ADAPTIVE CHANNEL SCANNING FOR IEEE 802.16e

Richard Rouil and Nada Golmie
National Institute of Standards and Technology
100 Bureau drive, stop 8920
Gaithersburg, MD 20899-8920
USA
richard.rouil@nist.gov, nada.golmie@nist.gov

Abstract—**In this article we propose an adaptive algorithm that determines the duration and frequency of channel scanning in order to facilitate the discovery of neighboring base stations and handovers across multiple IEEE 802.16 networks. The proposed algorithm supports application quality of service requirements and can be generalized to multiple mobile devices concurrently performing channel scanning. Performance results for select simulation scenarios are presented and discussed.**

I. INTRODUCTION

In the context of ubiquitous connectivity, a Mobile Station (MS) equipped with an IEEE 802.16 interface is likely to roam across multiple base stations (BS) in order to maintain connectivity. While the IEEE 802.16 technology also known as WiMAX was developed for fixed broadband connectivity in mind, extensions of the standard specifications, namely the IEEE 802.16e [2], have been developed in order to support mobility across multiple base stations.

However, as in most mobility scenarios, finding the target BS that best fits the mobility path and application requirements is far from being trivial. Generally the mobile device needs to scan multiple channels in order to find neighboring BSs and select an appropriate target. This selection can be based on different criteria, for example, measured signal strength, packet delay, error ratio, throughput, and security levels. In addition, since channel scanning can be a relatively time consuming activity, it is preferable for the MS to perform this scanning and obtain a list of neighboring BSs before it is ready to perform a handover. In fact, the IEEE 802.16e standard supports temporarily suspending the communication between the BS and the MS in order for the mobile to perform channel scanning. Thus, during this scanning period, both upstream and downstream packets originating at the mobile and destined to it are buffered at the MS and BS respectively. The questions are: How to determine the duration and frequency of this scanning interval? What factors should be considered in order to interleave periods of scanning and normal operation? It is evident that determining the duration and frequency of channel scanning will have a direct impact on the application traffic and the resulting quality of service supported. A long scanning interval increases the packet jitter and the end-to-end delay, thus imposing large buffer sizes. On the other hand, a short scanning interval requires multiple iterations, thus increasing the overall scanning duration. Related work in the literature to date has mostly focused on reducing the number of MSs that need to perform channel scanning [4] [3]. These approaches assume that communication between BSs is available in order to disseminate information about neighboring BSs. While this assumption may hold true in a subset of scenarios where BSs belong to the same service provider network, channel scanning remains inevitable for most practical scenarios.

Therefore, in this article, we focus on how to determine the channel scanning frequency and duration assuming that channel scanning is unavoidable. We propose an Adaptive Channel Scanning (ACS) algorithm to allocate scanning intervals for multiple MSs while maintaining the quality of service requirements of the applications they are supporting.

The remainder of this paper is organized as follows. Section II describes the IEEE 802.16 interface and its mobility support. In section III, we present the proposed ACS algorithm. In section IV, we evaluate the performance of ACS and discuss simulation results for select scenarios of interest.

II. OVERVIEW OF IEEE 802.16(e)

The IEEE 802.16 standard [1] defines the mechanisms for a user equipment to connect to a BS. This so-called network entry phase, depicted in Fig. 1, consists of both synchronization and association operations. During the synchronization step, the MS receives broadcast messages, which are sent by the BS and contain information about how and when to access the channel. The downlink (DL_MAP) and uplink map (UL_MAP) messages contain burst allocation for each frame. The Downlink Channel Descriptor (DCD) and Uplink Channel Descriptor (UCD) contain transmission parameters of each burst. The synchronization phase is followed by the association operation, where the MS adjusts its timing and transmission power to communicate with the BS. During this step, also known as Initial Ranging, the MS randomly picks a ranging slot according to a truncated exponent algorithm. It then waits for a contention slot in an uplink frame in order to transmit its ranging request. The next steps following the network entry phase include basic capability negotiation, authentication, and registration. After successfully completing all of these steps, the MS is connected to the BS and an IP connection is established.

In [6] the authors show that the synchronization time depends mostly on the frequency of the synchronization messages and can be in the order of seconds. Therefore, if an MS needs to perform a network entry operation each time it performs a handover, any ongoing connections it has may be severely disrupted.

