Performance Analysis for Polling Service in IEEE 802.16 Networks Under PMP Mode

Haitham Abu-Ghazaleh, Jun Cai, and Attahiru Sule Alfa
Department of Electrical and Computer Engineering
University of Manitoba, Canada.
Email: {haitham, jcai, alfa}@ee.umanitoba.ca

***Abstract*—In this paper, we focus on the polling mechanism adopted in IEEE 802.16 networks operating in PMP (Point-to-Multipoint) mode between the base station and the subscriber stations. We develop a queueing model for computing the performance of the polling service traffic, which takes into account the polling period amongst the various subscriber stations served by a single base station. Using our model, the waiting time distribution, the blocking probability, and other relevant performance measures are computed. The model can be used to investigate how the traffic in the polling service for each subscriber station can be allocated the necessary network resources to meet some certain Quality-of-Service (QoS) requirements.**

I. Introduction

Network service differentiation and QoS considerations make up a significant part of the IEEE 802.16 protocol. This is achieved by assigning the network traffic to one of three priority levels/services: 1) Unsolicited Granted Service (UGS) which handles traffic of the highest priority, 2) Polling Service (PS), and 3) Best-Effort service (BE) for traffic that requires no QoS guarantee. Bandwidth allocation for UGS traffic is usually handled in a static manner, while BE traffic exploits the remaining bandwidth after *enough* has been allocated for the other two higher priority services, namely PS and UGS.

In this work, we will focus our attention on IEEE 802.16 networks that operate in the Point-to-Multipoint (PMP) mode. The nodes are organized in cellular-like structure (see Figure 1) with a central base station (BS), as opposed to the Mesh mode with ad hoc structure. According to the polling mechanism defined in the standards for the PMP mode, the BS is responsible for polling each subscriber station (SS) in specific intervals for their PS traffic [1]. The PS supports some level of QoS guarantee, and the amount of bandwidth that is allocated to each of the polled stations typically remains fixed, which depends on the amount of bandwidth assigned to the UGS traffic. However, each SS may be given a different service time period T (or slot times) during each polling period by considering the possible unbalance of traffic load in each SS.

The objective of this paper is to present a queueing model for computing the performance of the polling service traffic, which takes into account the polling periods amongst the various subscriber stations served by a single base station. Our model is used to compute the various performance measures such as blocking probability and waiting time distribution. The model could eventually provide a guideline as to how one can assign T for each SS with queued traffic, subject to some QoS constraints such as the blocking probability, mean waiting time, and other higher moments of waiting-time which can be computed from the distribution.

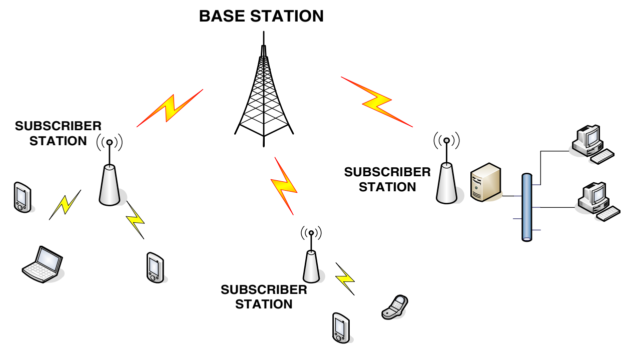

Fig. 1. Example of IEEE 802.16 Wireless Network Topology in PMP Mode.

This paper is organized as follows. After briefly reviewing some related works in Section II, the details of our model is described in Section III. Section IV presents the derivations of the relevant performance measures. A simple numerical example is given in Section V, followed by some conclusions in Section VI.

II. Previous Works

Recent works, such as [2] and [3], have focused on simulating systems running the IEEE 802.16 protocol and studying how the performance of the system varies with several factors in terms of throughput and delay. The authors in [4] and [5] have proposed their own admission control policies and scheduling algorithms to meet certain QoS requirements in such networks, and have demonstrated their effectiveness using simulations.

In [6], a queueing model was presented for dynamically allocating the bandwidth to the PS according to some variations in traffic load. However, their work does not consider the correlation with the other SSs in the network. Dynamic bandwidth allocation for PS in terms of mean queue-length and transmission period was modeled in [7] using the generating function approach. In [8], the authors considered the queueing performance of the PS, and proposed a bandwidth allocation scheme for the entire set of priority services that depends on the variations in the queue and traffic source, but ignored the polling period which likely underestimate their delay statistics.

