Server Allocation With Delayed State Observation: Sufficient Conditions For the Optimality of an Index Policy

Navid Ehsan, Mingyan Liu
Electrical Engineering and Computer Science Department
University of Michigan, Ann Arbor
{nehsan,mingyan}@eecs.umich.edu

Abstract

In this paper we study an optimal server allocation problem, where a single server is shared among multiple queues based on the queue backlog information. Due to the physical nature of the system this information is delayed, in that when the allocation decision is made, the server only has the backlog information from an earlier time. This results in imperfect and partial state observation. Queues have different arrival processes as well as different buffering/holding costs to differentiate heterogeneous classes of traffic/service. The objective of the dynamic server allocation is to minimize the expected total discounted holding cost over a finite or infinite horizon. We introduce an index policy where the index of a queue is a function of the state of the queue. More specifically, an index is defined as the one step reward of serving the queue. Our primary interest in this paper is to characterize conditions under which this index policy is optimal. We present a fairly general method bounding the reward of serving one queue instead of another. Using this result, sufficient conditions on the optimality of the index policy can be derived for a variety of arrival processes and packet holding costs (limited to convex functions). These conditions are in general in the form of sufficient separation among indices, and they characterize the part of the state space where the index policy is optimal. We also provide some examples and derive the indices and illustrate the region where the index policy is optimal. Although in this paper we consider the specific scenario of delayed observations, the method itself is quite general and can be extended to many different scenarios as discussed in the paper.

Index Terms

Optimal server allocation, resource allocation, optimization, index policy, delayed state observation, differentiated services, restless bandit, sufficient separation

I. Introduction

The optimal use of available resources is a key element to ensure the efficiency of any system, wireless networks in particular, as resources (e.g., bandwidth) are shared and tend to be rare. A simple way of sharing bandwidth between users is to assign a fixed portion to each user according to some criteria [1]. Although this might be a good solution for constant bit rate systems, it cannot respond to the time-dependent nature of different services [2]. Dynamic bandwidth allocation is therefore preferred as it can adjust allocated resources to each user according to their needs and the amount of resources available at each instant of time, thus increasing the system utilization and quality of service.

In this paper we study a class of bandwidth/resource allocation problems, where allocation decisions are based on partial and delayed information of the system state. Consider the problem of N users/queues competing for shares of a common channel to transmit packets. The channel consists of time frames of a fixed number of M time slots. Each slot is equivalent to one packet transmission time. A bandwidth allocation policy determines which slot to assign to which user within a frame, as shown in Fig. 1. The allocation decision is made once per frame based on backlog information, i.e., instantaneous queue occupancy, given by the users/queues at the beginning of each frame. Due to non-negligible delay (e.g., propagation delay), such information reaches the server only in time for the allocation decision to be made for the *next* frame, by which time the queue occupancies likely have changed due to packet arrivals within the current frame. In other words, the state information is delayed and partially obsolete. This results in possible over-allocation or under-allocation. Thus in this case the allocation needs to take into account unknown random arrivals that occur in between observations or state information updates. Every queued packet incurs a cost at the beginning of each frame, known as the *buffering* or *holding* cost. This cost may depend on the queue-size and may vary from one queue to another, allowing us to consider differentiated service classes, i.e., some queues are more expensive or of a higher priority than others. The objective of the problem is to minimize the total expected discounted cost over a finite or an infinite horizon. While in general reducing holding cost has the effect of reducing packet delay, different forms of the cost function lead to different performance criteria. For example, under a linear cost function equal to all queues (i.e., each packet incurs a constant unit cost) minimizing the cost is equivalent to maximizing system throughput. Similarly, a linear cost function with different unit costs for different queues leads to a weighted throughput criterion.

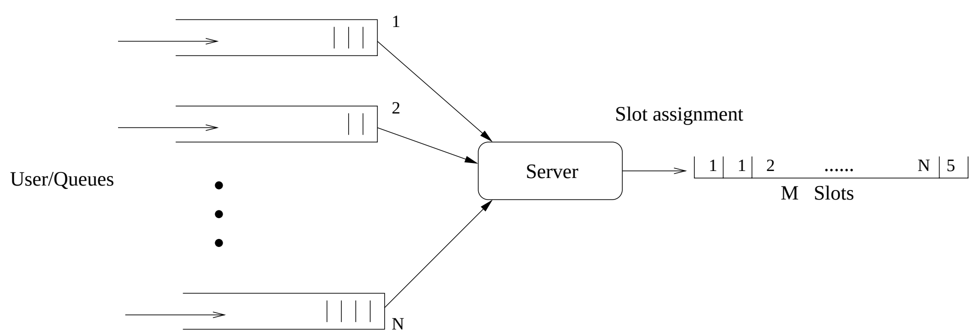

Fig. 1. The bandwidth allocation problem

This optimal bandwidth allocation problem is primarily motivated by wireless communication systems that either have large propagation delay (e.g., in satellite data communication), or where resource allocation is done relatively infrequently compared to packet transmission time, due to cost or design constraint such as energy (e.g., under the IEEE 802.15.4 standard for low-power indoor wireless networks). In the case of a satellite network, users/terminals transmitting to the satellite are assumed to follow a dynamic TDMA schedule, each assigned a certain number of slots within a TDMA frame that consists of a fixed number of slots. Users also inform the satellite their current backlog, carried in packet headers. The assignment is made based on the backlog information and broadcast to the users over a non-interfering channel. An allocation specifies which slot in the upcoming frame is reserved for/to be used by which user. Due to the long propagation delay of the satellite channel (250ms one way), the allocation decision for a particular frame is made based on the backlog information collected during the *previous* frame, which is partially obsolete by the time the allocation is used since by that time the backlog situation may have changed.

In this paper, we will largely focus on a class of policies known as *index policies*. Under an index policy, each user is assigned an *index*, which is a function of its own backlog and/or channel state, and does not depend on other users' states. The index is also updated as the user's state changes.

Definition 1: An index policy is defined as the policy that serves the user with the highest index at each instance of time.

If the indices are chosen properly, the index policy can be shown to be optimal in certain scenarios or under certain conditions [3].

Optimal resource allocation problems under various scenarios have been extensively and intensively studied in the literature. Here we review those most relevant to the one under consideration in this paper. In [4], [5], [6] the problem of assigning a single server to parallel queues with different holding costs was considered, where packet transmissions are successful with a certain probability (or equivalently the transmission time follows a geometric distribution) and that the state information on queue backlogs are perfectly observed. It was shown that the $c\mu$ rule was optimal, where c is the unit holding cost and μ is the probability of transmission success. This can be viewed as an index policy in that the server is always allocated to the non-empty queue with the highest $c\mu$ value, the index. [7], [8], [9] considered the server allocation problem to multiple queues with varying connectivity probability but of the same service class. In each of these papers policies that maximize throughput over an infinite horizon were determined. In particular, [7] derived the sufficient condition for stability and showed that the Longest Connected Queue (LCQ) policy stabilizes the system if the system is stabilizable and that the same policy minimizes the delay in the special case of symmetric queues. The LCQ policy can also be viewed as an index policy in that the index of a queue is defined as the queue size if it is connected and 0 otherwise. [10] further considered a similar problem but with differentiated service classes where different queues have different holding costs, with the objective being to minimize total discounted holding cost over a finite horizon. An interesting result is that the optimality of an index policy does not hold in general, but holds when the indices are sufficiently separated. The intuition, as pointed out in [10] is that due to different holding costs, allocation to shorter but more costly queues (which runs the risk of emptying the queue) is only justified (or compensated) if it is sufficiently more expensive than a longer but less costly queue. [11], [12] studied the stability of power allocation policies. In all of the above work the state of the system, i.e., connectivity and the number of packets in each queue, is precisely known before server allocation is made. This is a major difference between the above cited work and the problem considered here.

[13], [14] considered a server allocation problem with the assumption that the transmission times are asynchronous. [15] considered the problem of routing arriving packets to a set of queues each having its own server. The structures of these problems are quite different from the one examined in this paper and they lead to different solutions.

The problem studied in this paper (in the case of an infinite horizon) can also be cast as a special case of the *restless bandit* problem [16], [17], [18], [19], where projects undergo state transitions even when they are not played or selected. This is because in our case the backlog of each queue continues to change as packets arrive. [16] and [17] studied the asymptotic behavior of this class of problems when the number of projects (queues in this case) and servers (slots in a frame in this case) go to infinity with a fixed ratio. A general optimal solution is not known for this class of problems. However, an index policy can be defined based on the Whittle's heuristic, which is sub-optimal in the finite (number of servers and projects) case and asymptotically optimal in the infinite case.

