Optimal Scheduling for Multimedia Traffic in a Wireless LAN*

Munish Goyal, Anurag Kumar and Vinod Sharma

Dept. of Electrical Communication Engineering,
Indian Institute of Science, Bangalore, India.
{munish,anurag, vinod}@ece.iisc.ernet.in

Abstract. We develop polling strategies for carrying multimedia traffic over a polled multiple access based fading wireless local area network. We consider a slotted system with two classes of traffic (voice and streaming media), Markov arrivals and Markov fading. The performance objectives are a loss probability for voice and discounted queueing delay for the streaming media. An index policy based on the current channel state information is obtained via the Whittle relaxation of a constrained Markov decision problem. The proposed policy would yield a schedulable region that will guarantee the desired quality of service for the admitted calls. We provide numerical results for the proposed policy comparing it with other known policies such as certain stabilizing policies and the weighted round-robin policy. We propose a scheme for exchanging the control and measurement information between the controller and the controlled stations while staying within the IEEE 802.11 wireless LAN standard.
Keywords: QoS in wireless LANs, Scheduling over fading channels, Markov decision problems, Index policies.

1 Introduction

We consider a home or office environment, where mobile stations (MSs) communicate with the external world through an access point (AP) using an IEEE 802.11 based wireless LAN. Figure 1 depicts a possible scenario in an IEEE 802.11 WLAN based home network. Based on the traffic characteristics and quality of service (QoS) requirements, the MSs can broadly be divided into two categories: MSs handling packet voice telephony (e.g., 802.11 packet phones), and MSs handling streaming media transfer (e.g, a TV receiver). We define two sets $\mathcal{N}_V$ and $\mathcal{N}_S$ with the subscript representing a station handling a voice call (between an MS and the AP) or a streaming media transfer; $\mathcal{N}_\mathcal{V} \bigcup \mathcal{N}_\mathcal{S} := \mathcal{N}$. A voice call produces periodic packets in each direction and due to strict delay requirement the packets that exceed their delay target are assumed to be lost. The QoS requirement for streaming media is not as stringent as for voice. In this paper, we will capture the performance requirement for streaming media by associating a cost with the buffering at the wireless interface in the AP. Since we deal only with real-time interactive and streaming traffic, we focus on the point coordination function (PCF), as defined in IEEE 802.11 [1], provided mainly to support time bounded services.

PCF provides a centralized, contention-free channel access, based on a poll-and-response mechanism. The stations have contention-free access to the wireless medium, coordinated by a Point Coordinator (PC), which is co-located within the access point (AP). A virtual connection is established before commencing a transfer requiring the parameterized quality of service (QoS). A set of traffic characteristics are negotiated between the AP and the corresponding station. Accordingly, the AP implements an admission control algorithm to determine whether to admit a specific connection or not. Once a connection is set up, the point coordinator (PC) co-located within the AP endeavors to provide the contracted QoS by allocating the required resources. In order to meet the contracted QoS requirements, the PC needs to schedule the data and poll frame transmissions properly. Since the wireless medium involves the time-varying and location-dependent channel conditions, developing a good scheduling algorithm is a challenging problem. An efficient scheduling algorithm can result in better system performance.

In a typical frame exchange sequence as shown in Figure 2, the PC starts off by polling a station asking for a pending frame. If the PC itself has pending data for this station, it uses a combined data and poll frame by piggybacking the poll frame into the data frame. Upon being polled, the polled station

* This work was supported by a research grant from Intel.

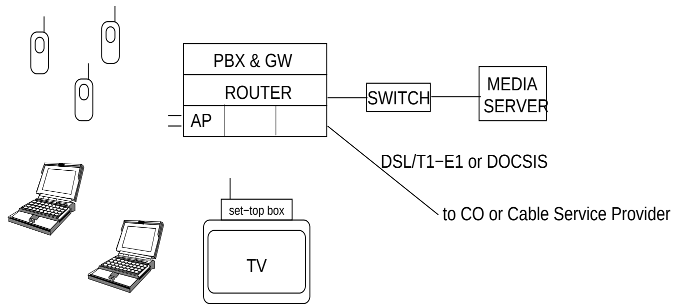

Figure 1. A home or small office wireless local area network being used for telephony and streaming media playback. The system in the large box handles the physical and MAC layers of 802.11.

acknowledges the successful reception along with data. If the PC receives no response from a polled station after waiting for a PCF interframe spacing (PIFS) interval, it polls the next station, or ends the contention free period (CFP). Therefore, no idle period longer than PIFS occurs during CFP. The PC continues with polling other stations until the CFP expires. A specific control frame, called CF-End, is transmitted by the PC as the last frame within the CFP to signal the end of the CFP.

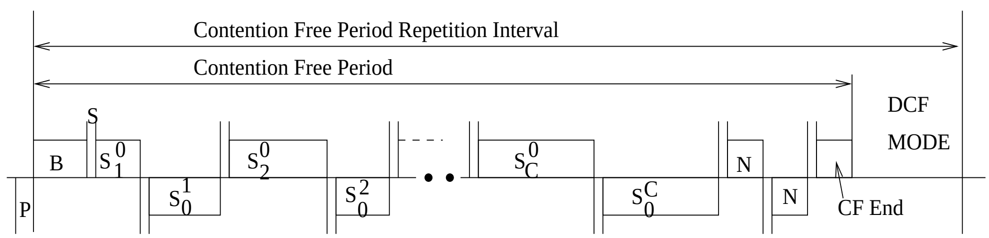

Figure 2. A typical frame exchange sequence in an 802.11 based wireless LAN. The point coordinator is indexed 0, S_j^k represents the packet size for a transmission from k to j, B denotes the beacon, P stands for PIFS, S for SIFS, and N represents no data transmission. The devices $\{1, 2, \cdots, C\} \in \mathcal{N}$ could send the data packet in the frame owing to the frame boundary constraint. Rest of the devices send NULL data packets.