978-1-4244-2202-9/08/$25.00 © 2008 IEEE

III. Model Description

We consider a system of a single BS serving N SSs. Each SS maintains its own packet buffer numbered $1, 2, \cdots, N$ for PS traffic, and each buffer i has a finite capacity of size $K_i < \infty$. There is only one server that attends to the buffers in a cyclic order. Each buffer i is attended by the server for a maximum of T_i time slots. The server attends to buffer i until the buffer becomes empty or the server has been attending to it for T_i time slots, whichever comes first. This results in an exhaustive time-limited polling system.

Studying this polling system with N SSs will result in a Markov Chain with huge dimensions. Instead we approximate this system by an exhaustive time-limited polling model with vacation [9] and analyze each queue separately. We focus on an arbitrary buffer i, and from the point of view of this queue the server is either attending to it or is away attending to other buffers. When the server is away attending to the other buffers this target buffer i sees the server as being away on a *vacation*. Our model further considers the correlation of the vacation periods and the service periods amongst all the PS buffers in the system, whereby each buffer's vacation period is equivalent to the service period of the other buffers in the polling system.

According to the scheduling service in IEEE 802.16, PS traffic can be categorized into two different classes: real-time PS (rtPS) for delay sensitive traffic and non-real-time PS (nrtPS). For simplicity, we will focus our attention on a single class of PS traffic. Moreover, we remove the descriptor i for notational convenience since we are only modeling one buffer.

A. Arrival Process

In our discrete-time model, time is discretized into time slots. During each time slot, we assume that exactly one packet that is at the head of the queue in service is processed, i.e. the service time for a single packet is one time slot. Our time slot quantum assumption implies that our packet arrivals could be in batches.

To capture the correlation in arrivals, if any, packets are assumed to arrive at the buffer according to the Batch Markovian Arrival Process (BMAP) [10] described by the sub-stochastic matrices D_x of order n, where $x = 0, 1, 2, \cdots, \eta$. The matrix D_x represents x packet arrivals during a time slot, with η being the maximum number of arrivals. Let $D = \sum_{x=0}^{\eta} D_x$ and $\boldsymbol{\pi} D = \boldsymbol{\pi}$, $\boldsymbol{\pi}\mathbf{1} = 1$ (where $\mathbf{1}$ is a column vector of ones), the average packet arrival rate can be computed as

$$\lambda = \sum_{x=1}^{\eta} x \boldsymbol{\pi} D_x \mathbf{1}. \tag{1}$$

B. Server Vacation Period

In our model, we let the server vacation period of buffer i be a phase type distribution (δ, L) of order θ. The server vacation period for buffer i is equivalent to the sum of the server visit periods of the remaining $N-1$ SSs.

In an exhaustive time-limited polling system, the server visit period at one station depends on the server visit periods at the remaining stations (see [11] and [12] for more details). Note that the server visit period depends on the station's queue length and may be less than the service time-limit T. This implies that there is a correlation in the queueing performance among all the polling stations. Hence, a station's vacation period is dependent on the remaining stations' server visit periods.

Since the focus of this paper is not on accurately modeling (δ, L) for the buffer under consideration, and for the sake of simplifying the presentation of our model, our analysis will assume the simple case where the buffers are so busy that it is not often that a server leaves a buffer before its limited visit time is up. This is especially true if each of the SS buffers experiences high traffic load, which accounts for the worst case conditions. Hence, buffer i's server vacation period is equivalent to the sum of the polling periods of the other $N-1$ SSs, which has a phase type distribution with $L = \bar{I}(\theta_i - 1)$ and $\delta = [1, 0, 0, \cdots, 0]$, where $\theta_i = \sum_{k=1}^{N} T_k - T_i$, such that

$$\bar{I}(\upsilon) \triangleq \begin{bmatrix} \mathbf{0} & I(\upsilon) \\ 0 & \mathbf{0} \end{bmatrix}$$

where $I(\upsilon)$ is an identity matrix of size υ, and $\mathbf{0}$ is a vector of zeros.

C. Markov Chain

By assuming that one packet is served during a single time slot, our BMAP/D/1/K system is operating like a number-limited vacation queue, i.e. server attends to a maximum of T packets during its visit. At time slot $t \geq 0$, let X_t be the number of packets in the system (buffer), Y_t the phase of arrival, U_t the clock time of the current visit at the buffer, and V_t the phase of vacation if the server is on vacation. Then, the state space of this process can be written as

$$\bar{\Delta} = \{(0, Y_t, V_t) \cup (X_t, Y_t, V_t) \cup (X_t, Y_t, U_t)\} . \tag{2}$$

