# Analysis of a Polling System Modeling QoS Differentiation in WLANs

T.J.M. Coenen*
University of Twente
Department of Applied Mathematics
Enschede, The Netherlands
t.j.m.coenen@utwente.nl

J.L. van den Berg†
TNO Information and
Communication Technology
Delft, The Netherlands
j.l.vandenberg@tno.nl

Richard J. Boucherie
University of Twente
Department of Applied Mathematics
Enschede, The Netherlands
r.j.boucherie@utwente.nl

## ABSTRACT

This paper investigates a polling system with a random polling scheme, a 1-limited service discipline and deterministic service requirement modeling WLANs with QoS differentation capability. The system contains high and low priority queues that are distinguished via the probability of being served next. We propose a new iteration algorithm to approximate the waiting time of customers in the high and low priority queues. As shown by simulation results, our approximation is accurate for light to moderately loaded networks.

**Keywords**: Polling model, QoS differentiation, WLAN, IEEE 802.11e

**AMS subject classification**: 90B15, 90B18, 90B22, 68M20, 60K25

## 1. INTRODUCTION

Wireless Local Area Networks (WLANs) have become widely available for internet access and there is currently a growing demand for the support of other applications, in particular speech and video. Specific mechanisms then need to be deployed in order to provide appropriate QoS to the various applications. A typical approach to provide such QoS differentiation is for example by giving a larger share of the available capacity to preferred users, or giving priority to preferred classes. Introduction of such mechanisms requires insight into their performance. This paper investigates the influence of prioritization of the packet delay handling at the Medium Access Control (MAC) layer in WLANs.

In IEEE 802.11 WLAN prioritization appears in the support of different QoS classes. These QoS classes are implemented via different settings of MAC layer parameters, like their access time, the maximum and minimum value for their back-off counter or the number of consecutive packets that may be transmitted, see [10] for an overview of IEEE 802.11e that incorporates these mechanisms. QoS provisioning for IEEE 802.11 systems has been investigated mainly via discrete event simulations. Analytical models yielding robust insight into system behaviour are scarce. To a large extent, such models are based on the pioneering work of Bianchi [1], in which a basic 802.11 system with persistent sources, i.e. sources that always have packets ready to be transmitted, is modeled and analysed using a Markov chain approach and validated via simulation showing excellent agreement with actual system behaviour. Extensions to include physical layer details are given in e.g. [9],[16]. The extension to non-persistent sources is provided in [3],[14], where a flow level model is introduced that is analysed using a Processor Sharing queueing model. Comparison with discrete event simulation shows that indeed the MAC layer can be adequately modeled via the Processor Sharing mechanism. Extensions to multiple traffic classes with different QoS requirements, as e.g. in 802.11e, are among others presented in [17],[18],[19].

Although the flow level modeling of [3],[14],[17],[18],[19] captures the resource sharing behaviour of the MAC layer of 802.11 protocols, the essential behaviour at the packet level is not captured. At that level a flow consists of a series of packets that are transmitted one by one, where transmissions of different flows are intertwined. Especially for real time applications, such as speech/telephony, the packet level is of high importance. In [4], a packet level analysis for non-persistent sources is presented, extending the Markov model of Bianchi to include the probability of the node going into an empty backoff state. We take a further step to analyze the packet level by modeling the MAC layer as a polling model where the server works off packets at different queues. The essential characteristics of the QoS aware MAC protocol are incorporated via the frequency at which the server visits the different nodes. In particular, we give the server a high probability of visiting a node with high priority packets.

In our polling model, we consider two types of queues, viz. high and low priority queues, each type with a different probability of the server moving to it. Upon departure from a queue, the server randomly selects a queue according to these probabilities, which mimics the behaviour of the MAC layer in 802.11 systems. Note that we do not claim to accurately model the behaviour of the IEEE 802.11e protocol, but analyse a mathematically interesting model that provides insight into the effect of prioritization such as used in

*Corresponding author: Tom Coenen, University of Twente, Postbox 217, 7500 AE, Enschede, The Netherlands, t.j.m.coenen@utwente.nl

†Also affiliated with: University of Twente, Department of Computer Science, Enschede, The Netherlands


ValueTools 2008, October 21 – 23, 2008, Athens, GREECE.

the IEEE 802.11 MAC layer. In our model, we will take the probability of moving to a high priority (HP) queue to be $\alpha$ times as high as moving to a low priority (LP) queue. The service time of a packet is considered to be deterministic as the packet sizes in the system are equal for all queues and the channel speed is assumed to be constant at all times. As a queue is only allowed to transmit one packet when obtaining the channel, the service discipline is 1-limited. This paper analyzes the steady state waiting time for this 1-limited polling system with random polling.

For the 1-limited polling model, general results are available in literature. In [6] Fuhrmann and Cooper derive the well known decomposition result for queues with server vacations, which is very useful for analyzing polling models. For symmetric queues, so with identical arrival and service rates at the queue, and a cyclic polling order, [7] extends this result to give analytical results on the average waiting time of packets in the queues. In [2], Boxma gives a pseudoconservation law for the mean waiting time in a polling system with Markovian polling, that includes random polling. This law provides an exact expression for a weighted sum of the mean waiting times at all queues, which need not be symmetrical. However, results for individual queues cannot be derived from this law when the network is not symmetric.