With the above situation in mind we consider a model with periodic frames of equal length. Voice and streaming packets arrive at each device. These arrivals are modelled as Markov processes. The channel gain between any transmitter-receiver pair is constant over each frame but varies in Markovian manner from frame to frame. With this model our aim is to develop dynamic scheduling policies that optimize certain long run performance objectives. We model the system mathematically and analyse it using the dynamic programming approach. In accessing the wireless channel, the voice calls would be given priority over streaming sessions as packets of the latter can be buffered. The cost associated with the voice calls is the number of packets dropped due to violation of the delay constraint. The cost associated with a streaming media transfer is the discounted buffering cost. Recently, there has been a lot of interest in delay optimal scheduling of transmissions over fading wireless networks [3], [5], [11]. The optimal policies more often than not turn out to be too complicated. The major contribution of this work is the development of index based polling strategies for carrying multimedia traffic, over IEEE 802.11 based wireless LANs, in the polled mode.

This paper is organized as follows. In Section 2, we model the system under consideration. We formulate the problem mathematically in Section 3. We obtain polling strategy for the voice calls in Section 4. We consider the performance optimization problem for streaming calls in Section 5 followed by a formulation of a relaxed version of the problem in Section 5.3. This is followed by a detailed analysis of the relaxed problem using the dynamic programing technique. An index based heuristic polling policy for streaming calls is obtained in Section 5.6.

2 System Model

A source for a multimedia transfer could be an access point (AP), a mobile station (MS) or even both (as in a voice call). Further, an access point could be a source for many such transfers. Henceforth, we refer to the source of each transfer by a device. Let there be a set $\mathcal{N}$ of such devices. We focus on the contention free periods; and divide time into fixed length frames of duration τ seconds each. A voice source $i \in \mathcal{N}_V$ generates a packet of size b_i in each frame. A voice packet that cannot be sent in the next frame is considered lost. A streaming source $i \in \mathcal{N}_S$ (for example a variable rate coded video source) places a random number of packets of length b_i each, into its transmitter buffer (infinite capacity) at the start of each frame. We assume that the packet arrival process $A_i[n]$ is a finite state Markov chain with a single ergodic class and the transition probability matrix is $\mathbf{P}_i^{(a)}$ for $i \in \mathcal{N}_S$.

The channel "Power" gain process on a link between a device and its sink is assumed to remain constant over the duration of a frame and is modelled as a finite state Markov chain with single ergodic class, embedded at the frame boundary instants, with transition probability matrix $\mathbf{P}_i^{(h)}$. The channel gain process is assumed to be independent from one link to another. The IEEE 802.11 standard imposes a peak power constraint for all devices. Based on the link gains, we can compute a maximum reliable transmission rate for each device when transmitting at this peak power level. This is done using the well known mapping between signal to noise ratio and the transmission rate for reliable transmission. Let $R_i[n]$ be the packet transmission rate from node i in frame n. It follows that the process $R_i[n]$ for transmitter i is also a finite Markov chain with transition matrix $\mathbf{P}_i^{(r)}$. For simplicity, we assume that the available rate is bounded strictly away from zero.

At time instant $n\tau, n = \{0, 1, 2 \cdots\}$, the AP is provided with the information about the available transmission rates $r[n]$ and the number of packets $a[n]$ that arrive during the previous frame. We will later comment on how this information can be supplied to the AP by using some spare bits available in the MAC layer header. Based on this and the information about the queue lengths at each device (can be tracked by the AP), the AP decides upon a subset of devices who can send and how much they can send in the current frame, i.e., during the time period $[n\tau, (n+1)\tau)$. The objective of the access point, which acts as a controller, is to obtain an optimal resource (frame time) allocation or polling strategy that guarantees a desired quality of service for each device subject to the constraints imposed by the wireless LAN standard (IEEE 802.11). This policy would yield a schedulable region comprising of sets $\mathcal{N}_V$ and $\mathcal{N}_S$ which can be handled by the system so that each session obtains its desired QoS. Figure 3 shows a typical schedulable region.

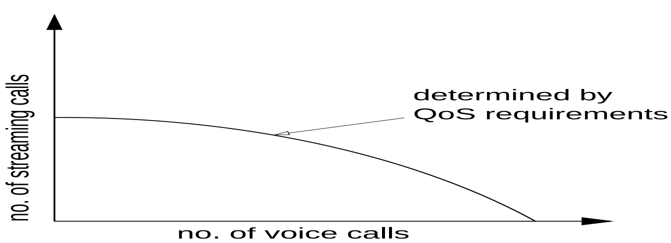

Figure 3. A schedulable region for streaming and voice calls

Exchange of control information: We propose to use the DURATION/ID field in the MAC header to convey the rate measurement information from AP to the device and the arrival information from device to the AP. Duration/ID field normally contains the time needed to transmit the current packet/frame. For the frames transmitted during the PCF mode, the duration field is always set to 32768 and is thus unused. We further assume that each device is polled at least once in each frame even when it is optimal not to send any data. This helps in getting fresh information. This can be done using a CF-Poll+CF-Ack (no data) type frame (See [1]).

3 Problem Formulation

We associate with device $i \in \mathcal{N}$ a weight ω_i defining its priority over other devices. Consider a voice call $i \in \mathcal{N}_V$ and let $S_i[n] \in \{0, 1\}$ be the number of packets for call i transmitted in the n^{th} frame, i.e., during time $[n\tau, (n+1)\tau)$, where $n = \{0, 1, 2, \cdots\}$ at a rate $R_i[n]$. The objective of minimizing the packet loss probability is captured by maximizing the expected number of packets transmitted. The controller objective is

$$\max_{\{S_i \in \{0,1\}, i \in \mathcal{N}_V\}} \left\{ \sum_{i \in \mathcal{N}_V} \omega_i S_i : \sum_{i \in \mathcal{N}_V} \frac{S_i}{R_i} \leq \tau \right\}. \tag{1}$$

Let $\tau_V[n]$ denote the time occupied by voice packets in frame n. Next we consider a streaming device $i \in \mathcal{N}_S$. Let $A_i[n]$ be the number of packets that arrive during $[(n-1)\tau, n\tau)$ (see Fig. 4). Arriving packets are placed into the transmitter buffer at the end of each frame. Let $Q_i[n]$ be the queue length at time instant $n\tau$ for device i. Let $S_i[n]$ be the number of packets transmitted in the n^{th} frame, i.e., during $[n\tau, (n+1)\tau)$. obviously, $S_i[n] \in [0, Q_i[n]]$, a natural constraint that one can transmit only up to whatever is available in the buffer. The transmitter queue evolves according to the equation $Q_i[n+1] = Q_i[n] - S_i[n] + A_i[n]$ (See Figure 4).