The transition matrix of this Markov Chain is of the form

$$\mathbf{P} = \begin{bmatrix} C_0 & C_1 & C_2 & \cdots & C_\eta & & & & \\ E & A_1 & A_2 & \cdots & A_\eta & A_{\eta+1} & & & \\ & A_0 & A_1 & A_2 & \cdots & A_\eta & A_{\eta+1} & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & A_0 & A_1 & A_2 & \cdots & \hat{A}_i & \\ & & & & & \ddots & \ddots & & \\ & & & & & A_0 & A_1 & \hat{A}_{K-1} & \\ & & & & & & A_0 & \hat{A}_K & \end{bmatrix} \tag{3}$$

with $C_0 = \begin{bmatrix} D_0 \otimes (L + \ell\boldsymbol{\delta}) \end{bmatrix}$, $E = \begin{bmatrix} \mathbf{0} \\ \mathbf{e} \otimes D_0 \otimes \boldsymbol{\delta} \end{bmatrix}$,

$$C_j = \begin{bmatrix} D_j \otimes L & \mathbf{e}_1^{'} \otimes D_j \otimes \ell \end{bmatrix}, \ j = 1, 2, \cdots, \eta$$

$$A_0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{e}_T \otimes D_0 \otimes \boldsymbol{\delta} & \bar{I}(T-1) \otimes D_0 \end{bmatrix},$$

$$A_j = \begin{bmatrix} D_{j-1} \otimes L & \mathbf{e}_1^{'} \otimes D_{j-1} \otimes \ell \\ \mathbf{e}_T \otimes D_j \otimes \boldsymbol{\delta} & \bar{I}(T-1) \otimes D_j \end{bmatrix}, \quad j = 1, 2, \cdots, \eta$$

$$A_{\eta+1} = \begin{bmatrix} D_\eta \otimes L & \mathbf{e}_1^{'} \otimes D_\eta \otimes \ell \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\hat{A}_i = \sum_{j=K-i+1}^{\eta} A_j + A_{\eta+1}.$$

Here, $\otimes$ is the Kronecker product of two matrices, $\ell = \mathbf{1} - L\mathbf{1}$ is the vector probability denoting the end of the vacation period, and $\mathbf{e}_x$ is a column vector of zeros with a single entry of 1 in position x. In this paper, our attention will be restricted to the case where $K \geq T$ and $K > \eta$. The case of $K < T$ and/or $K \leq \eta$ can be developed in a similar manner.

In (3), the block matrix C_0 represents the transition probability for the case where the system remains empty due to no arrivals and the server is still on vacation, or returns from vacation to find the system empty and starts another vacation. The block matrices C_j represents the case where j arrivals have occurred while the server is still on vacation (since the system was empty) as given by the left-most element, or the vacation has ended and the server starts serving the first arriving packet as given by the right-most element.

The block A_0 describes the single departure from the system which can only occur if the system is in a service period (i.e. not in vacation mode), and there are no arrivals. The lower left-most element in A_0 is the probability that the server goes on vacation after the departure of the packet at the end of the buffer's polling period, while the lower right-most element is the probability that the server starts processing the next packet in the non-empty queue after the departure of the served packet. Note that for the upper elements in matrix A_0 which represent the state when the system is in vacation mode, the transition probabilities are zero since it is not possible for the server to be on vacation while having served any packets.

The blocks A_j describe the transitions involving the arrivals of $j-1$ packets into the system if it is in vacation mode as given by the upper elements in the matrix, or the arrival of j packets during a server visit period with a service completion and departure of a single packet from the head of the buffer.

The remaining blocks can be described in a similar manner.

D. System Steady-State Distribution

Let $\boldsymbol{x}$ be the steady-state distribution of the system with the transition probability matrix given by $\mathbf{P}$, such that $\boldsymbol{x} = \boldsymbol{x}P$, and $\boldsymbol{x}\mathbf{1} = 1$. Then we can obtain $\boldsymbol{x}$ by using standard linear algebra techniques, where $\boldsymbol{x} = [\boldsymbol{x}_0^v, \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_K]$ and $\boldsymbol{x}_i = [\boldsymbol{x}_i^v, \boldsymbol{x}_i^s]$ for $1 \leq i \leq K$, with the vectors

$$\begin{aligned} \boldsymbol{x}_i^v &= \left\{\boldsymbol{x}_{i,j,k}^v\right\} \quad \text{for} \quad 1 \leq j \leq n \quad 1 \leq k \leq \theta, \\ \boldsymbol{x}_i^s &= \left\{\boldsymbol{x}_{i,j,t}^s\right\} \quad \text{for} \quad 1 \leq j \leq n \quad 1 \leq t \leq T. \end{aligned}$$

$\boldsymbol{x}_i^v$ is the vector probability of the system being in the vacation period with i packets in the system, and $\boldsymbol{x}_i^s$ is the vector probability of the system being in the service period with i packets in the system. These steady-state probabilities are next used to derive our performance measures of interest.