The main contribution of this paper is an analysis of the steady state marginal distribution of the waiting time of packets for different types of queues in a 1-limited asymmetric polling model. We consider the different queues in the system individually and model a particular queue as a queue with server vacations, where these vacations depend on the state of the other queues. To obtain the steady state waiting time distribution, we propose an iteration algorithm. The algorithm computes the marginal steady state distribution of the number of packets at a tagged queue, assuming a steady state at all other queues. Iterating this approach over the queues, for various settings, we obtain the steady state waiting time distribution for packets at the different queues.

The remainder of this paper is organised as follows. Section 2 describes the queueing networks under consideration and the analytical approach for determining the distribution of the waiting time of customers per queue. Numerical results of the proposed algorithm are compared with simulation in Section 3, and Section 4 concludes the paper.

# 2. MODEL DESCRIPTION AND ANALYSIS

Consider a polling model consisting of queues $Q_1, ..., Q_n$ with finite buffer $B$ and a single server $S$ visiting the queues. Customers arrive at a queue $Q_i$ according to a Poisson process with rate $\lambda_i$. The service process at the queues is deterministic with service time $\tau$ and there is no switchover time between the queues. The routing policy for the server is random, meaning there is a probability $p_i$ that the server moves to queue $Q_i$ upon departure from queue $Q_j$, $j = 1, ..., n$. For a high priority queue, this probability is $\alpha$ times as high as for a low priority queue, that is $p_{HP} = \alpha p_{LP}$. The service policy is assumed to be 1-limited, meaning at most one customer is served at each visit of the server, and customers are served FCFS at each queue. When the server reaches an empty queue, it will immediately proceed to the next. When all queues are empty, the server waits at the last queue to instantly move to the first queue that receives a customer. To ensure stability of the system we assume that $\rho = \sum_{i=1}^{n} \lambda_i \tau < 1$.

In the following, we derive expressions for the average waiting time of a packet for both types of queue. We start by considering one high priority queue surrounded by $n$ low priority queues. At both queues, packets arrive according to a Poisson process. The server will move to the HP queue with probability $\frac{\alpha}{n+\alpha}$ and to a certain LP queue with probability $\frac{1}{n+\alpha}$. We present an algorithm to approximate the waiting time of a packet for both types of queue. This algorithm considers queues separately as served by a server with vacations. The length of the vacations depends on the number of customers at the other queues. Starting with an arbitrary distribution of the number of customers at the other queues, the steady state of the number of customers in the considered queue is determined, using the vacation time distribution. This process is iterated over the different types of queues repeatedly, until convergence occurs. For specific cases, being that either the HP or LP queues are saturated, meaning they always have packets ready to be transmitted, exact results are presented. Exact results are also given for the case where all queues have equal priority.

## 2.1 General case

To determine the average waiting time of a packet in the queue, we consider the queues separately, as if they are in isolation. From the point of view of a queue, the server is either present and serving a packet, or away while serving an other queue. We thus can consider a queue as an M/D/1/B queue with vacations (c.f. [5],[11],[12]), where the absence of the server while serving other queues are the vacations. The length of these vacations, which depends on the number of customers at the other queues, influences the waiting time of the packets in the queue. For illustratory reasons, we first give the analysis for the scenario where there are two queues, one high priority and one low priority queue, which as we show later can be extended to any number of queues.

### 2.1.1 Two queues

In the two queue scenario, each queue can be considered separately as a queue with a server that goes on vacation. The duration of a vacation now depends on the state of the other queue. We approximate the distribution of the length of the vacation $V_x$, given the number of customers $N_y$ at the other queue (HP or LP) using the following recursion:

$$P(V_x = k\tau | N_y = i) = \tag{1}$$

$$q_y \sum_{j=i-1}^{B} P(V_x = (k-1)\tau | N_y = j) P(A_y = j - i + 1), \forall k \geq 1$$

$$P(V_x = 0 | N_y = i) = \begin{cases} 1, & i = 0 \\ (1 - q_y), & i = 1, ..., B \end{cases}$$

where $V_x$ is the length of the vacation seen by the queue $x$, $N_y$, $q_y$ and $A_y$ are the number of customers at queue $y$, the probability of the server polling queue $y$ and the number of arriving customers at queue $y$ during the service time at queue $x$, respectively. Note that length of a service period is known to be $\tau$ due to the 1-limited service discipline, hence we will denote this as a service time. The variable $x$ can be the $HP$ or $LP$ queue and $y$ is the other type of queue. The

vacation length distribution is then determined using

$$P(V_x = k\tau) = \sum_{i=0}^{B} P(V_x = k\tau | N_y = i) P(N_y = i),\ k \geq 0 \tag{2}$$

As the steady state distribution $P(N_y = i)$, is not known, we start with an arbitrary distribution, for example an always empty queue. Using this distribution, the vacation distribution for the other queue is obtained.