In [20] we considered a special case of the problem studied here, with single slot allocation ($M = 1$) and the simplification that the holding costs are linear (but differentiated), and that the arrival processes are Bernoulli. We introduced an index policy and showed that similar in spirit to [10], under sufficient separation of the indices the index policy is optimal. We also showed examples where the policy is not optimal when the conditions are not satisfied. Given this prior work, a question naturally arises as to whether a method exists so that similar index policies along with their sufficient optimality conditions may be derived for cases of more general holding cost functions and arrival processes. Motivated by this question, our primary interest in this paper is to develop such a method. We will limit our attention to the special case of allocating frames consisting of a single slot (i.e., $M = 1$) in this paper, resulting in a single server allocation scenario for every allocation period. Further discussion on batch assignment ($M > 1$) can be found in [21].

Remark 1: In this paper we use the term index policy to denote all the policies that satisfy Definition 1. With this definition a greedy policy can be considered as an index policy as will be shown later. In [20], for linear cost functions, we have shown the relation between the greedy policy and Whittle's heuristic indices defined in [16]. For more discussion on the relation between these policies see [3], [20].

In subsequent sections we present a fairly general method that achieves the above goal. We start by bounding the reward of serving one queue instead of another. We then introduce an index policy where the index of a queue is defined as the one step immediate reward of serving the queue, and depends only on its state and arrival statistics. The resulting index policy is greedy or myopic in nature.

Using those bounds, sufficient conditions on the optimality of the index policy are derived for a variety of arrival processes and packet holding costs (limited to convex functions). These sufficient conditions are in the form of sufficient separation among indices, and they characterize the part of the state space where the index policy is optimal. We emphasis that although we have considered a very specific problem scenario in this paper (delayed state information) the method itself is quite general and can be applied in a broader class of server allocation problems (for example it can be used to extend the results in [10] to the more general case of convex cost functions and arbitrary arrival and channel state processes).

The rest of the paper is organized as follows. In the next section we formulate the problem and state our assumptions. Section III summarizes some preliminary results from earlier work. In Section IV we derive the sufficient conditions on the optimality of serving one queue versus another, and apply these conditions to specific examples in Section V. We study these cases for both finite and infinite horizon. In Section VI we introduce an index based on the one step reward and show by using the results from the previous sections that if the indices are sufficiently separated, then the corresponding index policy is optimal. Section VII concludes the paper.

II. Problem formulation and assumptions

In this section we present the network model used as an abstraction of the bandwidth allocation problem described in the previous section, and formally formulate the optimization problem along with a summary of assumptions and notations.

A. Problem Formulation

Consider a set of $\mathcal{N} = \{1, 2, \cdots, N\}$ queues that need to transmit packets to a single server/receiver and compete for shares of a common channel. Time is slotted, and packets are of equal length and one packet transmission time equals one slot time. Transmissions are assumed to be always successful.

M consecutive slots constitute a frame, The bandwidth allocation decision is based on the backlog information of each queue (number of packets waiting/existing in the queue) provided by the queues at the beginning of a frame. We will ignore the transmission time of such information. This is reasonable since one can always increase the frame length with dedicated fixed number of slots at the beginning for the transmission of such information, which does not affect our discussion of optimal allocation. Based on this information an allocation decision is made by the server and broadcast to all queues over a non-interfering channel. This broadcast is received by the queues at the end of that frame, in time to be used for the next frame. The same procedure then repeats, as shown in Fig. 2.

Each user advertises to the server its buffer size at the beginning of the t^{th} frame, denoted by the N-vector $\mathbf{b}_t$, with $b_{i,t}, i \in \mathcal{N}$ being the queue size of queue i at time t. The server allocates slots to be used for transmission in the next time frame, denoted by the N-vector $\mathbf{x}_{t+1}$, with $x_{i,t+1}$ being the allocation to queue i. $0 \leq x_{i,t+1} \leq M, i \in \mathcal{N}$ and $\sum_{i=1}^{N} x_{i,t+1} = M$. This procedure starts from $t = 0$ and ends at $t = T$ (T can be infinite). Note that in this scenario queues do not transmit during the first frame and only start transmitting in the second frame (starting $t = 1$). Similarly, the state information update is not shown for the last frame (starting $t = T - 1$) since the horizon ends at $t = T$ for the finite horizon case.

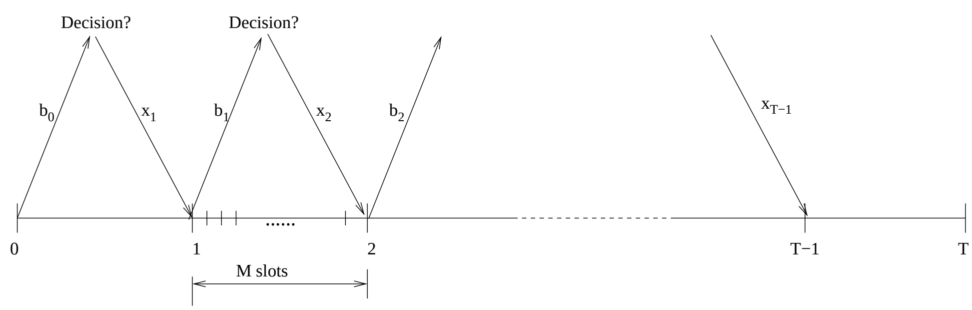

Fig. 2. The bandwidth allocation dynamics

As mentioned before, for the rest of this paper we will only consider the special case where each frame consists of exactly one slot, i.e., $M = 1$, and the allocation decision is made once for every slot. Consequently, $x_{i,t+1} \in \{0, 1\}, i \in \mathcal{N}$ and $\sum_{i=1}^{N} x_{i,t+1} = 1$. For more discussion of multiple slot assignment, see [21].

Let $b_{i,t}$ denote the number of packets in queue i at time t, incurring a cost $c_i(b_{i,t})$ for that time slot, with the total cost of the system being the summation over all queues, i.e., $c(\mathbf{b}_t) = \sum_{i=1}^{N} c_i(b_{i,t})$. We will further assume that functions $c_i(.)$ are non-decreasing and convex. The objective is to find an allocation policy π that minimizes the following cost function:

$$
\begin{aligned}
J_T^{\pi} &= E^{\pi}[C|\mathcal{F}_0], \qquad (1) \\
C &= \sum_{t=1}^{T} \beta^{t-1} \sum_{i=1}^{N} c_i(b_{i,t}),
\end{aligned}
$$

where $\mathcal{F}_0$ summarizes all the information available at time $t = 0$, and $0 < \beta \leq 1$ is the discount factor.

B. Assumptions

Below we summarize important assumptions underlying our network model.

1) We assume that each user has an infinite buffer. Without this assumption we need to introduce penalty for packet dropping/blocking. This is an important extension to work presented here but is out of the scope of this paper and will be considered in a separate study.
2) We assume that if for some i and t we have $x_{i,t} > b_{i,t}$ (which implies $b_{i,t} = 0$ and $x_{i,t} = 1$), then the one slot allocation cannot be used to transmit packets that may have arrived during the t^{th} frame/slot, i.e., within $[t, t+1)$. This is because the exact arrival time of this packet is random, and unless it arrives right before t it cannot be transmitted during that slot.
3) We assume that the arrivals to each queue are mutually independent and they are also independent and identically distributed in each frame. The arrival process statistics are assumed known to the server.
4) The server recalls the latest allocation it has made.
5) We will also adopt the trivial assumption that $\mathbf{x}_0 = \mathbf{0}$ for simplicity of discussion. It does not affect our results on optimal policy and can be easily relaxed in a straightforward way.

C. Notations

We consider time evolution in discrete time steps indexed by $t = 0, 1, \cdots T$, with each increment representing a frame length. Frame t refers to the frame defined by the time interval $[t, t+1)$. In subsequent discussion we will use terms *frames*, *slots*, *steps* and *stages* interchangeably.

As a general rule, boldface letters represent column vectors and normal letters represent scalars/random variables. Whenever we need to distinguish between two policies, we show the policy as a superscript. For example $b^{\pi}_{i,t}$ means the buffer size of the i-th queue at time t under policy π. If w is a scalar, $[w]^+$ takes value w or 0, whichever is greater. For a vector $\mathbf{w}$, we define $[\mathbf{w}]^+$ in the same way for each component.

A list of notations are as follows.

$\mathbf{b}_t = [b_{1,t}, b_{2,t}, \cdots b_{N,t}]'$: The column vector of all queue occupancies at time t.