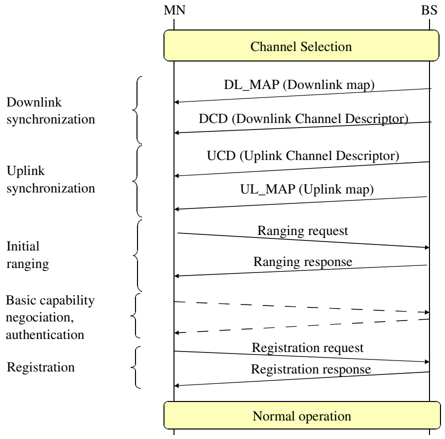


Fig. 1. Network Entry in IEEE 802.16

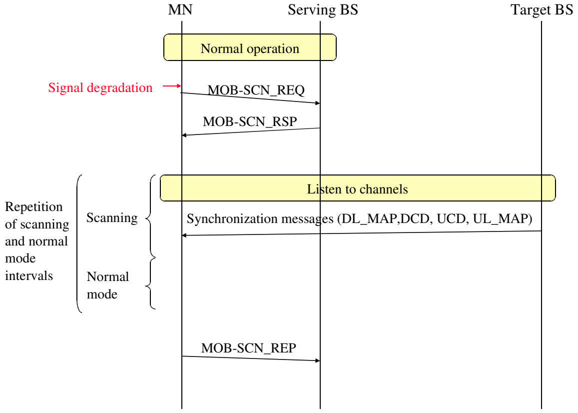


Fig. 2. Scanning in IEEE 802.16e

To overcome this problem, the 802.16e extension to the standard defines several mechanisms related to BS communication and channel scanning in order to facilitate neighbor discovery and handovers. Regarding BS communication, the assumption in IEEE 802.16e is that neighboring BSs exchange downlink and uplink channel descriptors (DCD and UCD messages) over the backbone. The information is then embedded in messages sent periodically by the serving BS to the MSs. This allows an MS to acquire channel information prior to any scanning. Mechanisms related to channel scanning are in the form of requests sent by the MS seeking to maintain information about neighboring BSs as shown in Fig. 2. The MS sends a MOB-SCN_REQ message to the serving BS that processes the information and returns the scanning interval information using a MOB-SCN_RSP message. This coordination between the MS and the BS allows each entity to buffer packets while the communication is temporarily suspended. At the end of the scanning, the MS reports the scan status to its serving BS in the form of a MOB-SCN_REP message.

Furthermore, the 802.16e specifications define four modes of channel scanning depending on the level of association desired.

- In scan without association, the MS attempts to identify and synchronize with one or more BSs. It also estimates the signal quality.
- In association level 0, the target BS has no information about the scanning MS and only provides contention-based ranging allocations. After sending a ranging request, the MS waits for a response from the BS with a default timeout value of 50ms.
- In association level 1, the serving BS negotiates with the target BSs a time at which the MS will find a dedicated ranging region. After sending a ranging request, the MS waits for a response from the BS with a default timeout value of 50ms.
- In association level 2, also called network assisted association reporting, the procedure is similar to level 1 except that the MS does not wait for a response from the target BS. The ranging response is sent by the target BS to the serving BS, which then forwards it to the MS.

III. THE ADAPTIVE CHANNEL SCANNING (ACS) ALGORITHM

The main objective of the ACS algorithm is to minimize the disruptive effects of scanning on the application traffic. It consists of two main components, namely, (1) estimating the time needed by an MS to scan a list of neighboring BSs and (2) interleaving of channel scanning and data transmission intervals.

A. *Estimating the channel scanning time*

The channel scanning time depends on the synchronization and the association latencies related to the MS association level. Let l_s and l_a represent the synchronization and association latencies, respectively.

As described in section II, the synchronization with the target BS consists of receiving the downlink and uplink channel descriptors and frame allocation information contained in the DL_MAP, UL_MAP, DCD, and UCD messages. While the DCD and UCD messages are sent periodically by the serving BS, the DL_MAP and UL_MAP are generally contained within a single frame sent by the target BS. Therefore, an estimate for the average synchronization latency l_s is in the order of two frame cycles.