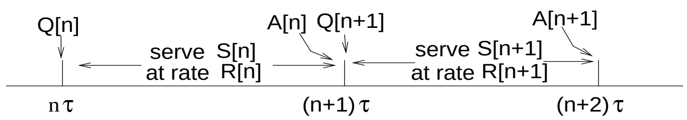

Figure 4. System Model

Focusing only on the streaming transfers, the quadruplet $\mathbf{X} = (\mathbf{Q}, \mathbf{R}, \mathbf{A}, \tau_V)$ defines the state of the system. The quality of service measure is $\sum_{k=0}^{\infty} \alpha^k Q_i[k]$, where $\alpha \in (0,1)$. The parameter α is a discount factor. If α is small, the recent queue lengths have more value than those in a distant future whereas if α is large, queue lengths in a distant future are also important. The controller objective is to obtain a sequence $\{S_i[k]\}, i \in \mathcal{N}_S$ that minimizes a weighted sum of the performance measure

$$\sum_{i \in \mathcal{N}_S} \omega_i E\Big[\sum_{k=0}^{\infty} \alpha^k Q_i[k]\Big], \text{ subject to, } \sum_{i \in \mathcal{N}_S} \frac{S_i[k]}{R_i[k]} \leq \tau - \tau_V[k];\ S_i[k] \in \{0, 1, \cdots, Q_i[k]\} \tag{2}$$

where the measure over which the expectation operator E is taken is conditioned on the state at time $k = 0$, and the actions $\mathbf{S}[k] = \{S_i[k], i \in \mathcal{N}\}$ based on the history of the process. This is a Markov Decision process with state dependent action space. Recall that sequence of actions $S_i[k]$ are integer valued.

4 Analysis: Voice Calls

First, we consider the problem stated in Equation (1). This problem is identical to a knapsack problem where there are certain quantities of material of different densities, and different sizes having different associated values per unit quantity. Amount of the materials need to be chosen to fit into a container while maximizing the aggregate value. During the n^{th} frame, the knapsack volume is the frame time τ, the frame time per packet for the i^{th} call is $\frac{1}{R_i[n]}$ and the value per packet associated with the i^{th} call is ω_i. The following is a well known heuristic for the above said problem obtained from a linear relaxation of the integer knapsack problem [4].

Order the devices in decreasing order of $\omega_i R_i[n]$; this can be interpreted as the reward per unit transmission time for device i. Determine $m_V[n]$ so that the $(m_V[n]+1)^{th}$ queue in this order causes the total transmission time to exceed τ, the frame duration. Now, for a queue i among the top $m_V[n]$ queues in this order $s_i[n] = 1$, and $s_i[n] = 0$ for the rest. We could have sent a fraction of the packet at $(m_V[n]+1)^{th}$ queue but this would violate our modelling assumption that a packet cannot be fragmented. This policy yields a schedulable region for the voice calls determined by the QoS requirements.

If the number of active voice calls lie strictly inside the schedulable region (See Figure 3), some frame time will be available for the streaming media transfers. Alternatively to accommodate streaming media transfers the number of active voice calls will need to be restricted. Define $\tau_S[k] = \tau - \sum_{i=1}^{m_V[k]} \frac{1}{R_i[k]}$, the frame time available for streaming media in the k^{th} frame.

5 Analysis: Streaming Transfers

In view of the above result, the problem stated in Equation 2 can be restated as follows. For notational ease, we denote the random process representing the frame time available for streaming transfers $\tau_S[k]$ by $T[k]$. A realization of $T[k]$ will be denoted by t. Note that the process $T[k]$ is a Markov chain with finite state space since $\tau_V[k]$ can assume only finitely many values. The state of the system is now a quadruplet $\mathbf{X} = (\mathbf{Q}, \mathbf{R}, \mathbf{A}, T)$. The controller objective is to obtain a sequence $\{S_i[n]\}, i \in \mathcal{N}_S$ that solves

$$\min \sum_{i \in \mathcal{N}_S} \omega_i E\Big[\sum_{k=0}^{\infty} \alpha^k Q_i[k]\Big], \text{ subject to, } \sum_{i \in \mathcal{N}_S} \frac{S_i[k]}{R_i[k]} \leq T[k];\; S_i[k] \in \{0, 1, \cdots, Q_i[k]\},\; i \in \mathcal{N}_S \quad (3)$$

First, we review the well known results on a characterization of the stability region and a class of stabilizing policies for the above said problem. Then using a heuristic based on the MDP formulation, we obtain a new index based polling policy.

5.1 Stabilizing Policies (A Review)

At time k, the system state is the rate vector $\mathbf{R}[k]$ and the transmission time $T[k]$ available for streaming media after serving voice traffic. Since the state space is finite, we index the state by m. Then a state m corresponds to rate vector $\{r_i^m,\ i \in \mathcal{N}_S\}$ and frame time t^m. Let π^m be the stationary probability of being in state m. We associate with each state m a fixed vector μ^m with $\mu_i^m = \lfloor t^m r_i^m \rfloor$ representing the maximum number of packets that can be sent by user i if whole service effort is allocated to i. Let λ be the mean arrival rate vector. Consider a static service split policy ϕ (Refer [2]) where ϕ_i^m denote the fraction of service effort allocated to user i in state m. The matrix ϕ, with rows corresponding to the possible state m, is a stochastic matrix. Thus under ϕ, the service is applied to i with probability ϕ_i^m when the system state is m.

Theorem 1 **([2]).** *A scheduling rule under which the system is stable exists if and only if there exists a stochastic matrix ϕ such that $\lambda_i < \sum_m \pi^m \phi_i^m \mu_i^m$. Given a vector of arrival rates λ satisfying the above condition, there exists a positive vector α such that the non random static service split policy with $\phi_i^m = 1 \Rightarrow i = \arg\max \alpha_j \mu_j^m$ is stabilizing.* □

Remark: Consider a policy that in state m orders the devices in decreasing order of $\alpha_j \mu_j^m$ and the user with highest index transmits until it empties its queue or the slot ends followed by transmissions by the device next in the order and so on. Note that this policy performs at least as good as the stabilizing policy defined above. A policy that orders the transmissions in decreasing order of $\{\omega_i r_i q_i\}$ is also known to be stabilizing [2]. We call it the workload based policy.

