. Seo, N.-S. Lee, N.-H. Park, H.-W. Lee, and C.-H. Cho, "Performance analysis of IEEE 802.16d random access protocol," in *IEEE International Conference on Communications*, vol. 12, Jun 2006, pp. 5528–5533.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.