IV. Performance Measures

A. Blocking Probability

The blocking probability can be computed as the ratio of the average number of blocked packets due to buffer overflow to the average number of packet arrivals. Hence, the blocking probability P_b is given as

$$\begin{aligned} P_b &= \frac{\text{Average Number of Blocked Packets}}{\text{Average Number of Arrivals}} \qquad (4) \\ &= \frac{1}{\lambda}\left[\sum_{i=K-\eta+1}^{K} \sum_{k=K-i+1}^{\eta} (k-(K-i))\boldsymbol{x}_i^v \left(I_n \otimes \mathbf{1}_\theta\right) D_k \right. \\ &\quad \left. + \sum_{i=K-\eta+2}^{K} \sum_{k=K-i+2}^{\eta} \sum_{t=1}^{T} (k-(K-i)-1)\boldsymbol{x}_{i,t}^s D_k \right] \mathbf{1}_n \end{aligned}$$

where I_x is an identity matrix of order x, and $\mathbf{1}_x$ is a column vector of all ones with length x.

B. Mean Occupancy & Residence Time

Knowing the steady-state probability distribution $\boldsymbol{x}$ of the system, the mean number of packets in the system, μ_L, can be computed as

$$\mu_L = \sum_{i=1}^{K} i\left(\boldsymbol{x}_i^v \mathbf{1}_{n\theta} + \boldsymbol{x}_i^s \mathbf{1}_{nT}\right) = \sum_{i=1}^{K} i\boldsymbol{x}_i \mathbf{1}_{n(\theta+T)}. \qquad (5)$$

Given that the mean packet arrival rate into the buffer is $\lambda = \sum_{x=1}^{\eta} x\boldsymbol{\pi} D_x \mathbf{1}$, and using Little's Law we can compute the mean residence time, W_L, as

$$W_L = \frac{\mu_L}{\lambda^*} \qquad (6)$$

where the system's throughput $\lambda^* = \lambda\left(1 - P_b\right)$.

C. Waiting-Time Distribution

In order to obtain the waiting time distribution, we need to first compute the equivalent distribution of the buffer occupancy size as seen by an arriving packet that gets served. Let $\mathbf{y}$ be the distribution of the buffer occupancy as seen by the arriving packet that is accepted into the system. Define $\mathbf{y} = [\mathbf{y_0}, \mathbf{y_1}, \cdots, \mathbf{y_{K-1}}]$, and $\mathbf{y_i} = [\mathbf{y_i^v}, \mathbf{y_i^s}]$ for $1 \leq i \leq K-1$, with $\mathbf{y_0} = \mathbf{y_0^v}$ and $\mathbf{y_i^s} = \left[\mathbf{y_{i,1}^s}, \mathbf{y_{i,2}^s}, \cdots, \mathbf{y_{i,t}^s}, \cdots, \mathbf{y_{i,T}^s}\right]$, where t is the time phase of service. $\mathbf{y_i^v}$ is the probability of an arrival finding i packets waiting in the system when the server is on vacation. The derivation of $\mathbf{y_i^v}$ are given as follows.

$$\begin{aligned} \mathbf{y_0^v} &= \left(\lambda^*\right)^{-1}\left[\sum_{k=1}^{\eta} x_0^v D_k \otimes (L + \ell\boldsymbol{\delta}) \right. \\ &\quad \left. + \sum_{t=1}^{T} \sum_{k=1}^{\eta} x_{1,t}^s D_k \otimes \boldsymbol{\delta}\right] \left(\mathbf{1}_n \otimes I_\theta\right). \qquad (7) \end{aligned}$$

For $1 \le i \le \eta - 1$,

$$\mathbf{y_i^v} = (\lambda^*)^{-1}\left[\sum_{k=i+1}^{\eta} x_0^v D_k \otimes (L + \ell\boldsymbol{\delta}) + \sum_{j=1}^{i}\sum_{k=i-j+1}^{\eta} x_j^v D_k \otimes L + \sum_{j=2}^{i+1}\sum_{k=i-j+2}^{\eta} x_{j,T}^s D_k \otimes \boldsymbol{\delta} + \sum_{t=1}^{T}\sum_{k=i+1}^{\eta} x_{1,t}^s D_k \otimes \boldsymbol{\delta}\right](\mathbf{1}_n \otimes I_\theta). \quad (8)$$