$\mathbf{x}_t = [x_{1,t}, x_{2,t}, \cdots, x_{N,t}]'$: The number of slots (amount of bandwidth) allocated to users, $x_i(t) \in \{0, 1\}, i = 1, \cdots, N, t = 1, \cdots, T-1$.

$\mathbf{d}_t = [\mathbf{b}_{t-1} - \mathbf{x}_{t-1}]^+$. This value is completely determined from the buffer occupancy and allocation information of the $(t-1)^{th}$ frame. We will call this amount the *existing backlog* since this is the amount carried over from the previous slot due to under-allocation (as opposed to new arrivals occurred during the previous slot). Alternatively we will also call this value the amount of *deterministic packets* to be distinguished from the random arrivals occurred during that frame.

$\mathbf{a}_t = [a_{1,t}, a_{2,t}, \cdots a_{N,t}]'$: The number of packet arrivals during frame t. We sometimes use the notation a_i to denote the random arrivals in queue i during a time frame whenever it does not cause any confusion.

$p_i(k) = \mathbb{P}[a_{i,t} = k]$, $\forall t$ (note that the arrivals are independent and identically distributed in each frame).

$\mathbf{d}_t^{i+} := \mathbf{d}_t + \mathbf{e}_i$ where $\mathbf{e}_i$ is an N-dimensional vector with all entries zero except for a 1 in the i-th position.

$\mathbf{d}_t^{i-} := [\mathbf{d}_t - \mathbf{e}_i]^+$.

$c_i(b_i)$: The holding cost function for having b_i packets in queue i.

$C_t = \sum_{u=t}^{T} \beta^{u-1} \sum_{i=1}^{N} c_i(b_{i,u})$: The cost to go, from time t on (note that $C_1 = C$).

$\mathcal{F}_t$: The σ-field of the information available up to time t.

Remark 2: The information available for making the allocation at time t is the queue occupancy of the previous frame $\mathbf{b}_{t-1}$ and the allocation made earlier, $\mathbf{x}_{t-1}$. This will determine the number of deterministic packets in the buffer at time t, $\mathbf{d}_t$. The total number of packets in the queue at time t is the sum of this deterministic part plus the random arrival during slot $t-1$, i.e.,

$$\mathbf{b}_t = \mathbf{d}_t + \mathbf{a}_{t-1} \ . \tag{2}$$

Separating the queue size into deterministic part and random part will prove convenient in our analysis of the optimal policy.

Given that the server knows the arrival statistics and that the server recalls its last allocation decision, the state of a queue at time t is completely described by its deterministic part, $d_{i,t}$.

III. Preliminaries

In [20] we studied the problem outlined above with the following simplifying assumptions.

(i) The cost is linear for all users, i.e. $c_i(b_{i,t}) = c_i b_{i,t}$ for some $c_i > 0$, $\forall i \in \mathcal{N}$.

(ii) Arrivals are Bernoulli. At each time frame there is one arrival to queue i with probability p_i and no arrival with probability $1 - p_i$.

An index for user i is defined as follows:

$$I_{i,t} = \begin{cases} c_i & \text{if} \quad d_{i,t} > 0, \\ c_i \cdot p_i & \text{if} \quad d_{i,t} = 0. \end{cases} \tag{3}$$

The index defined in (3) is essentially the expected one-step gain obtained by serving queue i. The index policy then allocates the slot to the highest-indexed queue. This policy is myopic/greedy in nature, and is not always optimal in minimizing the objective cost function (1). However, if the indices are sufficiently separated, then this policy is guaranteed to be optimal, as proved in [20] cited below:

Theorem 1: (From [20]) Let the time horizon be T. Suppose that at time t ($1 \leq t \leq T-1$) for some queue i, $I_{i,t} \geq I_{j,t}$, $\forall j \neq i$. We have

1) if $T = 2$, then it is optimal to allocate the slot at time t to queue i;
2) for arbitrary T, if $d_{i,t} > 0$, then it is optimal to allocate the slot at time t to queue i;
3) for arbitrary T, if $d_{i,t} = 0$, then it is optimal to allocate the slot at time t to queue i if for all $j \neq i$ we have:

$$I_{i,t}\left(\frac{1 - (\beta p_i)^{T-t}}{1 - \beta p_i}\right) \geq I_{j,t}\left(\frac{1 - \beta^{T-t}}{1 - \beta}\right) \ . \tag{4}$$

The intuition behind this sufficient condition is that due to the randomness in packet arrivals, which is unobservable at the time of decision making, assigning the slot to an empty queue, rather than another non-empty or empty queue, can be optimal if this queue is sufficiently "costly", so that the gain sufficiently compensates the loss due to potential over-allocating (i.e., a wasted slot if there is no packet arrival and the deterministic part is also zero).

The above intuition is obviously not limited to the specific linear cost and Bernoulli arrival assumptions, as it simply illustrates the assurance one needs in order for a myopic policy to be optimal in minimizing the total cost over a period. Rather it should hold for a more generally defined index policy that aims at minimizing/maximizing the immediate cost/reward.

This motivated us to look for a unifying method with which similar sufficient conditions on an index policy may be derived for more general assumptions on the cost functions and arrival processes [1]. In this paper we relax both assumptions made in [20] and consider general separable convex functions and arbitrary arrival processes. We will also derive the sufficient condition in Theorem 1 as a result of applying this method to the special case of linear cost and Bernoulli arrivals.

IV. Sufficient Conditions on Optimality

In this section we derive a sufficient condition under which serving a particular queue i is optimal. We start by introducing an upper and lower bound on the cost difference in assigning the slot to different queues. We then show how these bounds may be calculated to produce the desired sufficient condition.

A. Sufficient Conditions

We are interested in sufficient conditions under which it is optimal to assign the slot to one queue instead of another. Therefore the key is to find bounds on the cost difference between the two allocations. The following definition characterizes these bounds.

Definition 2: Let π, π' be the optimal policies given the states $\mathbf{d}_{t+1}, \mathbf{d}_{t+1}^{i-}$ respectively. Suppose there exist functions $S_i(\mathbf{d}, u)$ and $R_i(\mathbf{d}, u)$ such that

$$\beta^t S_i(\mathbf{d}_{t+1}, T-t) \leq E^{\pi}[C_{t+1}|\mathbf{d}_{t+1}, \mathcal{F}_{t+1}] - E^{\pi'}[C_{t+1}|\mathbf{d}_{t+1}^{i-}, \mathcal{F}_{t+1}] \leq \beta^t R_i(\mathbf{d}_{t+1}, T-t) \quad \textit{a.s.}, \tag{5}$$

where T is the time horizon. We call function S_i *the lower bound on cost increase for having one more packet in queue* i or in short the *lower bound on queue* i. We call function R_i *the upper bound on cost increase for having one more packet in queue* i or in short *the upper bound on queue* i.

Functions R_i and S_i are not unique. We will focus on those that only depend on the state of queue i, i.e d_i. This is possible, as we will show later via Lemmas 2 and 3.

Suppose at time t we want to allocate the slot to one of the queues. The following lemma compares the allocation to two different queues.

Lemma 1: Let T be the time horizon and suppose $\mathbf{d}_t$ is the state at time t. Let π be the policy that assigns the slot at time t to queue i and assigns optimally thereafter. Let π' be the policy that assigns the slot at t to queue j and assigns optimally thereafter. Suppose there exist functions $R_i(.,.), S_i(.,.), i \in \mathcal{N}$ that satisfy (5). If the following condition holds:

$$E_{\mathbf{a}_{t-1}}[R_j(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t) - S_i(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t)] \leq 0, \tag{6}$$

[1]The techniques used to prove Theorem 1 in [20] took advantage of the simplifying assumptions of linear cost and Bernoulli arrivals.

then we have:

$$E^{\pi}[C_t|\mathbf{d}_t, \mathcal{F}_t] \leq E^{\pi'}[C_t|\mathbf{d}_t, \mathcal{F}_t] \quad a.s..$$

The proof of this lemma can be found in Appendix A. This lemma immediately leads to the following theorem.

Theorem 2: Suppose the state at time t is $\mathbf{d}_t$. Then it is optimal to allocate the slot at t to queue i if the following condition holds:

$$E_{\mathbf{a}_{t-1}}[R_j(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t) - S_i(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t)] \leq 0 \quad \forall j \neq i \ . \tag{7}$$

Proof: This is a direct result of Lemma 1, by comparing allocation to queue i with allocation to all other queues. ■

B. Calculating The Bounds

In this part we present a general method for finding functions $S_i(.,.), R_i(.,.)$ for an arbitrary arrival process. We make the following assumption on the cost function.

Assumption 1: The cost function $c_i(b_i)$ is non-decreasing and convex in b_i.