5.2 Index Policies and Whittle's Relaxation

It should be noted here that a stabilizing policy may not perform well in terms of the objectives in Equation 3. Let us look at the discounted cost value iteration algorithm for solving the problem (Equation 3) to motivate the approach that we will follow in the rest of the paper. For a given state $\mathbf{x} = (\mathbf{q}, \mathbf{r}, \mathbf{a}, t)$, define the constraint set $S(\mathbf{x}) = \{\mathbf{s} : \mathbf{s} \in [0, \mathbf{q}];\ \ \sum_{i \in \mathcal{N}_S} \frac{s_i}{r_i} \leq t\}$. Consider the following value iteration algorithm,

$$V_{n+1}(\mathbf{x}) = \min_{\mathbf{s} \in S(\mathbf{x})} \Big\{ \sum_{i \in \mathcal{N}_S} \omega_i q_i + \alpha E_{\mathbf{a}, \mathbf{r}, t}[V_n(\mathbf{q} - \mathbf{s} + \mathbf{A}, \mathbf{R}, \mathbf{A}, \mathbf{T})] \Big\}.$$

where $E_{a,r,t}[\cdot]$ denotes the conditional expectation with respect to the arrival, rate and the T processes. Let f_n be the optimal policy for the n^{th} stage problem. Initialize $V_0(\mathbf{x}) = 0$. This implies $V_1(\mathbf{x}) = \sum_{i \in \mathcal{N}_S} \omega_i q_i$. Thus $f_2(\mathbf{x})$ is $\arg\min_{\mathbf{s} \in S(\mathbf{x})} \big\{ \sum_{i \in \mathcal{N}_S} \omega_i (q_i(1 + \alpha) - \alpha s_i + \alpha E_{a_i}[A]) \big\}$. This is a knapsack problem. Using Lagrangian approach, we associate a multiplier β and thus $f_2(x, \beta)$ equals $\arg\min_{\mathbf{s} \in [0, \mathbf{q}]} \{\sum_{i \in \mathcal{N}_S} \beta \frac{s_i}{r_i} - \omega_i \alpha s_i\}$. The knapsack heuristic solution is $f_2(x, \beta)|_i = q_i \theta_i(r_i, \beta)$, where $\theta_i(r_i, \beta) = I_{\{\omega_i \alpha r_i \geq \beta\}}$. The parameter β solves for the frame boundary constraint. In other words the solution is to order the users in decreasing order of $\omega_i r_i$ and the user with highest index transmits until the frame boundary constraint is exceeded or there is no data for transmission. This is an index policy. The index $\omega_i r_i$ is essentially that value of β at which the system makes a transition from active action to passive action; i.e., if $\beta > \omega_i r_i \alpha$ then $\theta_i(r_i, \beta) = 0$ and $\theta_i(r_i, \beta) = 1$ otherwise.

The function $V_2(\mathbf{x})$ is too complex to carry out any further iteration. Moreover, we are interested in index based policies similar to the one obtained for the voice calls because of their ease in implementation. There has been a lot of work on obtaining index based policies for bandit problems. For multiarmed bandit problems, it is well known that the policies based on Gittin's indices are optimal [8]. Gittin showed that to each project one could associate an index $\nu_i(x_i)$, a function of the project i and its state x_i alone, and that the optimal policy is to operate the one with the largest index.

Consider the "restless bandits" problem of designing an optimal sequential resource allocation policy for a collection of stochastic projects (say M), each of which is modelled as a Markov decision chain having two actions at each state with associated rewards; an active action, which corresponds to engaging the project, and a passive action, which corresponds to letting it go. A fixed number of resources needs to be allocated; i.e., at each time instant a fixed number of projects (say k) are active. The passive projects can change state, in general through a given transition rule. The performance objective is to maximize the time-averaged reward rate. Whittle [10] presented a simple heuristic based on a tractable optimal solution to a relaxed version, where instead of requiring that k projects be active at any time, *k projects are needed to be active on average.* This yielded an upper bound on the optimal reward. Further the heuristic policy is a priority index rule associated with each project, that engages the top k projects at any given point of time. Recent work of Nino-Mora [6] is nearly a complete reference for restless bandit problems.

Motivated by the Whittle's work on restless bandits, we introduce a relaxed problem. The state of the system is denoted by $\mathbf{x} = (\mathbf{q}, \mathbf{r}, \mathbf{a}, t) \in \mathcal{X}$. The set of feasible actions in state $\mathbf{x}$ is $S(\mathbf{x}) = [0, \mathbf{q}]$. Let Π be the space of all feasible policies. A deterministic, stationary Markov policy $f \in \Pi$ is a measurable mapping from $\mathcal{X}$ to $[0, \mathbf{q}]$. For every $\beta > 0$, the Lagrange multiplier, defines a cost function $c_\beta(\mathbf{x}, \mathbf{s}) = \sum_{i \in \mathcal{N}_S}(\omega_i q_i + \beta \frac{s_i}{r_i})$. The term $\beta \frac{s_i}{r_i}$ can be seen as a relaxed frame boundary constraint. The Lagrange multiplier β has an economic interpretation. The value $\beta \frac{s_i}{r_i}$ is a penalty for transmitting more data and thus reducing the frame time possibly available for other users. There is a tradeoff. If a user sends more he reduces his queue, but he also gets a higher penalty for doing so. Obviously, the penalty increases with s_i. The relaxed problem is to obtain a policy $\pi \in \Pi$ that minimizes the expected discounted cost $E^\pi_{\mathbf{x}}[\sum_{k=0}^\infty \alpha^k c_\beta(\mathbf{X}[k], \mathbf{S}[k])]$. Note that the relaxed problem is separable. Thus we solve it for each i. The amount of data s_i that can be transmitted in a frame of length t should satisfy $\frac{s_i}{r_i} \leq t$, the residual frame boundary constraint. We drop the subscripts i, α. Without loss of generality assume that $\omega = 1$. Exploring the separability, the relaxed problem (RP) for each user is

$$V(x) = \min_\pi E^\pi_x \left[\sum_{k=0}^\infty \alpha^k \left(Q[k] + \beta \frac{S[k]}{R[k]} \right) \right], \quad \text{subject to,} \quad S[k] \in \{0, 1, \cdots, Q[k]\}, \; \frac{S[k]}{R[k]} \leq T[k], \; \forall k.$$

Note that we have relaxed the sum constraint but not the individual constraint.