For $\eta \le i \le K - 1$,

$$\mathbf{y_i^v} = (\lambda^*)^{-1}\left[\sum_{j=i-(\eta-1)}^{i}\sum_{k=i-j+1}^{\eta} x_j^v D_k \otimes L + \sum_{j=i-(\eta-2)}^{i+1}\sum_{k=i-j+2}^{\eta} x_{j,T}^s D_k \otimes \boldsymbol{\delta}\right](\mathbf{1}_n \otimes I_\theta). \quad (9)$$

$\mathbf{y_0^v}$ is the probability that an arriving packet is accepted into the system while it is in vacation mode with no other packets ahead of it to be served. This is equivalent to the probability of finding the system empty and in vacation mode in the previous time slot with the packet arriving at the head of the batch arrivals (as given by the first summation), or the probability that the system processed the last packet in the queue at the previous time slot and forcing the server to go on vacation with the packet arriving at the head of the batch arrivals in the current time slot (as given by the second summation). The remaining $\mathbf{y_i}$ elements can be interpreted in a similar manner and is omitted for brevity.

$\mathbf{y_i^s}$ is the probability of finding i packets in the system when the server is in service and is derived as follows. For $1 \le i \le K - 1$,

$$\mathbf{y_{i,1}^s} = (\lambda^*)^{-1}\left[\sum_{j=\max\{1,i-(\eta-1)\}}^{i}\sum_{k=i-j+1}^{\eta} x_j^v D_k \otimes \ell\right]\mathbf{1}_n. \quad (10)$$

For $2 \le t \le T$,

$$\mathbf{y_{i,t}^s} = (\lambda^*)^{-1}\left[\sum_{j=\max\{2,i-(\eta-2)\}}^{i+1}\sum_{k=i-j+2}^{\eta} x_{j,t-1}^s D_k\right]\mathbf{1}_n. \quad (11)$$

It can be shown that λ^* is simply the sum of all the terms inside the main brackets of the elements $\mathbf{y_i}$, $\forall i$, which is equivalent to the system's throughput derived in Section IV-B.

Let the waiting time be W and $w_j = Pr\{W = j\}$, where w_j is the probability that the arbitrary arriving packet has to wait j units of time in the system *before* it starts receiving service, given that the packet arrived *successfully* into the system (i.e. not blocked). Define a transition matrix P_w which captures the transient state of the system as seen by an arbitrary arriving packet and ignores the subsequent arrivals (assuming a FIFO system). This is equivalent to analyzing the time for emptying the buffer of all packets ahead of the arbitrary arriving packet that is accepted into the system. P_w can be easily obtained from $\mathbf{P}$ in (3) by setting $D_x = 0$ for $x > 0$, and $D_0 = 1$ (i.e. ignoring all future arrivals), with the exception of the lower boundary conditions $\tilde{B}$ and $\tilde{E}$. Hence, the matrix P_w can be computed as follows.

$$P_w = \begin{bmatrix} \tilde{B} & & & & \\ \tilde{E} & \tilde{A}_1 & & & \\ & \tilde{A}_2 & \tilde{A}_1 & & \\ & & \tilde{A}_2 & \tilde{A}_1 & \\ & & & \ddots & \ddots \\ & & & & \tilde{A}_2 \quad \tilde{A}_1 \end{bmatrix}, \quad \text{and} \quad \Omega_0 = \mathbf{1} - P_w\mathbf{1} \quad (12)$$

where $\tilde{B} = L, \quad \tilde{E} = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_T \otimes \boldsymbol{\delta} \end{bmatrix},$

$$\tilde{A}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{e}_T \otimes \boldsymbol{\delta} & \bar{I}(T-1) \end{bmatrix}, \quad \tilde{A}_1 = \begin{bmatrix} L & \mathbf{e}_1^{'} \otimes \ell \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Given $\mathbf{y}$ and P_w, the waiting time distribution w_j can be studied as a phase type distribution $(\mathbf{y}, P_w)$. Therefore,

$$w_j = Pr\{W = j\} = \mathbf{y}P_w^{j-1}\Omega_0, \quad \text{for} \quad j \ge 1. \quad (13)$$

Using our expression for w_j, we can further compute the cumulative waiting time distribution. Moreover, the mean waiting time $E[W]$ can be computed as

$$E[W] = \mathbf{y}\,(I - P_w)^{-1}\,\mathbf{1}. \quad (14)$$

Furthermore, we know that each packet requires one unit of time for service, which is deterministic. Hence, the mean residence time in the system for an arbitrary packet is

$$W_L = E[W] + 1 = \mathbf{y}\,(I - P_w)^{-1}\,\mathbf{1} + 1 \quad (15)$$

which is equal to what was computed in Equation (6). Note that the unit of time used in this model is equivalent to the time taken to serve a single packet.