We derive the steady state distribution of the number of customers in the queue using the vacation time distribution, so that by using Little's law we acquire the expected waiting time of a packet. The queue under consideration can be seen as an M/D/1/B queue with vacations (c.f. [5],[11],[12]). To analyze the steady state of this queue, we first focus on the state of the system at embedded points, which are after the departure of a customer or the end of a vacation. The probability $p_n$ that an embedded point is the completion of a service and the departing customer leaves $n$ customers behind, and the probability $q_n$ that an embedded point is a vacation termination with $n$ customers in the system are related in the following manner

$$\begin{aligned}
p_n &= \sum_{k=1}^{n+1} g_{n-k+1} q_k,\ n = 0, 1, .., B-2 \\
p_{B-1} &= \sum_{k=1}^{B} g_{B-k}^{C} q_k \\
q_n &= \sum_{k=0}^{n} h_{n-k} p_k + h_n q_0,\ n = 0, 1, .., B-1 \\
q_B &= \sum_{k=0}^{B-1} h_{B-k}^{C} p_k + h_B^{C} q_0 \\
\sum_{n=0}^{B-1} p_n &+ \sum_{n=0}^{B} q_n = 1
\end{aligned}$$

where $g_j$ and $h_j$ denote the probability of $j$ customers arriving during a service and vacation time, respectively, $g_j^C$ and $h_j^C$ denote the probability of $j$ or more customers arriving. As these probabilities are known, this set of equations can be solved, giving the steady state distribution at the end of an interval (either a service or vacation). To determine the continuous time steady state distribution, we note that the number of times a departing customer leaves a certain number of customers behind equals the number of times an arriving customer finds this number of customers in the system. We have to take into account, however, that an arriving customer can find $B$ customers in the system in which case the customer is discarded and leaves. Let $P_B$ denote the probability that an arriving customer finds the system full. To evaluate this expression, observe that

$$P_B = \frac{\rho - \rho'}{\rho}$$

where $\rho = \lambda\tau$, $\lambda = \sum_i \lambda_i$ is the offered load and $\rho'$ is the carried load,

$$\rho' = \frac{(1-b)\tau}{bEV + (1-b)\tau}$$

where $EV$ denotes the expected vacation time and $b$ denotes the probability that an embedded point is a vacation termination point,

$$b = \sum_{n=0}^{B} q_n.$$

Let $\sigma$ denote the multiplicative inverse of the average interval between consecutive embedded points, that is

$$\sigma^{-1} = bEV + (1-b)\tau$$

then

$$P_B = 1 - \frac{(1-b)\sigma}{\lambda}.$$

The queue length distribution at arrival epochs, $\pi_n$, $n = 0, ..., B$ is

$$\begin{aligned}
\pi_n &= P(\text{Arrival sees } n \text{ packets} | \text{Arrival is accepted})(1 - P_B) \\
&\quad + P(\text{Arrival sees } n \text{ packets} | \text{Arrival not accepted}) P_B \\
&= p_n(1 - P_B) + P_B 1(n = B)
\end{aligned}$$

where

$$1(n = B) = \begin{cases} 1 \text{ if } n = B \\ 0 \text{ otherwise} \end{cases}$$

Combining these results, we obtain

$$\begin{aligned}
\pi_n &= \frac{(1-b)\sigma}{\lambda} p_n,\ n = 0, 1, ..., B-1 \\
\pi_B &= 1 - \frac{(1-b)\sigma}{\lambda}.
\end{aligned} \tag{3}$$

From PASTA we obtain that the continuous time steady state queue length distribution is given by $\pi_n$, $n = 0, ..., B$. Note that (3) requires the average vacation time $EV$, and (2) the distribution of the other queue to determine the vacation time distribution. We may iterate (2) and (3) to obtain an approximation of the steady state queue length distribution.

ALGORITHM 1. *Iteration*

1. Initialize
   $It := 1$, $x := 1$, $y := 2$
   $P(N_y = i) = \gamma_i$, $EN_i(0) = 0$ for $i = 0, ..., B$
   where $EN_i(j)$ denotes the average queue length of queue $i$ in iteration $j$

2. Determine the vacation time distribution at queue $x$ from (2), and $EV := EV_x$

3. Determine the queue length distribution $P(N_x = n) = \pi_n$, $n = 0, ..., B$, from (3) and determine the average queue length $EN_x(It)$

4. Set $y := x$, $x := 3 - y$ and repeat steps 2 and 3 for this setting

5. If $\frac{EN_x(It) - EN_x(It-1)}{EN_x(It)} < 0.01$ for both $x = 1, 2$, then STOP
   Else $y := x$, $x := 3 - y$, $It := It + 1$ Go to Step 2

The algorithm approximates in each iteration the number of customers found at the other queue to determine the vacation time for the tagged queue. When this vacation time is underestimated, the server switches back early

to the queue and starts servicing a packet at the considered queue (when available), thus leaving the server busy. When however the vacation time is overestimated, the approach leaves the server at the other queue for too long a period, where this queue might actually have become empty, thus leaving the server idle while it could process jobs in the tagged (non-empty) queue. The presented approach hence underestimates the capacity of the server, but equally for both queues. The average queue length of all customers in the total system, which for larger values of $B$ approximately can be seen as an M/D/1 queue as it is work conserving, is known and given by

$$EN_{total} = \frac{\rho(2-\rho)}{2(1-\rho)}$$

where $\rho = (\lambda_{LP} + \lambda_{HP})\tau$, the load of the total system. The results obtained by the iteration give a higher average queue length due to underestimation of the server capacity. The queue length of each type of customer should hence be scaled down, so that the average queue length of all customers in the system is correct. This leads to an improved estimation of the average queue length of a customer per type of queue. Using Little's law, we obtain the average waiting time for each type of queue.