5.3 Analysis of the Relaxed Problem

The state x is the quadruplet (q, r, a, t). Our model satisfies the nominal conditions (See [7], Proposition 2.1) required for the existence of the discount optimal stationary policy, and the value function $V(x)$ is obtained as a solution to the following dynamic programming optimality equation. Define $u = q - s$ and $U(x) = \{u \text{ integer} : (q - tr)^+ \leq u \leq q\}$. The variable u is the unfinished work after the policy has acted in an interval. Then

$$V(q, r, a, t) = \min_{u \in U(x)} \{q(1 + \frac{\beta}{r}) - \beta \frac{u}{r} + \alpha E_{a,r,t} V(u + A, R, A, T)\}. \tag{4}$$

Define $H(u, r, a, t) = E_{a,r,t} V(u + A, R, A, T)$.

Theorem 2. *$V(u, r, a, t)$ and hence $H(u, r, a, t)$ is convex nondecreasing in u.*

Proof. See Appendix. □

The unconstrained minimizer $u^*(r, a, t)$ in (4) is the value of u that solves the following inequalities,

$$H(u, r, a, t) - H(u - 1, r, a, t) \leq \frac{\beta}{r\alpha} \leq H(u + 1, r, a, t) - H(u, r, a, t).$$

Note that the unconstrained minimizer is not a function of q. The solution for the constrained problem $(u \in U(x))$ is,

– $s(x) = 0$ for $q < u^*(r,a,t)$
– $s(x) = \lfloor tr \rfloor$ for $q > u^*(r,a,t) + \lfloor tr \rfloor$
– $s(x) = q - u^*(r,a,t)$ otherwise

The solution is depicted in Figure 5. Observe that $u^*(r,a,t) = q$ is the breakoff point that will be used to define the indices as in [10] as it is the boundary between not sending anything from the queue and sending something.

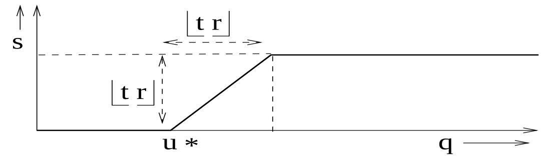

Figure 5. Characterization of the constrained solution

5.4 An Algorithm for Computing $u^*(\cdot)$

Consider the discounted cost value iteration algorithm corresponding to the above said relaxed problem (4).

$$V_n(q,r,a,t) = \min_{u \in S(q,r,a,t)} \left\{ q(1+\frac{\beta}{r}) - \beta\frac{u}{r} + \alpha E_{a,r,t} V_{n-1}(u+A,R,A,T) \right\} \tag{5}$$

It follow from the proof of Theorem 2 that the functions $H_n(u,r,a,t)$ are convex in u for each n. Let $u_n^*(r,a,t)$ be the value of u that solves the following inequalities,

$$H_n(u,r,a,t) - H_n(u-1,r,a,t) \leq \frac{\beta}{\alpha r} \leq H_n(u+1,r,a,t) - H_n(u,r,a,t).$$

Based on the above said constrained solution, we have,
– If $q \leq u_n^*(r,a,t)$, $\quad V_{n+1}(q,r,a,t) - V_{n+1}(q-1,r,a,t) = 1 + \alpha(H_n(q,r,a,t) - H_n(q-1,r,a,t))$
– If $u_n^*(r,a,t) < q \leq \lfloor tr \rfloor + u_n^*(r,a,t)$, $\quad V_{n+1}(q,r,a,t) - V_{n+1}(q-1,r,a,t) = 1 + \frac{\beta}{r}$
– If $q > u_n^*(r,a,t) + \lfloor tr \rfloor$,

$$V_{n+1}(q,r,a,t) - V_{n+1}(q-1,r,a,t) = 1 + \alpha(H_n(q - \lfloor tr \rfloor, r,a,t) - H_n(q - \lfloor tr \rfloor - 1, r,a,t))$$

Define $W_n(q,r,a,t) = V_n(q,r,a,t) - V_n(q-1,r,a,t)$. Thus $H_n(q,r,a,t) - H_n(q-1,r,a,t) = E_{a,r,t} W_n(q+A,R,A,T)$. Then the iterative algorithm to compute $u^*(r,a,t)$ is as follows. Initialize $W_0(q,r,a,t) = 0$. Let $u_n^*(r,a,t)$ be the value of u that solves the following inequalities,

$$E_{a,r,t} W_n(u+A,R,A,T) \leq \frac{\beta}{\alpha r} \leq E_{a,r,t} W_n(u+1+A,R,A,T).$$

– If $q \leq u_n^*(r,a,t)$, $W_{n+1}(q,r,a,t) = 1 + \alpha E_{a,r,t} W_n(q+A,R,A,T)$.
– If $u_n^*(r,a,t) < q \leq \lfloor tr \rfloor + u_n^*(r,a,t)$, $W_{n+1}(q,r,a,t) = 1 + \frac{\beta}{r}$.
– If $q > u_n^*(r,a,t) + \lfloor tr \rfloor$, $W_{n+1}(q,r,a,t) = 1 + \alpha E_{a,r,t} W_n(q - \lfloor tr \rfloor + A,R,A,T)$.

The convergence of the value iteration algorithm (5) ensures that this algorithm converges and $u_n^*(r,a,t)$ converges to the optimal solution $u^*(r,a,t)$.

5.5 Indexability

Definition 1. *(Indexability) [10]: The system is said to be indexable if the set of states where a passive action is taken increases monotonically from an empty set to the full set as the parameter β increases from 0 to ∞.*

For our problem the requirement is natural. As the penalty β for using the frame time increases, we choose to transmit less and less. We show that the relaxed problem is indexable in the sense of the above definition and obtain indices associated with each state. Given the state (q,r,a,t), based on the constrained solution, an active action (a packet is transmitted) is taken if $q > u^*(r,a,t)$ and the action is passive (no transmission) otherwise.