Our solution for the waiting time distribution can be used to obtain higher moments than the first one (i.e. mean), which can help to provide a better understanding of the system's performance in terms of the waiting time. This is particularly useful when considering delay sensitive applications.

V. Numerical Example

In this section, a simple example is presented to both verify our modeling analysis, and illustrate the behavior of a simple polling system applied for the case of a single class of PS traffic in IEEE 802.16. The analysis was carried out using the performance measures given in the previous section. A simulation program, written in Matlab, was used to verify the results obtained from our analysis.

For the purpose of simplicity, we will only consider three subscriber stations, each with its own buffer for holding the arriving PS traffic. In our example, the PS buffers in each SS will have a size of $K = 25$ and the BMAP arrival process into all the buffers will be identical. This may not be a realistic situation but we chose to analyze this type of configuration due to the symmetry in the performance which can serve as

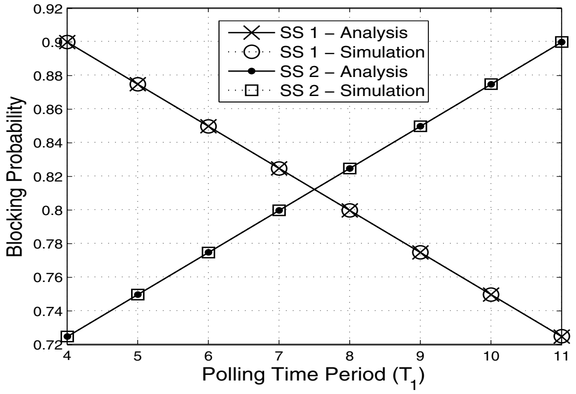

Fig. 2. Blocking Probability in SS 1 and SS 2, with varying T_1 and T_2.

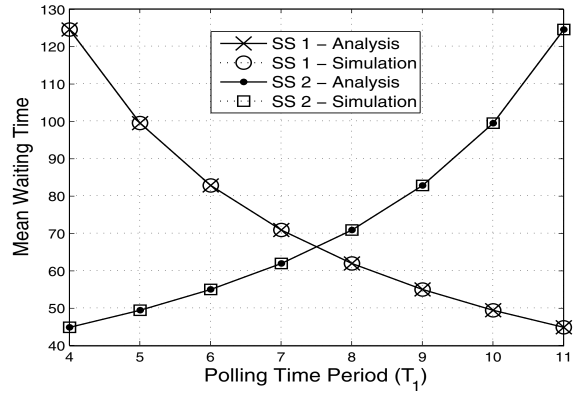

Fig. 3. Mean Waiting Time in SS 1 and SS 2, with varying T_1 and T_2.

an added check to our analysis. Similar results can be obtained for the case of heterogeneous traffic into the PS buffers.

In our simulation, the three polling stations were collectively considered and cyclically processed, such that the exact state of each of the stations are known at all times. Hence, there was no need to make any assumptions in the simulation on the server vacation periods for each of the stations. This was also useful in verifying our assumption for the server vacation distribution under heavy traffic load conditions, as mentioned in Section III-B. A single run of the simulation with a run-time of 4 million time slots was made for each scenario, which was found to be sufficient for ensuring that the system reached a steady-state behavior.

For our BMAP arrival process, we considered the case where the maximum batch size $\eta = 4$ with

$$D_0 = \begin{pmatrix} 0.1 & 0.05 \\ 0.1 & 0.2 \end{pmatrix}, \ D_1 = \begin{pmatrix} 0.05 & 0.1 \\ 0.2 & 0.1 \end{pmatrix}, \ D_2 = \begin{pmatrix} 0.05 & 0.05 \\ 0.05 & 0.05 \end{pmatrix},$$

$$D_3 = \begin{pmatrix} 0.2 & 0.1 \\ 0.05 & 0.1 \end{pmatrix}, \ \ D_4 = \begin{pmatrix} 0.1 & 0.2 \\ 0.1 & 0.05 \end{pmatrix}.$$

With this BMAP, the mean arrival rate into each of the buffers is $\lambda = 2$ packets per time slot.

In the analysis and simulation, the single polling period was fixed at $T_1 + T_2 + T_3 = 20$. In order to evaluate the effects of T_i on the system's performance, we fixed $T_3 = 5$ and varied T_1 and T_2 with the constraint $T_1 + T_2 = 15$. Note that a single packet is served during each time slot. Given our system parameters, the buffers are under heavy traffic load and thus the server vacation period for each buffer is approximately the sum of the polling time-limit for the other two buffers.