On the other hand, the association latency, l_a depends on the association level considered. For association level 0, performing contention based ranging requires the MS to select a backoff window using the backoff exponent B_{exp} provided by the target BS. Let M be the maximum number of contention slots that an MS needs to wait before sending its ranging request. $M = 2^{B_{exp}} - 1$. Let n_{cs} be the number of contention slots per frame. The number of frames, n_f, that an MS needs to wait before sending its ranging request can then be estimated according to:

$$n_f = \lfloor \frac{M}{n_{cs}} \rfloor \tag{1}$$

Base station		l_a (ms)	l_s (ms)	t_{st} (ms)
Parameter	Value			
BS1		8	110	118
Association level	0			
B_{exp}	4			
n_{cs} (slot/frame)	1			
t_f (ms)	4			
BS2		8	50	58
Association level	1			
t_f (ms)	4			
BS3		8	306	314
Association level	0			
B_{exp}	6			
n_{cs} (slot/frame)	2			
t_f (ms)	8			

TABLE I

EXAMPLE OF CHANNEL SCANNING TIME COMPUTATION FOR DIFFERENT BS CONFIGURATIONS

The association latency, l_a, follows:

$$l_a = n_f * t_f + t_{out} \tag{2}$$

where t_f is the frame duration and t_{out} is a timeout interval expressed in ms. Thus for a given BS, the total channel scanning time, t_{st} is defined as $l_a + l_s$, and for N neighboring BSs

$$t_{st} = \sum_{i}^{N} (l_s + l_a) \tag{3}$$

Numerical examples for computing t_{st} are given in Table I for different BS configuration parameters. Considering BS1 with $B_{exp} = 4$, $t_f = 4ms$, and $n_{cs} = 1$, $t_{st} = 2 * 4 + \lfloor \frac{2^4-1}{1} \rfloor * 4 + 50 = 118ms$. Similarly, $t_{st} = 314ms$ for BS3.

For association level 1, there is no contention based ranging therefore $l_a = t_{out}$.

While each BS can estimate its own t_{st}, computing estimates of t_{st} for neighboring BSs, may require few extensions. We propose extending the messages exchanged between the BSs over the backbone in order to include the average number of contention opportunities per frame and the association levels supported by the BSs. Additional information including B_{exp}, and t_f is already contained in the UCD and DCD messages.

B. Interleaving of channel scanning and data transmission intervals

The serving BS computes the channel scanning duration, the time between two channel scanning operations, and the number of scanning intervals. The ACS algorithm uses the application quality of service requirements and available bandwidth in order to interleave channel scanning operations within data transmission periods.

Let n_s be the number of scanning intervals, t_s be the channel scanning interval, and t_d be the interval between two scanning operations. Let the available bandwidth be denoted by BW in byte/s. Let N be the number of MSs that request to perform channel scanning concurrently. The traffic requirements of MS_i include, its throughput, b_i in byte/s, its maximum tolerated jitter, j_i in seconds, and its maximum tolerated latency, l_i in seconds.

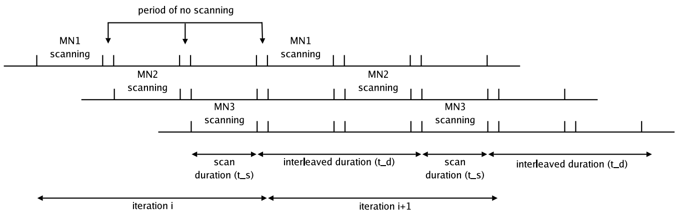


Fig. 3. An example of staggering channel scanning intervals for three MSs

In order to limit the maximum tolerated latency and jitter occurring during the channel scanning operation, t_s is set to the minimum of all jitter and latency requirements of all participating MSs as follows:

$$t_s = \min(j_i, l_i); \quad \forall i \tag{4}$$

The algorithm computes the number of scanning intervals using:

$$n_s = \lceil \frac{t_{st}}{t_s} \rceil \tag{5}$$

In order to make the most efficient use of the bandwidth available, while one MS is performing channel scanning, others can be transmitting data traffic. Therefore, staggering the channel scanning intervals as shown in Fig. 3 yields:

$$t_d = (N - 1) * t_s \tag{6}$$

Based on the throughput requirements of the scanning stations, additional time for data transmission may be needed, and t_d is adjusted as a function of the queue length, Q_i and the data transmitted by other MSs while MS_i is scanning, D_i.