Theorem 3. *As $\beta \to 0$, the solution $u^*(r,a,t) \to 0$ and $u^*(r,a,t) = \infty$ for $\beta > \frac{\alpha r_{\max}}{1-\alpha}$.*

Proof. (Sketch) As $\beta \to 0$, Equation 4 implies that the cost of serving decreases to zero except that the constraint should be satisfied. Thus the solution would be to serve as much as possible, i.e., $s(x) \to \min(q, \lfloor tr \rfloor)$. Thus the action is active in any state where it is possible to do so. To show the other part, it is enough to show that $W_n(q,r,a,t) \le \frac{1}{1-\alpha}$. Since $W_0(q,r,a,t) = 0$, if $\beta > \frac{\alpha}{1-\alpha} r_{\max}$, then $u_0^*(r,a,t) = \infty$ and $W_1(q,r,a,t) = 1$. Let $W_n(q,r,a,t) \le \frac{1}{1-\alpha}$. Then $u_n^*(r,a,t) = \infty$ and $W_{n+1}(q,r,a,t) \le 1 + \frac{\alpha}{1-\alpha}$. By induction hypothesis it follows that $W(q,r,a,t) \le \frac{1}{1-\alpha}$ and $u^*(r,a,t) = \infty$. Thus all actions are passive. □

Given a state $x = (q,r,a,t)$ with $q > 0$, the amount served $s(x)$ decreases to zero as β increases and $s(x) = 0$ for $\beta > \frac{\alpha}{1-\alpha} r_{\max}$. This is natural to expect since the larger is the β, higher is penalty for transmitting.

Theorem 4. *If $\beta < \frac{\alpha r_{\max}}{1-\alpha}$, then the solution $u^*(r,a,t) = 0$ for $r = r_{\max}$.*

Proof. (Sketch) Observe that for n (iteration index) satisfying $\frac{1-\alpha^n}{1-\alpha} < \frac{\beta}{\alpha r_{\max}}$, the optimal policy $u_n^*(r,a,t) = \infty$ and $W_n(q,r,a,t) = \frac{1-\alpha^n}{1-\alpha}$. Since $\beta < \frac{\alpha r_{\max}}{1-\alpha}$, $k = \min\{n : \frac{1-\alpha^n}{1-\alpha} \ge \frac{\beta}{\alpha r_{\max}}\}$ is finite. It follows that $u_k^*(r_{\max},a,t) = 0$ and $W_{k+1}(q,r,a,t) \ge 1 + \frac{\beta}{r_{\max}}$. Since $W_n(\cdot)$ is increasing in n, it can be shown that for $\beta < \frac{\alpha r_{\max}}{1-\alpha}$, $W_n(q,r,a,t) \ge 1 + \frac{\beta}{r_{\max}}$ for all $n > k$. This would imply that $u_n^*(r_{\max},a,t) = 0$ for all $n > k$. Hence the results follows by induction. □

Lemma 1. *$W_n(q,r,a,t)$ is nondecreasing in q for each n.*

Proof. The result follows from the convexity of $V_n(q,r,a,t)$ in q. □

Theorem 5. *The unconstrained minimizer $u^*(r,a,t)$ is monotonically nondecreasing with β.*

Proof. We introduce the parameter β as a variable in the functions defined earlier. Observe that the recursive algorithm stated for $W_n(q,r,a,t)$ in the previous section is equivalent to the following recursion (obtained by dividing throughout by β as $\beta > 0$). Initialize $W_0(q,r,a,t,\beta) = 0$. Let $u_n^*(r,a,t,\beta)$ be the value of u that solves the following inequalities,

$$\alpha E_{a,r,t} W_n(u + A, R, A, T, \beta) \le \frac{1}{r} \le \alpha E_{a,r,t} W_n(u + 1 + A, R, A, T, \beta). \tag{6}$$

Furthermore,
- If $q \le u_n^*(r,a,t,\beta)$, $W_{n+1}(q,r,a,t,\beta) = \frac{1}{\beta} + \alpha E_{a,r,t} W_n(q + A, R, A, T, \beta)$.
- If $u_n^*(r,a,t,\beta) < q \le \lfloor tr \rfloor + u_n^*(r,a,t,\beta)$, $W_{n+1}(q,r,a,t,\beta) = \frac{1}{\beta} + \frac{1}{r}$.
- If $q > u_n^*(r,a,t,\beta) + \lfloor tr \rfloor$, $W_{n+1}(q,r,a,t,\beta) = \frac{1}{\beta} + \alpha E_{a,r,t} W_n(q - \lfloor tr \rfloor + A, R, A, T, \beta)$.

Using Lemma 1, it follows from (6) that in order to show that $u_n^*(r,a,t,\beta)$ is monotonically nondecreasing in β, it is enough to show that the function $W_n(q,r,a,t,\beta)$ is nonincreasing in β for all n. We show this by induction. The function $W_0(u,r,a,t,\beta) = 0$. Let $W_n(q,r,a,t,\beta)$ be nonincreasing in β. This implies $E_{a,r,t} W_n(q + A, R, A, T, \beta)$ is nonincreasing in β and $u_n^*(r,a,t,\beta)$ is monotone nondecreasing in β. Now, given (q,r,a,t), the above recursion seen as a function of β is,
- For β where $u_n^*(r,a,t,\beta) + \lfloor tr \rfloor < q$, $W_{n+1}(q,r,a,t,\beta) = \frac{1}{\beta} + \alpha E_{a,r,t} W_n(q + A - \lfloor tr \rfloor, R, A, T, \beta)$.
- For β where $u_n^*(r,a,t,\beta) < q \le \lfloor tr \rfloor + u_n^*(r,a,t,\beta)$, $W_{n+1}(q,r,a,t,\beta) = \frac{1}{\beta} + \frac{1}{r}$.
- For β where $u_n^*(r,a,t,\beta) \ge q$, $W_{n+1}(q,r,a,t,\beta) = \frac{1}{\beta} + \alpha E_{a,r,t} W_n(q + A, R, A, T, \beta)$.