The algorithm can start with an arbitrarily chosen steady state distribution for the queue length of the HP queue. From this, a new steady state is computed for the same queue. Starting from each initial distribution for the HP queue, Algorithm 1 converges to the steady state distribution. Theorem 1 below states that this convergence is monotone starting from either an empty or full HP queue. We need stochastic ordering. Let $X$ and $Y$ be random variables with distribution $F_X(.)$ and $F_Y(.)$, respectively. We say that $X \leq_{st} Y$ iff $F_X(x) \geq F_Y(x)$ for all $x \geq 0$ (c.f. [15], p.410).

THEOREM 1. *For each initial distribution, Algorithm 1 converges monotonically.*

PROOF. Let $X_i^{HP}$ and $X_i^{LP}$ denote random variables for the queue length distributions of the HP and LP queue after the $i^{th}$ iteration and let $Y_i^{LP}$ and $Y_i^{HP}$ denote the random variables for the corresponding vacation length distributions. From (2) it follows that if $X_0^{HP} \leq_{st} X_1^{HP}$ also $Y_0^{LP} \leq_{st} Y_1^{LP}$ as a higher queue length for the HP queue leads to a longer vacation length for the LP queue. From (3) it follows that if $Y_0^{LP} \leq_{st} Y_1^{LP}$ also $X_0^{LP} \leq_{st} X_1^{LP}$ as a longer vacation for the server of the LP queue leads to a higher number of packets in the LP queue. Following the same reasoning for the LP node, we have that $X_0^{LP} \leq_{st} X_1^{LP}$ leads to $Y_0^{HP} \leq_{st} Y_1^{HP}$ and $Y_0^{HP} \leq_{st} Y_1^{HP}$ leads to $X_1^{HP} \leq_{st} X_2^{HP}$. It thus follows that $X_0^{HP} \leq_{st} X_i^{HP}$ for any $i \geq 1$ as long as $X_0^{HP} \leq_{st} X_1^{HP}$. Similarly we have that $X_0^{HP} \geq_{st} X_i^{HP}$ for any $i \geq 1$ as long as $X_0^{HP} \geq_{st} X_1^{HP}$.
□

From Theorem 1, an obvious approach is to start with

$$P(X_0^{HP} = n) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n > 0 \end{cases} \quad (4)$$

since it then holds that $X_0^{HP} \leq_{st} X$ for any $X$ with a non-negative distribution. Let $X_i^*$ denote the random variable following an equilibrium distribution, that is $X_i^* = X_{i+1}^*$. We then have that as $X_0^{HP} \leq_{st} X_i^*$, also $X_i^{HP} \leq_{st} X_i^*$, so the iteration process cannot jump past an equilibrium. In every iteration, the distribution may change, moving closer towards the equilibrium distribution. Similarly, we can start with the distribution

$$P(X_0^{HP} = n) = \begin{cases} 1 & \text{for } n = B \\ 0 & \text{for } n < B \end{cases} \quad (5)$$

where $B$ is the maximum number of customers in the queue. It then follows that $X_0^{HP} \geq_{st} X$ for any $X$, so that after every step we have that $X_i^{HP} \geq_{st} X_i^*$ as $X_0^{HP} \geq_{st} X_i^*$. In this case every iteration takes a step closer to the equilibrium from above. Using Algorithm 1 starting from both (4) and (5), we find our approximation.

### 2.1.2 Multiple queues

The approach for two queues can easily be extended to multiple queues of any priority class. The vacation length of a considered queue then depends on the state of all the other queues, and can be computed by analogy to (2). The vacation length distribution in this case is given by

$$P(V_x = k\tau | N_y = i_y, y \neq x) = \quad (6)$$
$$\sum_{y \neq x} q_y \sum_{\substack{a_z = i_z, z \neq x, y \\ a_y = i_y - 1}}^{B} P(V_x = (k-1)\tau | N_z = a_z, z \neq x) \cdot$$
$$\prod_{z \neq x, y} P(A_z = a_z - i_z) \cdot P(A_y = a_y - i_y + 1)$$
$$P(V_x = 0 | N_y = i_y, y \neq x) = \frac{q_x}{q_x + \sum_{y, i_y > 0} q_y}$$

Here $q_x$ denotes the probability of the server jumping to queue $x$. The vacation length distribution is found using

$$P(V_x = k\tau) = \quad (7)$$
$$\sum_{i_y = 0, y \neq x}^{B} P(V_x = k\tau | N_y = i_y, y \neq x) P(N_y = i_y, y \neq x)$$

where again the steady state queue length distribution of the other queues is needed. Starting again with a random distribution for all but one queue we find the vacation time for this tagged queue and hence the corresponding steady state queue length distribution of this queue. This distribution can now be used for all queues of the same class and the other class can be analyzed using the steps of the algorithm. Note that the proof of convergence remains the same, as the analysis is done for each type of queue. In the case of multiple queues with balanced load, that is with identical arrival rates at the queues, the random variables $X_i^{HP}$, $X_i^{LP}$, $Y_i^{HP}$ and $Y_i^{LP}$ can be used for all queues of the same type as they are indentical. When arrival rates at the queues are different, the same reasoning can be used for all separate variables $X_i^{HP_j}$, $X_i^{LP_j}$, $Y_i^{HP_j}$ and $Y_i^{LP_j}$, where the subscript $j$ denotes a specific queue of the type HP or LP.