The maximum time t_{dmax} interleaved between scanning intervals is defined as $255 * t_f$ since the number of interleaved frames is encoded with 8 bits in the standard. This value is the upper bound assigned to t_d.

The details of the ACS algorithm are given below.

IV. PERFORMANCE RESULTS

To evaluate the performance of the ACS algorithm we performed simulations using a model of IEEE 802.16e developed in the network simulator, NS-2 [5]. This section presents the assumptions used in the simulation set-up and the results obtained.

A. Simulation set-up and performance metrics

The network topology used in the simulations is shown in Fig. 4. We consider three MSs connected to a serving BS that is implementing the ACS algorithm. We assume that before leaving the coverage area of the serving BS, an MS requests to perform a channel scanning as described in section II. There are three neighboring BSs including BS1, BS2, and BS3, configured according to Table I. We assume MS1 and MS3 are running video applications, while MS2 is running an audio application. The traffic directionality is from the BS to the MS. The quality of service requirements for the audio and video applications are described in Table II.

ACS Algorithm

```
$t_s = \min(j_i, l_i) \forall i$
$t_d = (N-1) * t_s$
$n_s = \frac{t_{st}}{t_s}$
$t_{dmax} = 255 * t_f$
for $i = 1$ to $N$ do
    $Q_i = t_s * b_i$
    $D_i = t_s * (BW + b_i)$
end for
for $i = 1$ to $N$ do
    for $j = 1$ to $N-1$ and $Q_i > 0$ do
        $k = (j+i) \bmod N$
        if $Q_i \leq D_k$ then
            $D_k = D_k - Q_i$
            $Q_i = 0$ {No more buffered data left}
        else
            $Q_i = Q_i - D_k$
            $D_k = 0$ {Have buffered data left}
        end if
    end for
    if $Q_i > 0$ then
        {Need more bandwidth in normal mode}
        if $BW = 0$ then
            $t_d = t_{dmax}$
        else
            $t_d = \min(t_d + \frac{Q_i}{BW}, t_{dmax})$
        end if
    end if
end for
for $i = 1$ to $N$ do
    $MS_i$ starts scanning at $(i-1) * (t_s + \frac{t_d - (N-1)*t_s}{N})$
end for
```

QOS Parameters	Requirements
Video	
data rate (byte/s)	49600
jitter (ms)	100
delay (ms)	200
Audio	
data rate (byte/s)	8000
jitter (ms)	50
delay (ms)	75

TABLE II

QUALITY OF SERVICE REQUIREMENTS FOR AUDIO AND VIDEO APPLICATIONS

We perform two sets of experiments and vary the number of MSs requesting to perform a channel scan. In all experiments, the load of the serving BS varies from 10% to 100% of the total channel capacity. The load is defined as percentage of the channel capacity.

Performance metrics include the packet delay, jitter, and total scanning time. Delay is defined as the time to transmit the packet at the base station. Jitter is defined as the difference between the expected packet arrival time and the measured packet arrival time. The total scanning time is t_{st} defined in section III-B as the time required to complete channel scanning.