It follows from the definition of the minimizer and (6) that for the domain of β where the first item holds, $\alpha E_{a,r,t} W_n(q + A - tr, R, A, T, \beta) \ge \frac{1}{r}$ and for the domain of β where the third item holds $\alpha E_{a,r,t} W_n(q + A, R, A, T, \beta) \le \frac{1}{r}$. Thus combining this with the hypothesis that $E_{a,r,t} W_n(q + A, R, A, T, \beta)$ is nonincreasing in β implies that $W_{n+1}(q,r,a,t,\beta)$ is nonincreasing in β and the result follows. □

From Theorems 3 and 5 we obtain the following conclusion:

Corollary 1. *The system is indexable.* □

Given a state (q,r,a,t), define the index $\nu(q,r,a,t)$ as the largest value of β for which $u^*(r,a,t,\beta) = q$. It is essentially that value of β where a transition is made from an active action to a passive action in the state (q,r,a,t). It follows from Theorems 3 and 4 that for $r = r_{\max}$, $\nu(q,r,a,t) = \frac{\alpha r_{\max}}{1-\alpha}$. Note that the index is independent of the queue lengths when $r = r_{\max}$.

Lemma 2. *The index associated with the state (q,r,a,t) when the weight is ω, is $\nu(q,r,a,t,\omega) = \omega \nu(q,r,a,t)$.* □

5.6 Index Based Heuristic Policy

The transition probability matrices associated with device i are $P_i^{(r)}$ and $P_i^{(a)}$. Let $\nu_i(q_i, r_i, a_i, t, \omega_i)$ be the index for device i when it is in state (q_i, r_i, a_i, t) and the weight is ω_i. Let $u_i^*(r_i, a_i, t, \beta)$ be the solution in that state for the relaxed problem. Given the state of the system $(\mathbf{q}, \mathbf{r}, \mathbf{a}, t)$, the controller has to decide upon who should send and how much in a frame of duration t seconds. Select a value for β. The amount of data served from user i is $s_i(q_i, r_i, a_i, t, \beta)$. The time taken to transmit this data is $\sum_{i=1}^{M} \frac{s_i(q_i, r_i, a_i, t, \beta)}{r_i}$. This could exceed the frame boundary or fall short of it depending on the choice of β. We know from indexability that for β arbitrary large, the solution $u_i^*(\cdot)$ is infinite and thus $s_i(\cdot)$ is zero implying that the frame time is zero. While for $\beta \to 0$, $s_i(\cdot) \to \min(q_i, \lfloor tr_i \rfloor)$, the frame boundary could exceed depending on the choice of q_i. Since as β decreases, $s_i(q_i, r_i, a_i, t, \beta)$ increases and thus the frame time utilized increases. Thus the controller has to tune β such that the available frame time is maximally utilized or the frame boundary constraint is met. Note that $s_i(q_i, r_i, a_i, t, \beta)$ has only one degree of freedom because fixing β fixes $s_i(\cdot)$ for all i.

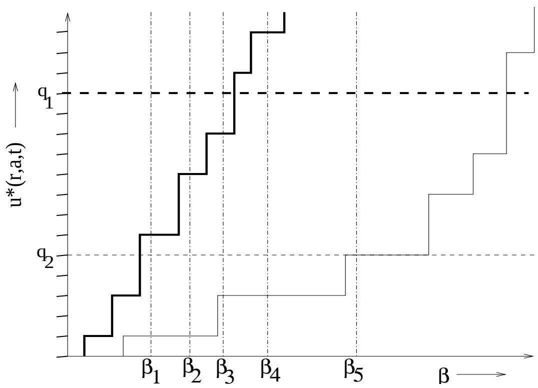

Figure 6. Consider two devices with state $(\mathbf{q}, \mathbf{r}, \mathbf{a}, t)$ with q_1 and q_2 as shown in the adjacent figure. Let the transmission times be the same for each packet. Suppose that a maximum of eight packets can be transmitted in t seconds. The darker staircase function represents $u^*(\cdot)$ for device 1 while the other staircase corresponds to that of device 2. Let s_1 and s_2 be the number of packets that are sent in the frame. At $\beta = \beta_4$, $s_2 = 2$ and $s_1 = 0$; $\beta = \beta_3$, $s_2 = 2$ and $s_1 = 2$; $\beta = \beta_2$, $s_2 = 4$ and $s_1 = 4$; $\beta = \beta_1$, $s_2 = 4$ and $s_1 = 7$. Thus $\beta = \beta_2$ is the solution. Observe that for $\beta > \beta_5$, the number of packets served, s_1 and s_2 are both zero.

The tuning of β is in general not an easy task. But since $u^*(r, a, t, \beta)$ is monotone nondecreasing in β, we have a simpler form for the policy.

Index Policy: *Given the state* $(\mathbf{q}, \mathbf{r}, \mathbf{a}, t)$*, a user with the largest value of* $\nu_i(q_i - 1, r_i, a_i, t, \omega_i)$ *transmits one packet. Let* $j = \arg\max_i \nu_i(q_i - 1, r_i, a_i, t, \omega_i)$*. The state changes to* $(\mathbf{q} - e_j, \mathbf{r}, \mathbf{a}, t)$*, where* e_j *is the unit vector with one at the* j^{th} *entry and rest are all zero. This continues till the frame boundary is exceeded or there is no data in the buffers. The ties are broken probabilistically.*

Consider a case where the rate available for transmission is fixed but it can be different for different devices. Let r_i be the transmission rate for device i. The index policy obtained above will order the transmissions in decreasing order of $\omega_i r_i$ and the one with the highest order transmits till it finishes or the frame boundary is exceeded. Note that this is identical to the well known $c\mu$-rule [8].