Definition 3: For any user i and deterministic queue size d_i, define the cost $\hat{c}_i(.)$ to be

$$\hat{c}_i(d_i) = \sum_{a=0}^{\infty} p_i(a) c_i(d_i + a) \ .$$

$\hat{c}_i$ is essentially the expected holding cost of queue i given that the deterministic part is d_i. This definition connects cost as a function of the deterministic queue and cost as a function of the actual buffer occupancy. Note that by Assumption 1, function $\hat{c}_i$ is also non-decreasing and convex in d_i.

Definition 4: For any user i and b_i, $d_i > 0$ define $\alpha_i(b_i)$ and $\hat{\alpha}_i(d_i)$ to be:

$$\begin{aligned} \alpha_i(b_i) &= \inf\{\alpha \in \mathbb{R} | \alpha \geq 0, (c_i(b_i+1) - c_i(b_i)) \leq (1+\alpha)(c_i(b_i) - c_i(b_i - 1))\} \ , \\ \hat{\alpha}_i(d_i) &= \inf\{\hat{\alpha} \in \mathbb{R} | \hat{\alpha} \geq 0, (\hat{c}_i(d_i+1) - \hat{c}_i(d_i)) \leq (1+\hat{\alpha})(\hat{c}_i(d_i) - \hat{c}_i(d_i - 1))\} \ . \end{aligned} \tag{8}$$

The above definitions are introduced primarily for technical reasons. Note that if $c_i(b_i)$ is linear in b_i, then $\hat{c}_i(d_i)$ is linear in d_i (from Definition 3). In this case (linear cost functions) we have $\alpha_i(b_i) = \hat{\alpha}_i(d_i) = 0$, $\forall b_i, d_i \geq 1$. When c_i is strictly convex, then from the above definition we have the following relation between α_i and $\hat{\alpha}_i$:

$$\hat{\alpha}_i(d_i) = \frac{\sum_{a=0}^{\infty} p_i(a)\alpha_i(d_i+a)[c_i(d_i+a) - c_i(d_i+a-1)]}{\sum_{a=0}^{\infty} p_i(a)[c_i(d_i+a) - c_i(d_i+a-1)]} \ . \tag{9}$$

For simplicity of our discussion in the next few lemmas, we further introduce the following random processes.

Let $\{X_{i,t}\}$ be a Markov chain taking values in the set $\mathbb{Z}^+ = \{0, 1, 2, \cdots\}$ with transition probabilities

$$\mathbb{P}(X_{i,t+1} = l | X_{i,t} = k) = p_i(l-k), \tag{10}$$

where $p_i(l-k) = 0$ for all $l < k$. Note that if $X_{i,t_0} = d_{i,t_0}$, then $X_{i,t}$ represents the number of deterministic packets in queue i at time $t \geq t_0$ *if it is never served or allocated a slot.*

Let $\{Y_{i,t}\}$ be a Markov chain taking values in the set $\mathbb{Z}^+$ with transition probabilities

$$\mathbb{P}(Y_{i,t+1} = l | Y_{i,t} = k) = p_i(l-k+1), \quad \forall k \neq 0 \ , \tag{11}$$

where $p_i(l-k+1)=0$ for all $l<k-1$, and $\mathbb{P}(Y_{i,t+1}=0|Y_{i,t}=0)=1$ so that 0 is an absorbing state. Note that if $Y_{i,t_0}=d_{i,t_0}$, then $Y_{i,t}$ represents the number of deterministic packets in queue i at time $t\geq t_0$ *if queue i is assigned every slot until its deterministic part becomes zero.*

Note that the transition probabilities of both processes $X_{i,t}$ and $Y_{i,t}$ are functions of the i-th queue's arrival process.

When $d_i=0$, then it is easy to see that $R_i(\mathbf{d},u)=S_i(\mathbf{d},u)=0$ satisfies (5). In the following lemmas we calculate these bounds when $d_i>0$.

Lemma 2: Let π be the optimal policy given $\mathbf{d}_{t+1}$ where $d_{i,t+1}>0$ and let π' be the optimal policy given $\mathbf{d}_{t+1}^{i-}$. Then we have

$$\begin{aligned}&E^{\pi}[C_{t+1}|\mathbf{d}_{t+1},\mathcal{F}_{t+1}]-E^{\pi'}[C_{t+1}|\mathbf{d}_{t+1}^{i-},\mathcal{F}_{t+1}]\\ \geq\;&\beta^t\sum_{k=1}^{\infty}(\hat{c}_i(k)-\hat{c}_i(k-1))\sum_{u=0}^{T-t-1}\beta^u\mathbb{P}(Y_{i,u}=k|Y_{i,0}=d_{i,t+1})\ .\end{aligned} \tag{12}$$

Proof of this lemma can be found in Appendix B. This result shows that we can find function S_i that satisfies (5) and is a function of the state of queue i, d_i rather than the whole vector $\mathbf{d}$.

Remark 3: Although Lemma 2 gives a good lower bound for the cost difference of starting from state $\mathbf{d}$ rather than $\mathbf{d}^{i-}$, we will use the following bound instead. It is not as tight as that given in Lemma 2, but has a more explicit expression and will be more useful in the examples presented in the next section. Since $d_{i,t+1}>0$, we have:

$$\begin{aligned}&\beta^t\sum_{k=1}^{\infty}(\hat{c}_i(k)-\hat{c}_i(k-1))\sum_{u=0}^{T-t-1}\beta^u\mathbb{P}(Y_{i,u}=k|Y_{i,0}=d_{i,t+1})\\ \geq\;&\beta^t\sum_{k=d_{i,t+1}}^{\infty}(\hat{c}_i(k)-\hat{c}_i(k-1))\sum_{u=0}^{T-t-1}\beta^u\mathbb{P}(Y_{i,u}=k|Y_{i,0}=d_{i,t+1})\\ \geq\;&\beta^t(\hat{c}_i(d_{i,t+1})-\hat{c}_i(d_{i,t+1}-1))\sum_{u=0}^{T-t-1}\beta^u\sum_{k=d_{i,t+1}}^{\infty}\mathbb{P}(Y_{i,u}=k|Y_{i,0}=d_{i,t+1})\\ =\;&\beta^t\Delta\hat{c}_i(d_{i,t+1})\sum_{u=0}^{T-t-1}\beta^u\mathbb{P}(Y_{i,u}\geq d_{i,t+1}|Y_{i,0}=d_{i,t+1})\\ \geq\;&\beta^t\Delta\hat{c}_i(d_{i,t+1})\sum_{u=0}^{T-t-1}\beta^u\mathbb{P}(a_{i,t'}>0,t\leq t'\leq t+u)\\ =\;&\beta^t\Delta\hat{c}_i(d_{i,t+1})\sum_{u=0}^{T-t-1}\beta^u(1-p_i(0))^u\\ =\;&\beta^t\Delta\hat{c}_i(d_{i,t+1})\frac{1-(\beta(1-p_i(0)))^{T-t}}{1-\beta(1-p_i(0))}\ ,\end{aligned} \tag{13}$$

where $\Delta\hat{c}_i(d_i)=\hat{c}_i(d_i)-\hat{c}_i([d_i-1]^+)$. Note that the first and the second inequalities result from the convexity of $\hat{c}_i$ as a consequence of Assumption 1. The third inequality is due to the fact that $\{a_{i,t'}>0,\ t\leq t'\leq t+u\}$ is only one of the events resulting in $Y_{i,u}\geq d_{i,t+1}$ given $Y_{i,0}=d_{i,t+1}$.

Next we want to find function $R_i(\mathbf{d},T)$. We consider non-decreasing and convex cost functions that also satisfy the following assumption.

Assumption 2: $\hat{\alpha}_i(d_i)$ is a non-increasing function of d_i for $d_i>0$ for any queue i.