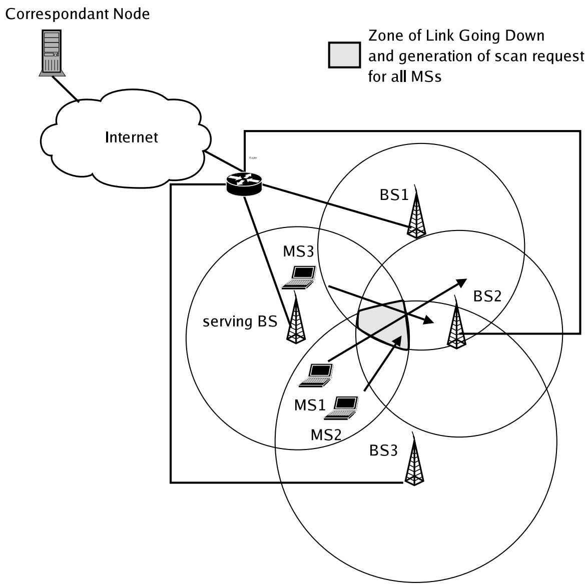


Fig. 4. Network topology

B. Experiments with a single scanning MS

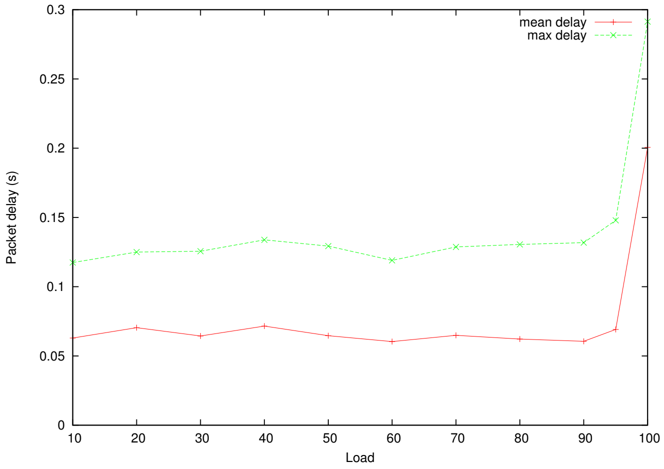
In this set of experiments, we assume that a single MS is performing a channel scanning operation. Fig. 5 and 6 show the delay and total scanning time for an MS when it is receiving either video or audio traffic.

The ASC algorithm sets the scanning duration to the minimum of jitter and delay requirements. Therefore, $t_s = 100ms$ for video traffic and $t_s = 50ms$ for audio traffic. These values become the new requirements for the jitter and delay.

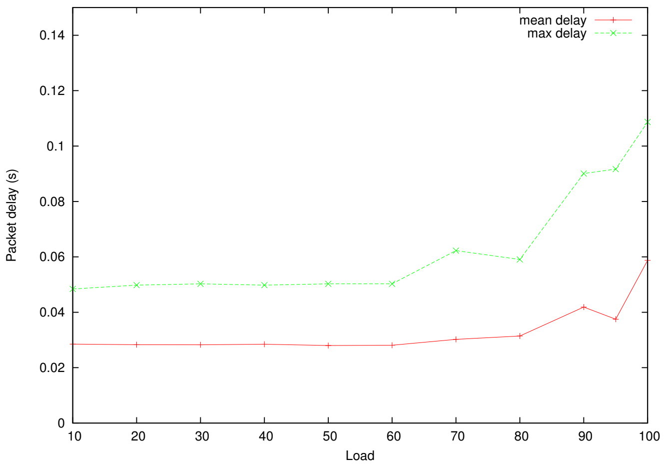
In Fig. 5(a), the average delay for a single MS receiving video traffic is kept below the 100ms requirements for loads up to 95%. However, the delay for some packets may still exceed this threshold since the maximum delay observed is around 140ms for loads below 95%. In Fig. 5(b), the average delay for a single MS receiving audio traffic is also kept below the 50ms requirement for loads up to 95%. Similarly, we observe a maximum packet delay exceeding the threshold for loads over 60%. As the load approaches the channel capacity, the time allocated for normal data transmission is no longer sufficient for sending all the data buffered during the scanning period. As a result higher packet delays are incurred.

Recall that the ASC algorithm estimates the scanning time required for three neighboring BSs to be $t_{st} = 490ms$. Thus, when the MS is receiving video traffic, $t_s = 100ms$ and there are five scanning intervals ($n_s = 5$). When the MS is receiving audio traffic, $t_s = 50ms$ requiring ten iterations ($n_s = 10$). From Fig. 6(a) and Fig. 6(b), we observe that when the load increases, the time spent to perform channel scanning increases as well. This is due to the algorithm compensating less available bandwidth with longer normal mode intervals. The maximum scanning time is 5.6s when an MS is supporting video traffic. This corresponds to five iterations with $t_d = 1.02s$ between two scanning operations. When the MS is supporting audio traffic, there are ten iterations, thus the maximum scanning time is $10.7s$.