## 2.2 Special cases

For a high priority queue, it may be needed that a certain average waiting time can be guaranteed. To obtain the maximal average waiting time in a network with one HP queue and $n$ LP queues, we give results for the situation with saturated LP queues. To analyze the impact of prioritizing the high priority queue on the low priority queues, we compare the average waiting time at the LP queues without an HP queue in the system, with the case where the HP

queue is saturated. For these special cases, exact results are available, which are given in this section.

### 2.2.1 Saturated LP queues

Consider one high priority queue with Poisson($\lambda_{HP}$) packet arrivals and $n$ saturated low priority queues, i.e. $\lambda_{LP} \to \infty$. Let the probability $q$ of visiting the high priority queue be

$$q = \frac{\alpha}{n+\alpha}$$

where $\alpha$ denotes the factor of importance given to the high priority queue, meaning the probability of visiting the HP queue compared to the LP queue is $\alpha$ times as high. For the HP queue, the vacation length distribution is then given by the geometric distribution

$$P(V = k\tau) = (1-q)^k q$$

as any time the server does not jump to the HP queue, it will service exactly one packet at an LP queue. As the average time between arrivals of the server is $\frac{\tau}{q}$ and the server only serves one customer at each visit, the HP queue is stable when $q > \lambda\tau$. With the exact distribution of the vacation length known, we can use the pgf of the number of customers in the queue as given by 3 to determine the average number of customers in the HP queue. The average waiting time then easily follows from Little's law.

### 2.2.2 Empty and saturated HP queue

We now consider the case where the low priority queues are no longer saturated, but each have an arrival process of rate $\lambda_{LP}$ and a deterministic service time of value $\tau$. Let queue $n+1$ be the HP queue, the conservation law (c.f. [8]) then states that

$$\sum_{i=1}^{n+1} \rho_i EW_i^q = \rho \frac{\sum_{i=1}^{n+1} \lambda_i \beta_i^{(2)}}{2(1-\rho)}$$

where $EW_i^q$ denotes the average waiting time in the queue (not including service) and $\rho_i = \lambda_{LP}\tau$ for $i = 1...n$ and $\rho_{n+1} = \lambda_{HP}\tau$, so that and $\rho = n\rho_{LP} + \rho_{HP}$. As the service time distribution is deterministic for any queue, we have that $\beta_i^{(2)} = \tau^2$ and the total waiting time of a customer is $EW_i = EW_i^q + \tau$.

Consider the case where there are only $n$ LP queues, so the arrival rate at the HP queue is set equal to zero. The stability condition is that $\rho = n\lambda_{LP}\tau < 1$ and it immediately follows that

$$\sum_{i=1}^{n} \rho_{LP} EW_{LP}^q = \rho \frac{\sum_{i=1}^{n} \lambda_{LP}\tau^2}{2(1-\rho)}$$

$$EW_{LP}^q = \frac{\rho\tau}{2(1-\rho)}$$

$$EW_{LP} = \frac{(2-\rho)\tau}{2(1-\rho)}$$

Now consider the case where the HP queue is saturated. We have $n$ identical LP queues, and from the perspective of the LP queues the server incurs a switchover time when it visits the HP queue. The stability condition for this system is that $\frac{n\lambda\sigma}{(1-\rho)} < 1$, where $\sigma$ denotes the mean switchover time, as this is the number of arriving customers during the average cycle time of a queue. Let $p_i$ denote the probability of jumping to queue $i$ and $s_i$ the average time it takes to switch to queue $i$. We have a pseudo-conservation law stating that (c.f. [2])

$$\sum_{i=1}^{n} \rho_i [1 - \frac{\lambda_i}{p_i}\frac{\sigma}{1-\rho}] EW_i =$$

$$\rho \frac{\sum_{i=1}^{n} \lambda_i \beta_i^{(2)}}{2(1-\rho)} + \frac{\sigma}{1-\rho}\sum_{i=1}^{n}\frac{\rho_i}{p_i} - \sum_{i=1}^{n}\rho_i s_i + \frac{\rho}{2\sigma}\sum_{i=1}^{n} p_i s_i^{(2)},$$

where for our model we have that $\lambda_i = \lambda_{LP} = \lambda$, $\rho_i = \lambda\tau$, $\rho = n\lambda\tau$, $\beta_i^{(2)} = \tau^2$, $p_i = \frac{1}{n}$, $s_i = \frac{q\tau}{1-q}$, $s_i^{(2)} = \frac{q(q+1)\tau}{(1-q)^2}$ and $\sigma = s_i$ as all switchover times are equal. Here $q$ denotes the probability of the server polling the HP queue. As the LP queues are statistically identical, the expression simplifies to

$$EW_{LP} = \frac{\frac{n\lambda\tau^2}{2(1-n\lambda\tau)} + \frac{q\tau n}{(1-q)(1-n\lambda\tau)} + \frac{q+1-2q\tau}{2(1-q)}}{[1 - \frac{n\lambda q\tau}{(1-q)(1-n\lambda\tau)}]}$$

and applying Little's law the average total number of customers in the queue is obtained. Note that this approach can easily be extended to a case with multiple high priority queues, as only the probability of the server being on vacation changes, so only the values of $s_i$ and $s_i^{(2)}$ need to be adjusted.