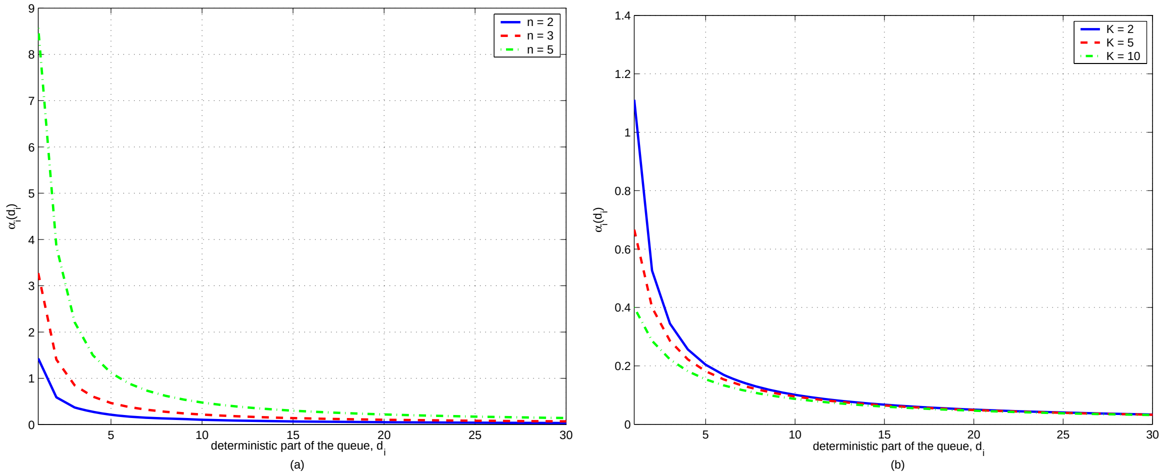

Fig. 3. The effect of cost function and arrival processes on $\hat{\alpha}$ (the Y-axis in both figures is $\hat{\alpha}_i$). (a) $K = 1$ fixed, n variable; (b) $n = 2$ fixed, K variable

A few comments are needed on this additional assumption before we proceed. Firstly, as long as $\hat{c}_i$ does not grow faster than exponential in d_i then Assumption 2 is true. In fact under many commonly used cost functions $\hat{\alpha}_i$ decreases with d_i, e.g., when the cost function is a polynomial of any degree and arrivals in each time frame are finite with probability one. For instance, consider $c_i(b_i) = c_i b_i^n$ where n is a positive integer and assume that packets arrive in batches of K packets. During each frame a batch of K packets arrive with probability p_i and there will be no arrivals with probability $1 - p_i$. In this case we have $\hat{c}_i(d_i) = c_i(1-p_i)d_i^n + c_i p_i (d_i + K)^n$. If we let $c_i = 1$ and $p_i = 0.2$, Figure 3-a plots $\hat{\alpha}_i$ as a function of d_i for different values of n where $K = 1$ is fixed. Figure 3-b plots $\hat{\alpha}_i$ as a function of d_i for different values of K where $n = 2$ is fixed. It can be seen from these figures that $\hat{\alpha}_i$ decays rapidly and approaches zero. Therefore we will also try to find simpler expressions from time to time for a class of convex cost functions where $\hat{\alpha}_i(d_i) \to 0$ as $d_i \to \infty$.

Lemma 3: Let π be the optimal policy given $\mathbf{d}_{t+1}$ where $d_{i,t+1} > 0$ and let π' be the optimal policy given $\mathbf{d}_{t+1}^{i-}$. If Assumptions 1 and 2 hold, then we have:

$$\begin{aligned} & E^{\pi}[C_{t+1}|\mathbf{d}_{t+1}, \mathcal{F}_{t+1}] - E^{\pi'}[C_{t+1}|\mathbf{d}_{t+1}^{i-}, \mathcal{F}_{t+1}] \\ \leq \;& \beta^t(\hat{c}_i(d_{i,t+1}) - \hat{c}_i(d_{i,t+1}-1)) \sum_{u=0}^{T-t-1} \beta^u \sum_{l=0}^{\infty} \mathbb{P}[X_{i,u} = l | X_{i,0} = 0] \cdot (1 + \hat{\alpha}_i(d_{i,t+1}))^l \; . \end{aligned} \tag{14}$$

Proof of this lemma can be found in Appendix C. Again we see from this lemma that there exist function R_i that satisfies (5) and is only a function of the state of queue i, d_i rather than $\mathbf{d}$.

We can find an approximation to the above expression for the case where $\hat{\alpha}$ is small. As $\hat{\alpha} \to 0$ we have $(1+\hat{\alpha})^l \approx 1 + l\hat{\alpha}$. Therefore when α is close to zero we can write:

$$\begin{aligned} & \beta^t(\hat{c}_i(d_{i,t+1}) - \hat{c}_i(d_{i,t+1}-1)) \sum_{u=0}^{T-t-1} \beta^u \sum_{l=0}^{\infty} \mathbb{P}[X_{i,u} = l | X_{i,0} = 0] \cdot (1 + \hat{\alpha}_i(d_{i,t+1}))^l \\ \approx\ & \beta^t \Delta\hat{c}_i(d_{i,t+1}) \sum_{u=0}^{T-t-1} \beta^u \{ \sum_{l=0}^{\infty} \mathbb{P}[X_{i,u} = l | X_{i,0} = 0] + \hat{\alpha}_i(d_{i,t+1}) \sum_{l=0}^{\infty} l \mathbb{P}[X_{i,u} = l | X_{i,0} = 0] \} \\ =\ & \beta^t \Delta\hat{c}_i(d_{i,t+1}) [\frac{1-\beta^{T-t}}{1-\beta} + \hat{\alpha}_i(d_{i,t+1}) \sum_{u=0}^{T-t-1} \beta^u E(X_{i,u} | X_{i,0} = 0)] \ . \end{aligned} \tag{15}$$

To summarize, in this section we derived a sufficient condition under which serving a particular queue i is optimal. This sufficient condition is characterized by upper and lower bounds on the cost difference between serving one of two queues. In the next section we derive these bounds in specific example scenarios.

V. Applications in Specific Scenarios

In the previous section we proved that it is optimal to allocate the slot at time t to queue i if (7) holds. Using Lemmas 2 and 3 we can define functions S_i and R_i that satisfy (5) when $d_i > 0$ as follows (note that we have used the bound defined in Remark 3 to calculate S_i):

$$R_i(\mathbf{d}, u) = \Delta c_i(d_i) \sum_{u'=0}^{u-1} \beta^{u'} \sum_{l=0}^{\infty} \mathbb{P}[X_{i,u'} = l | X_{i,0} = 0] \cdot (1 + \hat{\alpha}_i(d_i))^l \ , \tag{16}$$

$$S_i(\mathbf{d}, u) = \Delta c_i(d_i) \frac{1 - (\beta(1-p_i(0)))^u}{1 - \beta(1-p_i(0))} \ , \tag{17}$$

and when $d_i = 0$ we have $R_i(\mathbf{d}, u) = S_i(\mathbf{d}, u) = 0$.

Remark 4: Note that in this case both functions depend on d_i rather than the whole vector $\mathbf{d}$, i.e. we have $S_i(\mathbf{d}, u) = S_i(d_i, u)$ and $R_j(\mathbf{d}, u) = R_i(d_j, u)$. In this case condition (7) reduces to:

$$\sum_{k=0}^{\infty} p_i(k) S_i(d_{i,t} + k, T-t) \geq \sum_{l=0}^{\infty} p_j(l) R_j(d_{j,t} + l, T-t) \quad \forall j \neq i \ . \tag{18}$$

Therefore for the rest of this paper we will use functions R_i and S_i defined above and use (18) as the sufficient condition for the optimality of allocating to queue i at time t.

For finite T the sufficient condition in (18) depends on T. In this section we apply these results to two scenarios to derive more specific sufficient conditions for the optimality of assigning the slot to a certain queue.

A. Batch and Bernoulli Arrivals

Suppose that the arrivals occur in batches of K packets. During each time frame, with probability p_i there are K arrivals in queue i and with probability $1 - p_i$ there are no arrivals in queue i. Note that $K = 1$ represents the Bernoulli arrival process.