Observe that there is no packet loss in these experiments

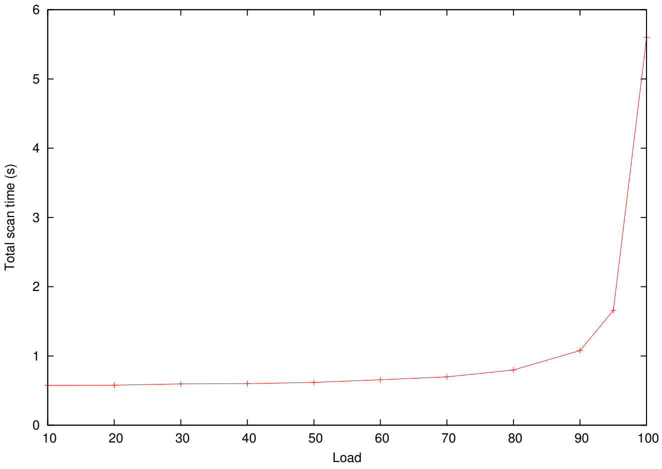


(a) Packet delay with single MS supporting video traffic

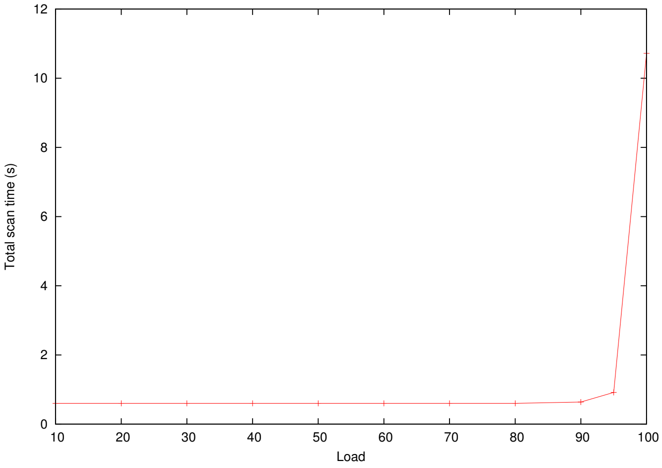


(b) Packet delay with single MS supporting audio traffic

Fig. 5. Packet delay measurements with a single MS



(a) Total scanning time with single MS supporting video traffic



(b) Total scanning time with single MS supporting audio traffic

Fig. 6. Total scanning time measurements with a single MS

since we assume limitless buffers.

C. *Experiments with multiple scanning MSs*

To evaluate the performance of the ACS algorithm when multiple nodes are requesting a channel scanning time, we assume in the next set of experiments that three MSs are leaving the cell at the same time. The ACS algorithm determines that an MS uses the bandwidth provided during the scanning of the other MSs. Therefore, $t_s = 50ms$, $t_d = 100ms$, and $n_s = 10$. The expected total scanning time is then $10 * (50 * 100) = 1500ms$. In the next set of experiments, the total scanning time measured is equal to $1.5s$.

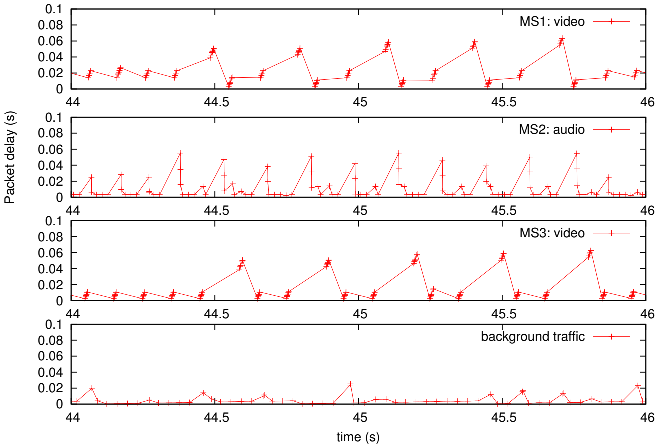
Fig. 7(a) shows the instantaneous packet delay (over time) measured for all three scanning MSs and other non-scanning MSs (labeled ”background traffic”). Observe that the packet delays stay below the 50ms and 100 ms limits for audio and video traffic respectively. The peak delay corresponds to the first packet sent after the scanning interval. Also, the delay for the background traffic is minimally impacted by the channel scanning operation. On the graph showing the MS receiving audio traffic, observe ten distinct peaks representing the scanning iterations starting at time 44.3s and ending at time 45.8s. For the video, only five peaks are observed since the packet inter-arrival is 100ms. Thus, for half of the scanning interval, there are no packets in the buffer. Fig.7(b) also shows that the maximum packet jitter is under 60ms.