Figures 2 and 3 show both the blocking probability and the mean waiting time of an arriving packet into SS 1 and SS 2 with T_1 varying from 4 to 11, respectively. Note that T_2 also varies with T_1 since we fix $T_1 + T_2 = 15$. The graphs clearly show how our simulation verifies our results from the analysis. The results suggest that a linear increase in T_1 (with a simultaneous linear decrease in T_2) causes a linear decrease in the blocking probability as shown in Figure 2, which simultaneously decreases the mean waiting time $E[W]$ at a faster rate. In addition, note that due to the symmetry of our parameters in the two systems, the performance for the SS 2 with T_2 is the direct opposite of that for SS 1.

Focusing only on the case where $T_1 = 6$ and $T_2 = 9$, while keeping all other system parameters the same as before (with $T_3 = 5$), Figures 4 to 7 show our results for the waiting time distribution and the cumulative distribution for both SS 1 and SS 2, respectively. These graphs also show how our simulation verifies our results from the analysis. For SS 1, note how the peak of 80 time units in Figure 4 is close to the mean for $T_1 = 6$ in Figure 3. This peak of 80 time units corresponds to the waiting time that a packet may experience in SS 1 as a result of 4 server vacation periods with a length $T_2 + T_3$ each and the time taken to serve a buffer with $K - 1$ packets, with the waiting packet being the last packet in the buffer. The lower peak of 94 time units is the result of an extra server vacation period to the higher peak of 80 units. This suggests that a packet waiting in SS 1 is more likely to witness 4 vacation periods as opposed to 5. Note that for this example, 5 server vacation periods (i.e. $\lceil K/T_1 \rceil$) is the maximum for a packet that is waiting in SS 1. The peaks in Figure 6 can be interpreted in a similar manner, whereby the lower peak is the result of the final packet in the queue waiting for two vacation periods (T_1 + T_3) and the service of $K - 1$ packets before being served, and the higher peak corresponds to the packet waiting an extra vacation period.

A second look at the results reveal that an arriving packet in SS 2 has a higher likelihood of experiencing the maximum number of server vacations when compared to SS 1. The difference is attributed to the size of the service periods of the two stations since it is the only factor that differentiates both stations. However, the waiting time in SS 2 is much less compared to SS 1. This behavior could provide some insight on how to select the polling times for each station and is currently being investigated.

VI. Conclusions & Future Work

In this paper, we presented a queueing model for analyzing the performance of the polling service mechanism adopted in IEEE 802.16 networks. The model considers the correlation of the vacation and service periods amongst the PS buffers in the

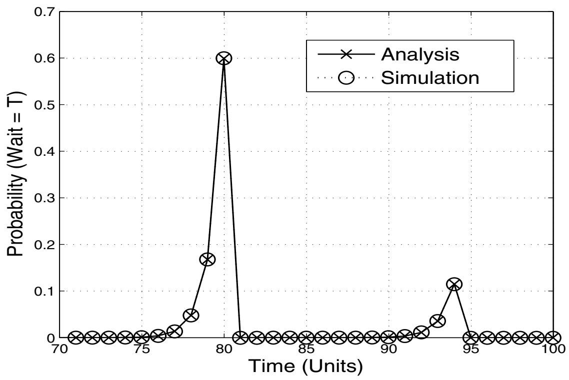

Fig. 4. Waiting Time Distribution in SS 1, with $T_1 = 6$ and $K = 25$.

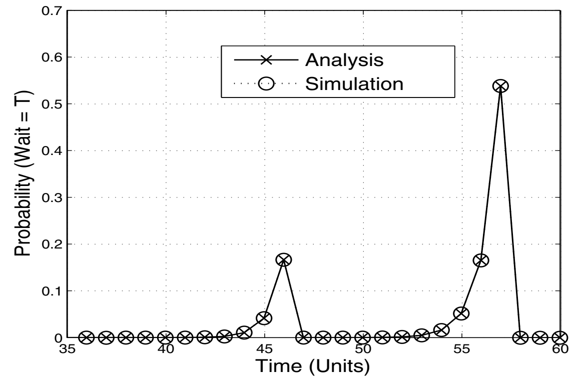

Fig. 6. Waiting Time Distribution in SS 2, with $T_2 = 9$ and $K = 25$.