We denote by $S_i^B(d_i)$ and $R_i^B(d_i)$ the lower and upper bounds satisfying (5). We have that $p_i(0) = 1-p_i$, therefore we can calculate $S_i^B(d_i, u)$ (for $d_i > 0$) using (17) as follows:

$$S_i^B(d_i,u) \quad = \quad \Delta\hat{c}_i(d_i)\frac{1-(\beta p_i)^u}{1-\beta p_i} \ . \tag{19}$$

$R_i^B(d_i, u)$ can be calculated (for $d_i > 0$) using (16). We have

$$\begin{aligned} R_i^B(d_i,u) &= \Delta\hat{c}_i(d_i)\sum_{u'=0}^{u-1}\beta^{u'}\sum_{l=0}^{\infty}\mathbb{P}[X_{i,u'}=l|X_{i,0}=0]\cdot(1+\hat{\alpha}_i(d_i))^l \\ &= \Delta\hat{c}_i(d_i)\sum_{u'=0}^{u-1}\beta^{u'}\sum_{h=0}^{u'}\mathbb{P}[X_{i,u'}=h\cdot K|X_{i,0}=0]\cdot(1+\hat{\alpha}_i(d_i))^{h\cdot K} \\ &= \Delta\hat{c}_i(d_i)\sum_{u'=0}^{u-1}\beta^{u'}\sum_{h=0}^{u'}\binom{u'}{h}p_i^h(1-p_i)^{u'-h}(1+\hat{\alpha}_i(d_i))^{h\cdot K} \\ &= \Delta\hat{c}_i(d_i)\sum_{u'=0}^{u-1}\beta^{u'}[(1-p_i)+p_i(1+\hat{\alpha}_i(d_i))^K]^{u'} \\ &= \Delta\hat{c}_i(d_i)\frac{1-[\beta((1-p_i)+p_i(1+\hat{\alpha}_i(d_i))^K)]^u}{1-\beta[(1-p_i)+p_i(1+\hat{\alpha}_i(d_i))^K]} \ . \end{aligned} \tag{20}$$

Using Theorem 2 for Batch arrivals as defined above we have the following result. It is optimal to allocate the packet at time slot t to queue i if

$$\begin{aligned} & p_j R_j^B(d_{j,t}+K, T-t) + (1-p_j)R_j^B(d_{j,t}, T-t) \\ \leq\ & p_i S_i^B(d_{i,t}+K, T-t) + (1-p_i)S_i^B(d_{i,t}, T-t)\ , \qquad \forall j\in\mathcal{N}\ . \end{aligned} \tag{21}$$

By replacing the expressions for S_i^B and R_j^B for this special case we get the following result.

Theorem 3: Let T be the time horizon. Consider batch arrivals where during each time frame queue i has K arrivals with probability p_i and no arrivals with probability $1-p_i$. Suppose the state at time t is $\mathbf{d}_t$, then it is optimal to allocate the slot at time t to queue i if the following two conditions hold.

i) For all $j \neq i$ where $d_{j,t} > 0$ we have,

$$\begin{aligned} & \frac{1-(\beta p_i)^{T-t}}{1-\beta p_i}\{p_i\Delta\hat{c}_i(d_{i,t}+K)+(1-p_i)\Delta\hat{c}_i(d_{i,t})\} \\ \geq\ & \frac{p_j(1-\gamma_j(d_{j,t}+K)^{T-t})\Delta\hat{c}_i(d_{j,t}+K)}{1-\gamma_j(d_{j,t}+K)} + \frac{(1-p_j)(1-\gamma_j(d_{j,t})^{T-t})\Delta\hat{c}_j(d_{j,t})}{1-\gamma_j(d_{j,t})} \quad \text{if } d_{j,t}>0\ . \end{aligned} \tag{22}$$

ii) For all $j \neq i$ where $d_{j,t} = 0$ we have,

$$\frac{1-(\beta p_i)^{T-t}}{1-\beta p_i}p_i\Delta\hat{c}_i(d_{i,t}+K) \geq \frac{1-\gamma_j(d_{j,t}+K)^{T-t}}{1-\gamma_j(d_{j,t}+K)}p_j\Delta\hat{c}_i(d_{j,t}+K)\ ,$$

where $\gamma_j(d_{j,t}) = \beta[(1-p_j)+p_j(1+\hat{\alpha}_j(d_{j,t}))^K]$.

Theorem 3 holds true for all values of T. Therefore one can let $T \to \infty$ to get the following result.

Theorem 4: Consider an infinite horizon and batch arrivals where during each time frame queue i has K arrivals with probability p_i and no arrivals with probability $1-p_i$. Suppose the state at time t is $\mathbf{d}_t$. It is optimal to allocate the slot at time t to queue i if the following two conditions hold.

i) For all $j \neq i$ such that $d_{j,t} > 0$ we have $\gamma_j(d_{j,t}) < 1$ and,

$$\begin{aligned} & \frac{1}{1-\beta p_i}\{p_i \Delta\hat{c}_i(d_{i,t}+K) + (1-p_i)\Delta\hat{c}_i(d_{i,t})\} \\ \geq\ & \frac{p_j \Delta\hat{c}_i(d_{j,t}+K)}{1-\beta[(1-p_j)+p_j(1+\hat{\alpha}_j(d_{j,t}+K))^K]} + \frac{(1-p_j)\Delta\hat{c}_j(d_{j,t})}{1-\beta[(1-p_j)+p_j(1+\hat{\alpha}_j(d_{j,t}))^K]}\ , \end{aligned} \tag{23}$$

ii) For all $j \neq i$ such that $d_{j,t} = 0$ we have $\gamma_j(K) < 1$ and,

$$\frac{p_i}{1-\beta p_i}\Delta\hat{c}_i(d_{i,t}+K) \geq \frac{p_j}{1-\beta[(1-p_j)+p_j(1+\hat{\alpha}_j(d_{j,t}+K))^K]}\Delta\hat{c}_i(d_{j,t}+K)\ .$$

B. Poisson Arrivals

Suppose that arrivals occur according to a Poisson distribution with rate λ_i packets per frame, i.e. we have

$$p_i(k) = e^{-\lambda_i}\frac{\lambda_i^k}{k!}\ .$$

We denote by $S_i^P(d_i)$ and $R_i^P(d_i)$ the lower and upper bounds satisfying (5). We have that $p_i(0) = 1 - e^{-\lambda_i}$, therefore we can calculate $S_i^P(d_i, u)$ using (17) as follows:

$$S_i^P(d_i, u) \;=\; \Delta\hat{c}_i(d_i)\frac{1-(\beta(1-e^{-\lambda_i}))^u}{1-\beta(1-e^{-\lambda_i})}\ . \tag{24}$$

A lower bound for the right hand side of (18) can be calculated as follows. First assume that $d_i > 0$.

$$\begin{aligned} & \sum_{k=0}^{\infty} e^{-\lambda_i}\frac{\lambda_i^k}{k!} S_i^P(d_i+k, T-t) \\ =\ & \frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})}\sum_{k=0}^{\infty}\Delta\hat{c}_i(d_i+k)e^{-\lambda_i}\frac{\lambda_i^k}{k!} \\ \geq\ & \frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})}\Delta\hat{c}_i(d_i)\ , \end{aligned} \tag{25}$$

where the inequality is due to the fact that $\Delta\hat{c}_i$ is non-decreasing as $\hat{c}_i$ is convex.

Similarly when $d_i = 0$ it can be shown that:

$$\sum_{k=0}^{\infty} e^{-\lambda_i}\frac{\lambda_i^k}{k!} S_i^P(d_i+k, T-t) \geq \frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})}(1-e^{-\lambda_i})\Delta\hat{c}_i(1)\ . \tag{26}$$

On the other hand, $R_j^P(.,.)$ can be calculated using (16) as follows:

$$
\begin{aligned}
R_j^P(d_j,u) &= \Delta\hat{c}_j(d_j)\sum_{u'=0}^{u-1}\beta^{u'}\sum_{l=0}^{\infty}\mathbb{P}[X_{j,u'}=l|X_{j,0}=0]\cdot(1+\hat{\alpha}_j(d_j))^l \\
&= \Delta\hat{c}_j(d_j)\sum_{u'=0}^{u-1}\beta^{u'}\sum_{l=0}^{\infty}e^{-\lambda_j u'}\frac{(\lambda_j u')^l}{l!}(1+\hat{\alpha}_j(d_j))^l \\
&= \Delta\hat{c}_j(d_j)\sum_{u'=0}^{u-1}\beta^{u'}e^{\lambda_i u'\hat{\alpha}_j(d_j)} \\
&= \Delta\hat{c}_j(d_j)\frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(d_j)})^u}{1-\beta e^{\lambda_j\hat{\alpha}_j(d_j)}} \ .
\end{aligned}
\tag{27}
$$

Now the left hand side of Equation (18) can be upper bounded as follows, again considering the two cases $d_j > 0$ and $d_j = 0$ separately.

In the case of $d_j > 0$:

$$
\begin{aligned}
&\sum_{l=0}^{\infty}p_j(l)R_j(d_j+l,T-t) \\
=\ & \sum_{l=0}^{\infty}e^{-\lambda_j}\frac{\lambda_j^l}{l!}R_j(d_j+l,T-t) \\
=\ & \sum_{l=0}^{\infty}e^{-\lambda_j}\frac{\lambda_j^l}{l!}\Delta\hat{c}_j(d_j+l)\frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(d_j+l)})^{T-t}}{1-\beta e^{\lambda_j\hat{\alpha}_j(d_j+l)}} \\
\leq\ & \Delta\hat{c}_j(d_j)\frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(d_j)})^{T-t}}{1-\beta e^{\lambda_j\hat{\alpha}_j(d_j)}}\sum_{l=0}^{\infty}e^{-\lambda_j}\frac{\lambda_j^l}{l!}(1+\hat{\alpha}_j(d_j))^l \\
=\ & \Delta\hat{c}_j(d_j)\frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(d_j)})^{T-t}}{1-\beta e^{\lambda_j\hat{\alpha}_j(d_j)}}e^{\lambda_j\hat{\alpha}_j(d_j)} \ ,
\end{aligned}
$$

where the second equality is due to the calculation given in (27), and the inequality is due to the definition of $\hat{\alpha}$ given in Definition 4, and the assumption that $\hat{\alpha}$ is non-increasing.