5.7 Numerical Results

We assume that there are no voice calls. The discount factor is set to $\alpha = 0.99$ implying that the long term evolution of the queue length process contribute significantly towards the performance measure. The other parameters for the numerical computation of the policy are: the frame time $T = 10$ms, the transmission rate set $\{10, 3.3, 2.5\}$ kbps. We consider two transition probability matrices for the rate process: $P_1 = \{\{0, .5, .5\}; \{.99, .01, 0\}; \{0, .99, .01\}\}$, $P_2 = \{\{0, .5, .5\}, \{.01, 0, .99\}, \{.01, .99, 0\}\}$. For the rate process governed by P_1, with a very large probability the rate increases from one of the lower rates to the next higher rate and then goes to one of the lower rates with equal probability whereas for the rate process governed by P_2, the rate process switches between the two lower rate states with high probability. Thus P_2 resembles a device operating far away from the AP and restricted mobility where as P_1 resembles a device that is highly mobile. The packet arrival process is assumed to independent and identically distributed, on-off $\{0, 40\}$ with probability $\{.5, .5\}$. Since the arrival process is i.i.d. and the frame time available is fixed to T (no voice calls), the policy $u^*(r, a, t, \beta)$ is independent of a and t. Also $u^*(r, a, t, \beta)$ for $r = r_{\max}$ is $\frac{\alpha r_{\max}}{1-\alpha} = 9.9 \times 10^5$. Figure 7 plots u^* vs β for $r = \{3.3, 2.5\}$ kbps and the rate transition probability matrices P_1 and P_2.

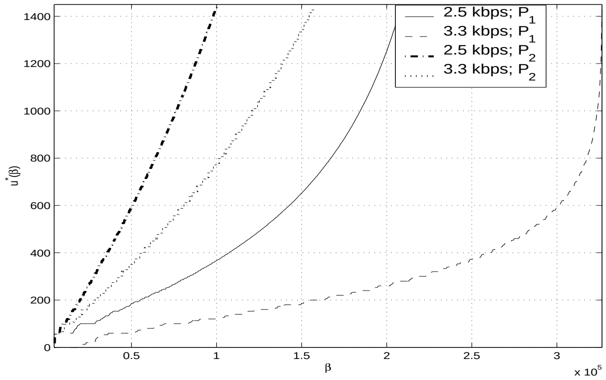

Figure 7. Plots are used for computing indices ν. For example consider two devices with the rate transition probability matrices P_1 and P_2. The weights are 1 for both the devices, $q_1 = q_2 = 600$, $r_1 = 2.5$ and $r_2 = 3.3$ kbps. The indices $\nu_1 = 14.35 \times 10^4$ and $\nu_2 = 8 \times 10^4$. This shows that device 1 has priority over 2 even when $r_2 > r_1$. If one of the device has a rate of 10 kbps, then the service effort is applied to it as much as possible since the index is the largest independent of the queue length.

For the scenario discussed above, we compared the performance of the index policy with that of a round robin policy, a weighted round robin policy that serves three packets of device 2 for each packet of device 1, an index based stabilizing policy with index $\omega_i q_i r_i$ [2]. For a fixed initial state $\mathbf{z} = (\mathbf{q}, \mathbf{r})$ with $q_1 = q_2 = 0$ and $r_1 = r_2 = 2.5$ kbps, the costs $(1-\alpha)V_\alpha(\mathbf{z})$ are 107, 398, 327 and 128 respectively.

6 Conclusion

We have developed index based polling strategies for PCF mode of transmission in 802.11 based wireless LAN. Index policies are always desired for the ease of implementation. The policy is shown to work significantly better than other known policies. As part of future work we are interested in algorithms for on-line computation of the indices and some useful structural results for the index based policy.

References

1. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* Nov. 1997, P 802.11.
2. Andrews M., Kumaran K., Ramanan K., Stolyar A., Vijayakumar R., and Whiting P.: Scheduling in a Queueing System with Asynchronously Varying Service Rates. *Bell Labs Technical Report*, 2000.
3. Berry Randall A., Gallager R. G.: Communication over Fading Channels with Delay Constraints. *IEEE Transaction on Information Theory*, vol. 48, no. 5, 1135-1149, May 2002.
4. Dantzig G. B.: Discrete Variable Extremum problems. *Operations Research*, 5, 1957, 266-277.
5. Goyal Munish, Kumar Anurag, Sharma Vinod: Power constrained and Delay Optimal Policies for Scheduling Transmission over a Fading Channel. *IEEE INFOCOM*, 2003.
6. Jose Nino-Mora: Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, 33(1), 2001, 76-98.
7. Schal M.: Average Optimality in Dynamic Programming with General State Space. *Mathematics of Operations Research*, 18(1), 1993, 163-172.
8. Walrand J.: An Introduction to Queueing Networks. Prentice-Hall, New Jersey, 1988.
9. Weber R. R. and Weiss G.: On an index policy for restless bandits. *J. Appl. Prob.*, 27, 1990, 637-648.
10. Whittle Peter: Restless bandits: Activity allocation in a changing world. In *A Celebration of Applied Probability*, J. Gani (Ed.), *J. Appl. Prob.*, 25A, 1988, 287-298.
11. Yeh E. M. and Cohen Aaron S.: Throughput and Delay Optimal Resource Allocation in Multiaccess Fading Channels. *Proc. of ISIT*, Yokohama, Japan, 2003.

Appendix

Proof of Theorem 2. Since $H(q, r, a, t)$ is a convex combination of $V(q+a, r, a, t)$, it suffices to show that $V(q, r, a, t)$ is convex in q. Consider the value iteration algorithm (5). For $n = 0$, $V_0(q, r, a, t) = 0$ hence convex. Assume $V_{n-1}(q, r, a, t)$ is convex in q. Fix q. Let u_1 and u_2 be the optimal policy for $q - 1$ and $q + 1$.

$$
\begin{aligned}
& V_n(q+1, r, a, t) + V_n(q-1, r, a, t) \\
& = 2q(1 - \frac{\beta}{r}) - \frac{\beta}{r}(u_1 + u_2) + \alpha E_{a,r,t}[V_{n-1}(u_1 + A, R, A, T) + V_{n-1}(u_2 + A, R, A, T)]. \\
& \geq 2q(1 - \frac{\beta}{r}) - \frac{\beta}{r}(u_1 + u_2) + \alpha E_{a,r,t} V_{n-1}(\lfloor \frac{u_1 + u_2}{2} \rfloor + A, R, A, T) + \alpha E_{a,r,t} V_{n-1}(\lceil \frac{u_1 + u_2}{2} \rceil + A, R, A, T). \\
& \geq^* 2V_n(q, r, a, t)
\end{aligned}
$$

where the inequality $(*)$ follows from the fact that the policies $\lfloor \frac{u_1+u_2}{2} \lfloor$ and $\lceil \frac{u_1+u_2}{2} \rceil$ are feasible for the state (q, r, a, t). That the functions are nondecreasing can also be proved along similar lines. □