## 3. VALIDATION

In the following we validate our approximation approach by comparison with simulation results. For a wide variety of settings, varying the load of the system and the grade of prioritization, the average waiting times of packets at the individual queues are determined. Note that the approach presented calculates the distribution of the waiting time, but only the averages are used in the following for comparison with simulation. Results for the scenario with one high priority and one or two low priority queues are considered, together with the special cases.

## 3.1 General case

### 3.1.1 Two queues

Table 1 shows the average waiting time of packets in a queue computed by the algorithm compared with simulation results for different loads of the system in the case of two queues, one HP and one LP queue. The table shows the impact of varying $\alpha$, the relative importance of the HP queue compared to a LP queue. The load at the queues is balanced, i.e. each queue has the same arrival rate of packets. The probability of moving to the HP queue is $q = \frac{\alpha}{n+\alpha}$, which is $\alpha$ times as high as for the LP queue and the buffer size is set to 15 for all cases. The impact of the differentiation appears to be higher when the load of the system increases. For a low load, the queues are often empty, thus making it possible for the server to attend to packets directly upon arrival. As the load increases, the queues will be fuller and the waiting time depends more on the frequency at which the server visits the queues. We observe that the accuracy of the algorithm deteriorates as the load of the system increases. For a highly loaded system, the queues will at times be fully loaded, causing arriving packets to be lost. This effect is not taken into account when using the pseudoconservation law to scale the obtained results. Simulation however shows that the impact of this approximation is limited, as the average

**Table 1: Average waiting time in a two node network with balanced load**

| | Rates | | Simulation | | Algorithm | | Error | |
|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $\lambda_{LP}$ | $\lambda_{HP}$ | LP | HP | LP | HP | LP | HP |
| 2 | 0.1 | 0.1 | 0.1129 | 0.1120 | 0.1125 | 0.1125 | 0.3455 | 0.4437 |
| 3 | 0.1 | 0.1 | 0.1132 | 0.1118 | 0.1166 | 0.1084 | 3.0128 | 2.9663 |
| 4 | 0.1 | 0.1 | 0.1133 | 0.1117 | 0.1174 | 0.1076 | 3.6018 | 3.6286 |
| 2 | 0.2 | 0.2 | 0.2723 | 0.2609 | 0.2829 | 0.2504 | 3.9085 | 4.0431 |
| 3 | 0.2 | 0.2 | 0.2753 | 0.2580 | 0.2911 | 0.2422 | 5.7375 | 6.1312 |
| 4 | 0.2 | 0.2 | 0.2771 | 0.2569 | 0.2961 | 0.2373 | 6.8574 | 7.6399 |
| 2 | 0.3 | 0.3 | 0.5623 | 0.4888 | 0.5918 | 0.4582 | 5.2475 | 6.2384 |
| 3 | 0.3 | 0.3 | 0.5792 | 0.4714 | 0.6253 | 0.4247 | 7.9612 | 9.9101 |
| 4 | 0.3 | 0.3 | 0.5881 | 0.4624 | 0.6454 | 0.4046 | 9.7323 | 12.5021 |

number of packets in the system remains close to a system with infinite queues.

In a similar fashion Table 2 shows results for unbalanced arrival rates, with the probability $q = \frac{2}{3}$ ($\alpha = 2$) of visiting the HP queue kept constant. For more unbalanced situations, the results deteriorate, especially for higher loads. For the node with the lower arrival rate, the error made by the algorithm is bigger, as the average queue length is smaller. Comparing the impact of increasing the load of the LP queue on the HP and vice versa shows that the increase in load of the HP queue has a bigger impact on the average waiting time at the LP queue than increasing the load of the LP queue has on the HP queue. As an increase of the load will cause the queue to be non-empty for a larger fraction of the time, the impact it has on the other queue by causing the server to go on a vacation becomes larger. As a HP has a higher probability of being visited, increasing the load of this queue has a bigger impact than increasing the load at the LP queue.

### 3.1.2 Three queues

In Table 3 we consider the scenario with three queues, one HP queue and two LP queues. The table shows the average waiting time of packets computed by the algorithm compared with simulation results for the situation with balanced load. As for the situation with two nodes, we observe that for higher loads, the impact of the prioritization increases. Again, the results deteriorate as the load of the system increases. Comparison with the results of Table 1 furhter shows that the impact of prioritization is higher when more nodes are active in the network. The decrease in the average waiting time of customers for the HP queue is stronger relative to the decrease for the two node situation. With more queues present, the relative increase in probability of being visited is higher when the value of $\alpha$ is increased. For example, increasing the value of $\alpha$ from 2 to 3 for both situations gives the following relative increase (r.i.):

| | $\alpha = 2$ | $\alpha = 3$ | r.i. |
|---|---|---|---|
| 2 nodes | $q = \frac{2}{3}$ | $q = \frac{3}{4}$ | 12.5% |
| 3 nodes | $q = \frac{1}{2}$ | $q = \frac{3}{5}$ | 20% |

For all settings, no more than 15 iterations were needed by the algorithm with the accuracy set in such a way that the last step gave an improvement less that 1%. Longer runs with higher accuracy did not improve the results significantly. To run the iterations, the values of $P(V_x = k\tau | N_y = i)$ for the two node case and $P(V_x = k\tau | N_y = i_y, y \neq x)$ for the three node case had to be computed once using the iterations given in (1) and (6), which is time consuming for large values of the buffer sizes. For highly filled buffers however, the geometric distribution can be used, as the probability of the vacation having a duration of $k\tau$ is then very close to the probability of first visiting $k$ other queues before visiting the considered queue, as the other queues will not become empty during the process. The time needed for the iteration itself is very limited, as (2) (or (7)) only encompasses the addition over all possible values of queue lenghts and (3) is a small enough system of equations to be solved within seconds.