In the case of $d_j = 0$:

$$
\begin{aligned}
&\sum_{l=0}^{\infty}p_j(l)R_j(d_j+l,T-t) \\
=\ & \sum_{l=0}^{\infty}e^{-\lambda_j}\frac{\lambda_j^l}{l!}R_j(d_j+l,T-t) \\
=\ & \sum_{l=1}^{\infty}e^{-\lambda_j}\frac{\lambda_j^l}{l!}\Delta\hat{c}_j(l)\frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(l)})^{T-t}}{1-\beta e^{\lambda_j\hat{\alpha}_j(l)}} \\
\leq\ & \Delta\hat{c}_j(1)\frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(1)})^{T-t}}{1-\beta e^{\lambda_j\hat{\alpha}_j(1)}}\sum_{l=1}^{\infty}e^{-\lambda_j}\frac{\lambda_j^l}{l!}(1+\hat{\alpha}_j(1))^{l-1} \\
=\ & \Delta\hat{c}_j(1)\frac{(1-(\beta e^{\lambda_j\hat{\alpha}_j(1)})^{T-t})(e^{\lambda_j\alpha_j(1)}-e^{-\lambda_j})}{(1-\beta e^{\lambda_j\hat{\alpha}_j(1)})(1+\alpha_j(1))} \ .
\end{aligned}
$$

The above derivation leads to the following result.

Theorem 5: Let T be the time horizon. Consider Poisson arrivals where during each time frame the number of arrivals to queue i follows a Poisson distribution with mean λ_i. Suppose the state at time t is $\mathbf{d}_t$, then it is optimal to allocate the slot at time t to queue i if the following two conditions hold.

i) For all $j \neq i$ such that $d_{j,t} > 0$ we have,

$$\frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})}\Delta\hat{c}_i(d_i) \geq \frac{1-(\beta e^{\lambda_j\hat{\alpha}_j(d_j)})^{T-t}}{1-\beta e^{\lambda_j\hat{\alpha}_j(d_j)}} e^{\lambda_j\hat{\alpha}_j(d_j)}\Delta\hat{c}_j(d_j) \ .$$

ii) For all $j \neq i$ such that $d_{j,t} = 0$ we have,

$$\frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})}(1-e^{-\lambda_i})\Delta\hat{c}_i(1) \geq \frac{(1-(\beta e^{\lambda_j\hat{\alpha}_j(1)})^{T-t})(e^{\lambda_j\alpha_j(1)}-e^{-\lambda_j})}{(1-\beta e^{\lambda_j\hat{\alpha}_j(1)})(1+\alpha_j(1))}\Delta\hat{c}_j(1) \ .$$

Since Theorem 5 holds true for all T we can let T go to infinity and get the following result.

Theorem 6: Consider an infinite horizon and Poisson arrivals where during each time frame the number of arrivals to queue i follows a Poisson distribution with mean λ_i. Suppose the state at time t is $\mathbf{d}_t$, then it is optimal to allocate the slot at time t to queue i if the following two conditions hold:

i) For all $j \neq i$ such that $d_{j,t} > 0$ we have $\beta e^{\lambda_j\hat{\alpha}_j(d_j)} < 1$ and,

$$\frac{\Delta\hat{c}_i(d_i)}{1-\beta(1-e^{-\lambda_i})} \geq \frac{e^{\lambda_j\hat{\alpha}_j(d_j)}\Delta\hat{c}_j(d_j)}{1-\beta e^{\lambda_j\hat{\alpha}_j(d_j)}} \ .$$

ii) For all $j \neq i$ such that $d_{j,t} = 0$ we have $\beta e^{\lambda_j\hat{\alpha}_j(1)} < 1$ and,

$$\frac{(1-e^{-\lambda_i})\Delta\hat{c}_i(1)}{1-\beta(1-e^{-\lambda_i})} \geq \frac{\Delta\hat{c}_j(1)(e^{\lambda_j\alpha_j(1)}-e^{-\lambda_j})}{(1-\beta e^{\lambda_j\hat{\alpha}_j(1)})(1+\alpha_j(1))} \ .$$

In this section we considered two specific examples and derived the upper and lower bound gain for allocating to a queue. Using these bounds we were able to find sufficient conditions for the optimality of a policy. These conditions although easy to verify, are not simple to interpret. Therefore in the next section we derive alternative sufficient conditions that appear as separation between the one step gain of allocating to a queue.

VI. Optimality of an Index Policy And Examples

In this section we will use the sufficient conditions derived in previous sections to study the optimality of an index policy that is myopic/greedy in nature defined as follows.

Definition 5: The index of queue i at state d_i is defined as the immediate expected reward from assigning the slot to the queue:

$$I_i(d_i) = \sum_{k=0}^{\infty} p_i(k)[\hat{c}_i(d_i+k) - \hat{c}_i([d_i+k-1]^+)] = \sum_{k=0}^{\infty} p_i(k)\Delta\hat{c}_i(d_i+k) \ . \tag{28}$$

Note that $\Delta\hat{c}_i(d_i) = 0$ when $d_i = 0$. The corresponding index policy is defined as one that assigns the slot to the queue with the highest index at each step.

This index policy is a natural one in that it compares the benefit in allocating the next slot to a user based on the expected reward gained in the next time slot. Results from previous sections can be utilized in the following way in the context of this index policy. Theorem 2 gives the sufficient condition under which it is optimal to assign a slot to queue i. By deriving appropriate functions R and S for given arrival process and cost functions, as shown in the previous section, we can obtain sufficient conditions under which the above index policy is optimal. We shall see that this sufficient condition appears as a separation condition in that the index policy is optimal when the highest index is sufficiently larger than the other indices.

Remark 5: The required separation exists only if the loss from not allocating to queue j is bounded for all $j \neq i$. Assumption 2 can be viewed as the condition required for this loss to be bounded.

In the remainder of this section we derive the sufficient conditions under which this index policy is optimal for the two special cases of batch and Poisson arrivals.

A. Batch and Bernoulli Arrivals

Consider the model of batch arrivals. The index in this case is reduced to

$$I_i(d_i) = p_i \Delta \hat{c}_i(d_i + K) + (1 - p_i)\Delta \hat{c}_i(d_i) \ . \tag{29}$$

Using Theorem 3 and the above index definition, we immediately obtain the following result.

Theorem 7: Let T be the time horizon. Consider batch arrivals where during each time frame queue i has K arrivals with probability p_i and no arrivals with probability $1-p_i$. Suppose the state at time t is $\mathbf{d}_t$, then it is optimal to allocate the slot at time t to queue i if both of the following conditions hold.

i) For all $j \neq i$ such that $d_{j,t} > 0$ we have,

$$\frac{1-(\beta p_i)^{T-t}}{1-\beta p_i} I_i(d_{i,t}) \geq \max\{\frac{(1-\gamma_j(d_{j,t}+K)^{T-t})}{1-\gamma_j(d_{j,t}+K)}, \frac{(1-\gamma_j(d_{j,t})^{T-t})}{1-\gamma_j(d_{j,t})}\} I_j(d_{j,t}) \ .$$

ii) For all $j \neq i$ such that $d_{j,t} = 0$ we have,

$$\frac{1-(\beta p_i)^{T-t}}{1-\beta p_i} I_i(d_{i,t}) \geq \frac{(1-\gamma_j(d_{j,t}+K)^{T-t})}{1-\gamma_j(d_{j,t}+K)} I_j(d_{j,t}) \ .$$

Remark 6: Note that for $\alpha_j = 0$ (e.g., when the costs are linear) and $K = 1$ (Bernoulli arrival), $\gamma(\cdot)$ reduces to β. Thus the condition in Theorem 7 reduces to the following. It is optimal to allocate to queue i at time t if for all $j \neq i$ we have:

$$I_i(d_{i,t})(\frac{1-(p_i\beta)^{T-t}}{1-p_i\beta}) \geq I_j(d_{j,t})(\frac{1-\beta^{T-t}}{1-\beta}) \ . \tag{30}$$

This is the same condition derived in [20] for linear cost and Bernoulli arrivals (when $d_{i,t} = 0$) as presented in Section III. In this case with linear cost function $c_i(b_i) = c_i b_i$, the index for queue i reduces to c_i when $d_i \neq 0$, and the index is $p_i c_i$ when $d_i = 0$. Note that when $d_{i,t} > 0$ the above condition is stronger than the one derived in [20], but the weaker sufficient condition there cannot be applied to the more general case of strictly convex cost functions.