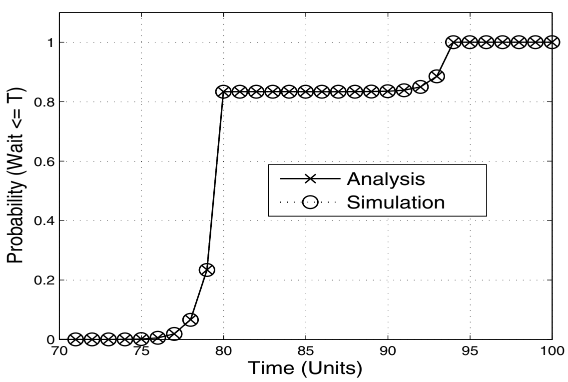

Fig. 5. Cumulative Distribution of Waiting Time in SS 1, with $T_1 = 6$ and $K = 25$.

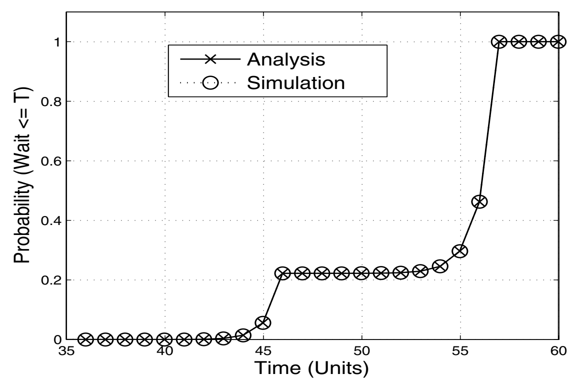

Fig. 7. Cumulative Distribution of Waiting Time in SS 2, with $T_2 = 9$ and $K = 25$.

system. Our proposed vacation period approximation helped in capturing the correlation between the SSs under heavy traffic load conditions. The model can be utilized for assigning a suitable polling service time-limit T for each PS buffer with different traffic loads, while adhering to some QoS constraints, based on the blocking probabilities, mean queue lengths and waiting time.

Our model assumes that the network resources available to the UGS and PS traffic are completely partitioned amongst them, which allowed for an independent analysis of PS traffic. Other scenarios call for the resources to be shared between these types of traffic, which implies that the performance of UGS traffic will have an influence on the PS traffic. This influence will be considered in our future work. Moreover, under heavy traffic conditions, the correlation between the SSs play a minor role and hence our proposed server vacation period is a good approximation. The case of lighter traffic loads will also be investigated.

REFERENCES

[1] Carl Eklund, Roger Marks, Kenneth Stanwood, and Stanely Wang, “IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access”, *IEEE Commun. Mag.*, June 2002, pp. 98-107.

[2] Claudio Cicconetti, Alessandro Erta, Luciano Lenzini, and Enzo Mingozzi, “Performance Evaluation of the IEEE 802.16 MAC for QoS Support”, *IEEE Trans. Mobile Computing*, vol. 6, no. 1, January 2007.

[3] Dongmei Zhao and Xuemin Shen, “Performance Of Packet Voice Transmission Using IEEE 802.16 Protocol”, *IEEE Wireless Comm.*, February 2007.

[4] Kitti Wongthavarawat and Aura Ganz, “Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems”, *International Journal of Communication Systems*, vol. 16, 2003, pp. 81-96.

[5] Alexander Sayenko, Olli Alanen, and Timo Hämäläinen, “Scheduling Solution for the IEEE 802.16 Base Station”, *Computer Networks*, 2007.

[6] Jenhui Chen and Chiang-Wei Chang, “Traffic-Variation-Aware Connection Admission Control Mechanism for Polling Services in IEEE 802.16 Systems”, *IEEE WOCN*, July 2007.

[7] Jun-Bae Seo et al., “An Efficient Capacity Allocation Scheme of Periodic Polling Services for a Multimedia Traffic in an IEEE 802.16 System”, *IEEE MOBHOC*, October 2006.

[8] Dusit Niyato and Ekram Hossain, “Queue-Aware Uplink Bandwidth Allocation and Rate Control for Polling Service in IEEE 802.16 Broadband Wireless Networks”, *IEEE Trans Mobile Computing*, vol. 5, no. 6, June 2006, pp. 668-679

[9] Attahiru S. Alfa, “Vacation Models in Discrete Time”, *Queueing Systems*, Vol. 44, 2003, pp. 5-30.

[10] Chris Blondia, “A Discrete-Time Batch Markovian Arrival Process as B-ISDN Traffic Model”, *Belgian Journal of Operations Research*, vol. 32, 3-23, 1992.

[11] Duan-Shin Lee and Bhaskar Sengupta, “An Approximate Analysis of a Cyclic Server Queue with Limited Service Reservations”, Queueing Systems, vol. 11, 1992, pp. 153-178.

[12] Imed Frigui and Attahiru S. Alfa, “Analysis of a Time Limited Polling System”, Computer Communications, Vol. 21, 1998, pp. 558-571.