## 3.2 Special cases

### 3.2.1 Saturated low priority queues

For a user with important traffic, the QoS differentiation is of high importance. To get an idea of the impact of the settings for the differentiation, a worst case scenario can be analysed to see the minimal prioritization that is needed to obtain a certain average waiting time for the high priority packets. The worst case scenario is when all other (low priority) queues always have traffic to transmit. Figure 1 shows the average waiting time of a packet in the HP queue, for different values of $n$, the number of saturated low priority queues in the system. The arrival rate at the HP queue is set to $\lambda_{HP} = 0.01$. The three lines represent the results of the model for $\alpha = 2...4$, the grade of prioritization. It clearly follows from the figure that where for a sparce network (low number of LP queues) the differentiation has a limited effect and that for a dense network (high number of LP queues) giving more priority has a much bigger impact.

### 3.2.2 Empty and saturated high priority queue

The differentiation between users is primarily done to provide better performance for more important traffic. However, it also has to be taken into account what the impact is on performance of the less important traffic. If the prioritization of the high priority queue is too high, the low priority queues might be starved. To analyse the impact on the low priority queues, we compare the situation without the HP queue (or an empty HP queue) with the situation that the HP queue always has packets to transmit. In the latter case, we vary the grade of prioritization. Figure 2 shows the average waiting time of a packet in an LP queue, for different values of $n$, for different settings of the HP queue. The arrival rate $\lambda_{LP}$ is set to 0.01 for each of the $n$ LP queues. In this case the HP queue is either absent (or empty) in which

**Table 2: Average waiting time in a two node network with unbalanced load**

| Rates | | Simulation | | Algorithm | | Error | |
|---|---|---|---|---|---|---|---|
| $\lambda_{LP}$ | $\lambda_{HP}$ | LP | HP | LP | HP | LP | HP |
| 0.2 | 0.5 | 0.4724 | 1.0469 | 0.5068 | 1.0098 | 7.2785 | 3.5384 |
| 0.5 | 0.2 | 1.1693 | 0.3475 | 1.2053 | 0.3113 | 3.0780 | 10.4126 |
| 0.1 | 0.01 | 0.1061 | 0.0107 | 0.1061 | 0.0106 | 0.0416 | 0.5174 |
| 0.01 | 0.1 | 0.0106 | 0.1062 | 0.0111 | 0.1057 | 3.9743 | 0.4905 |
| 0.4 | 0.1 | 0.6120 | 0.1384 | 0.6176 | 0.1324 | 0.9038 | 4.3436 |
| 0.1 | 0.4 | 0.1554 | 0.5934 | 0.1712 | 0.5788 | 10.1452 | 2.4608 |
| 0.1 | 0.3 | 0.1373 | 0.3966 | 0.1475 | 0.3858 | 7.3913 | 2.7269 |
| 0.3 | 0.1 | 0.4081 | 0.1286 | 0.4093 | 0.1240 | 0.3084 | 3.5340 |

**Table 3: Average waiting time in a three node network with balanced load**

| | Rates | | Simulation | | Algorithm | | Error | |
|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $\lambda_{LP}$ | $\lambda_{HP}$ | LP | HP | LP | HP | LP | HP |
| 2 | 0.1 | 0.1 | 0.1217 | 0.1205 | 0.1245 | 0.1151 | 2.386 | 4.465 |
| 3 | 0.1 | 0.1 | 0.1228 | 0.1188 | 0.1261 | 0.1122 | 2.639 | 5.556 |
| 4 | 0.1 | 0.1 | 0.1229 | 0.1185 | 0.1269 | 0.1104 | 3.323 | 6.868 |
| 2 | 0.2 | 0.2 | 0.3654 | 0.3202 | 0.3766 | 0.2928 | 3.054 | 7.300 |
| 3 | 0.2 | 0.2 | 0.3711 | 0.3072 | 0.3882 | 0.2736 | 4.602 | 10.949 |
| 4 | 0.2 | 0.2 | 0.3749 | 0.2986 | 0.3946 | 0.2608 | 5.250 | 12.678 |
| 2 | 0.3 | 0.3 | 1.9029 | 0.9133 | 2.0479 | 0.8543 | 7.621 | 6.461 |
| 3 | 0.3 | 0.3 | 2.0098 | 0.7526 | 2.1639 | 0.6221 | 7.669 | 17.337 |
| 4 | 0.3 | 0.3 | 2.0653 | 0.6844 | 2.2162 | 0.5177 | 7.304 | 24.356 |