Fig. 8 shows the average delay for a variable load. Notice the delay increases as the load increases but stays under the 50ms requirement. Even though an MS has enough bandwidth to receive the buffered data while others are scanning the channel, the time required increases as the available bandwidth decreases, and thus the delay increases.

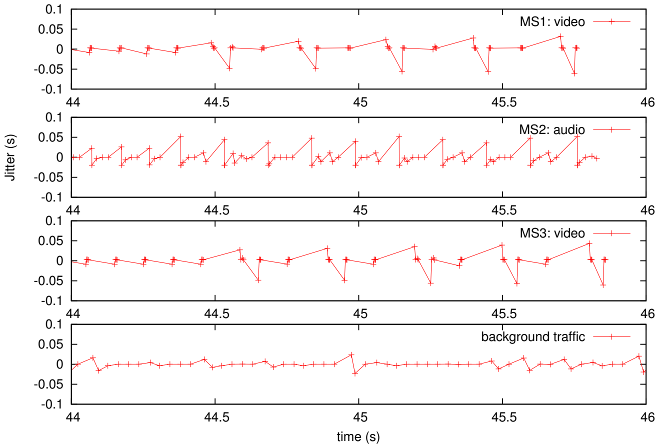
V. CONCLUSION

We presented the Adaptive Channel Scanning (ACS) algorithm to allocate scanning intervals in IEEE 802.16e. This algorithm relies on sharing configuration parameters of neighboring base stations in order to estimate the total scanning time required for a mobile station. This time is then interleaved by periods over normal data transmission in order to limit any disruptions on the data flow and support application quality of service requirements.

Simulation results to evaluate the performance of ACS were also presented and discussed. The results show that using the



(a) Packet delay over time with three MSs



(b) Packet jitter over time with three MSs

Fig. 7. Impact of scanning on traffic with three concurrent MSs

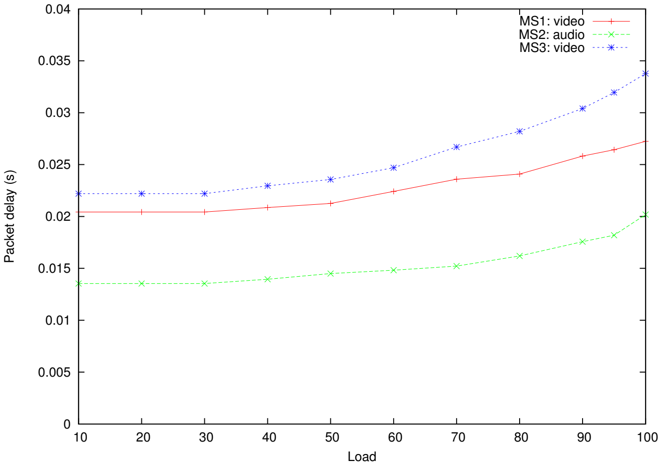


Fig. 8. Average packet delay during scanning of three concurrent MSs

application traffic characteristics in order to allocate periods of scanning, make it possible to limit communication disruptions for all traffic. A critical point in the proposed algorithm is the measurement of the available bandwidth. Inaccurate measurements can cause the algorithm to allocate either too much or too little time between scanning iterations.

Future work will investigate the use of ACS in discovering BSs and establishing rendez-vous times when using a different association level, namely, the associations level 1 and 2.

REFERENCES

[1] IEEE Std 802.16-2004. IEEE standard for local and metropolitan area networks- part 16: Air interface for fixed and mobile broadband wireless acess systems., October 2004.

[2] IEEE Std 802.16e 2005. IEEE standard for local and metropolitan area networks - part 16: Air interface for fixed and mobile broadband wireless acess systems.amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1, February 2006.

[3] S. Choi, G. Hwang, T. Kwon, A. Lim, and D. Cho. Fast handover scheme for real-time downlink services in ieee 802.16e bwa system. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, pages 2028 – 2032, January 2005.

[4] L. Doo Hwan, K. Kyamakya, and J.P. Umondi. Fast handover algorithm for ieee 802.16e broadband wireless access system. In *Wireless Pervasive Computing, 2006 1st International Symposium on*, pages 1 – 6, January 2006.

[5] Ns-2 simulator. http://nsnam.isi.edu/nsnam/index.php/main_page.

[6] R. Rouil and N. Golmie. Effects of ieee 802.16 link parameters and handover performance for select scenarios, March 2006.