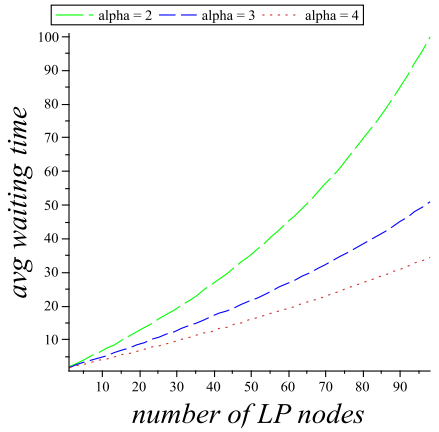


**Figure 1: Average waiting time for the scenario with saturated LP queues**

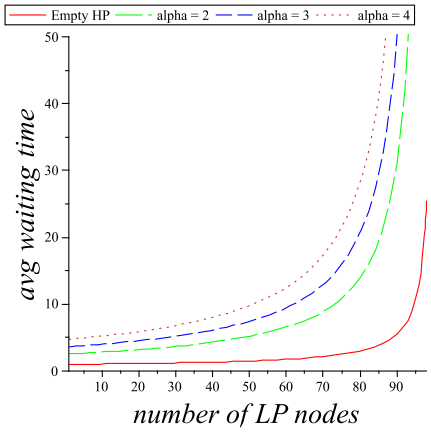


**Figure 2: Average waiting times for the scenario with an empty or saturated HP node**

case the complete network behaves as a standard M/D/1 queue where each separate queue has the same average behaviour or the HP is saturated, with different values for $\alpha$, the grade of prioritization. For higher values of $\alpha$ the server will more often be processing HP packets, leaving less capacity for the LP queues. This shows from the figure as the waiting time reaches high values already for lower values of $n$. When the network is sparce, we see there is already a substantial impact of the differentiation on the waiting time of the low priority packets.

## 4. CONCLUSION

In this paper we analyzed the impact of QoS differentiation on the delay of packets for different classes of queues using a 1-limited polling model with a random scheduling policy and deterministic service times, capturing the random nature of the MAC layer protocol. The model gives insight in the effect of the parameter settings on the QoS in a WLAN for the individual classes of queues. We developed an approximation approach for the packet delay in a network with high and low priority queues. Comparison with simulation results shows that for low to moderately loaded systems, the approach works well.

## 5. REFERENCES

[1] Bianchi, G., Performance analysis of the IEEE 802.11 distributed coordination function, *IEEE Journal on Selected Areas in Communications*, vol. 18,

nr. 3, 535-547, 2000.

[2] Boxma, O.J. and Weststrate, J.A., Waiting times in polling systems with Markovian server routing, *Informatik-Fachberichte*, vol. 218, 89-104, 1989.

[3] Cheung, S-K., van den Berg, J.L. and Boucherie, R.J., Decomposing the queue length distribution of processor-sharing models into queue lengths of permanent customer queues, *Performance Evaluation*, vol. 62, 100-116, 2005.

[4] Engelstad, P.E. and Østerbø, O.N., Non-Saturation and Saturation Analysis of IEEE 802.11e EDCA with Starvation Prediction, *Proceedings of ACM MSWiM '05,* Montreal, Canada, 2005.

[5] Fuhrmann, S.W., A Note on the M/G/1 Queue with Server Vacations, *Operations Research*, vol. 32, no. 6, 1368-1373, 1984.

[6] Fuhrmann, S.W. and Cooper, R.B., Stochastic decompositions in the M/G/1 queue with generalized vacations, *Operations Research*, vol. 33, 1117–1129, 1985.

[7] Fuhrmann, S.W., Symmetric queues served in cyclic order, *Operations Research Letters*, vol. 4, nr. 3, 139-144, 1985.

[8] Groenendijk, W.P., *Conservation laws in polling systems*, Ph.D. Thesis, University of Utrecht, 1990.

[9] Hadzi-Velkov, Z. and Spasenovski, B., Capture effect in IEEE 802.11 Wireless LANs, *Proceedings of IEEE ICWLHN '01*, Singapore, 2001.

[10] IEEE, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Medium Access Control (MAC) Quality of Service Enhancements, IEEE std. 802.11e, 2005.

[11] Kramer, M., Stationary distributions in a queueing system with vacation times and limited service, *Queueing Systems*, vol. 4, 57-68, 1989.

[12] Lee, T.T., M/G/1/N queue with vacation time and limited service discipline, *Performance Evaluation*, vol. 9, no. 3, 180-190, 1989.

[13] Levy, H., Polling Systems: Applications, Modeling, and Optimization, *IEEE Transactions on Communications*, vol. 38, no. 10, 1990.

[14] Litjens, R., Roijers, F., van den Berg, J.L., Boucherie, R.J. and Fleuren, M., "Performance analysis of Wireless LANs: An integrated packet/flow level approach", *Proceedings of the 18th International Teletraffic Congress,* Berlin, Germany, 651-660, 2003.

[15] Ross, S.M., *Stochastic Processes*, second edition, John Wiley & Sons, New York, 1996.

[16] Wu, H. et al., Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement, *Proceedings of IEEE INFOCOM '02*, New York, USA, 2002

[17] Xiao, Y., Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs, *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, July 2005.

[18] Xiong, L. and Mao, G., Saturated Throughput Analysis of IEEE 802.11e Using Two-Dimensional Markov Chain Model, *Proceedings QShine '06*, Waterloo, Canada, 2006.

[19] Zhu, H. and Chlamtac, I., An Analytic Model for IEEE 802.11e EDCF Differential Services, *Proceedings of IEEE ICCCN '03*, Dallas, USA, 2003.