Using Theorem 4 and the index definition, we obtain the following result, noting that $\hat{\alpha}_j(d_{j,t}+K) \leq \hat{\alpha}_j(d_{j,t})$.

Theorem 8: Consider an infinite horizon and batch arrivals where during each time frame queue i has K arrivals with probability p_i and no arrivals with probability $1-p_i$. Suppose the state at time t is $\mathbf{d}_t$. It is optimal to allocate the slot at time t to queue i if the following two conditions hold.

i) For all $j \neq i$ such that $d_{j,t} > 0$ we have $\gamma_j(d_{j,t}) < 1$ and:

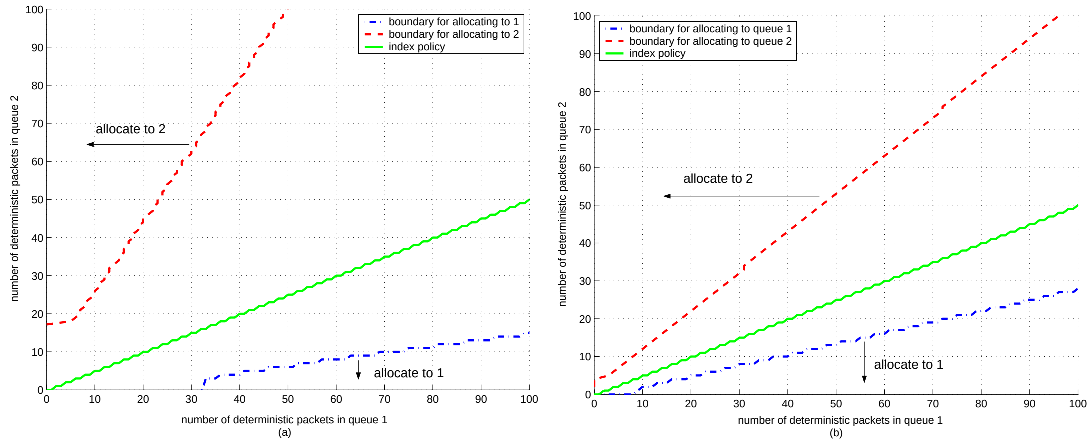

Fig. 4. Required separation between the indices of two queues (a) $\beta = 0.8$, (b) $\beta = 0.6$

$$\frac{1}{1-\beta p_i} I_i(d_{i,t}) \geq \frac{1}{1-\beta[(1-p_j)+p_j(1+\hat{\alpha}_j(d_{j,t}))^K]} I_j(d_{j,t}) \ . \tag{31}$$

ii) For all $j \neq i$ such that $d_{j,t} = 0$ we have $\gamma_j(K) < 1$:

$$\frac{1}{1-\beta p_i} I_i(d_{i,t}) \geq \frac{1}{1-\beta[(1-p_j)+p_j(1+\hat{\alpha}_j(d_{j,t}+K))^K]} I_j(d_{j,t}) \ . \tag{32}$$

Example 1: Consider two queues i, j and let $p_1 = 0.5, p_2 = 0.3, c_1 = 1, c_2 = 2$ and let $c(\mathbf{b}) = c_1 b_1^2 + c_2 b_2^2$. Figure 4 illustrates the separation discussed above with two values of $\beta = 0.6$ and $\beta = 0.8$. Below the dot-dash line is the region where it is optimal to allocate the slot to queue 1 and to the left of the dash line is the region where it is optimal to allocate to the second queue. The solid line shows the boundary determined by the index policy (above the boundary allocate to queue 2 and below the boundary allocate to queue 1).

B. Poisson Arrivals

Consider the case where queue i has a Poisson arrival process with parameter λ_i. The index in this case is

$$I_i(d_i) = \sum_{k=0}^{\infty} p_i(k)\Delta\hat{c}_i(d_i+k) = \sum_{k=0}^{\infty} e^{\lambda_i}\frac{\lambda_i^k}{k!}\Delta\hat{c}_i(d_i+k) \ . \tag{33}$$

Theorem 9: Let T be the time horizon. Consider Poisson arrivals where during each time frame, arrivals to queue i follow a Poisson distribution with rate λ_i. Suppose the state at time t is $\mathbf{d}_t$, then it is optimal to allocate the slot at time t to queue i if for all $j \neq i$ we have

$$\frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})} I_i(d_{i,t}) \geq \frac{1-(\beta e^{\lambda_j \hat{\alpha}_j(d_{j,t}+\xi_{j,t})})^{T-t}}{1-\beta e^{\lambda_j \hat{\alpha}_j(d_{j,t}+\xi_{j,t})}} I_j(d_{j,t}) \ , \tag{34}$$

where $\xi_{j,t} = \begin{cases} 0, & \text{if } d_{j,t} > 0 \\ 1, & \text{if } d_{j,t} = 0 \end{cases}$.

Proof: Let S_i^P and R_i^P be the functions satisfying (5), then by Theorem 2 it is optimal to allocate the next slot to queue i if we have

$$\sum_{k=0}^{\infty} p_j(k) R_j^P(d_{j,t}+k, T-t) \le \sum_{l=0}^{\infty} p_i(l) S_i^P(d_{i,t}+l, T-t), \qquad \forall j \neq i \ . \tag{35}$$

As derived in the previous section we have for $d_j > 0$,

$$\begin{aligned} R_j^P(d_j, u) &= \Delta\hat{c}_j(d_j)\frac{1-(\beta e^{\lambda_j \hat{\alpha}_j(d_j)})^u}{1-\beta e^{\lambda_j \hat{\alpha}_j(d_j)}} \ , \\ S_i^P(d_i, u) &= \Delta\hat{c}_i(d_i)\frac{1-(\beta(1-e^{-\lambda_i}))^u}{1-\beta(1-e^{-\lambda_i})} \ . \end{aligned}$$

Therefore by considering both cases where $d_{j,t} > 0$ and $d_{j,t} = 0$ we have the following results, noting that $\hat{\alpha}_i(\cdot)$ is non-increasing (note also that since $R_j(d_j, u) = 0$ whenever $d_j = 0$, the summation starts from $k = 1$).

$$\sum_{k=0}^{\infty} p_j(k) R_j^P(d_{j,t}+k, T-t) \le \frac{1-(\beta e^{\lambda_j \hat{\alpha}_j(d_{j,t}+\xi_{j,t})})^{T-t}}{1-\beta e^{\lambda_j \hat{\alpha}_j(d_{j,t}+\xi_{j,t})}} I_j(d_{j,t}) \ , \tag{36}$$

$$\sum_{l=0}^{\infty} p_i(l) S_i^P(d_{i,t}+l, T-t) = \frac{1-(\beta(1-e^{-\lambda_i}))^{T-t}}{1-\beta(1-e^{-\lambda_i})} I_i(d_{i,t}) \ . \tag{37}$$

Therefore if (34) holds, then (35) holds, thus completing the proof. ∎

Letting T go to infinity we immediately obtain the next result.

Theorem 10: Let T be the time horizon. Consider Poisson arrivals where during each time frame, arrivals to queue i follow a Poisson distribution with rate λ_i. Suppose the state at time t is $\mathbf{d}_t$, then it is optimal to allocate the slot at time t to queue i if for all $j \neq i$ we have $\beta e^{\lambda_j \hat{\alpha}_j(d_{j,t}+\xi_{j,t})} < 1$ and,

$$\frac{I_i(d_{i,t})}{1-\beta(1-e^{-\lambda_i})} \ge \frac{I_j(d_{j,t})}{1-\beta e^{\lambda_j \hat{\alpha}_j(d_{j,t}+\xi_{j,t})}} \ ,$$

where $\xi_{j,t}$ is the same as defined in Theorem 9.

The above theorems illustrate the required separation between the indices to ensure the optimality of the index policy. It's worth mentioning that even when the index cannot be explicitly derived, the results obtained in previous section (Theorems 5 and 6) can still be applied to find the region where it is optimal to allocate to any of the queues.

We next illustrate the separation between indices via an example in the case of Poisson arrivals. Finding a closed form expression for the index for all cost functions does not seem straightforward. We therefore consider a special cost function.

Example 2: Consider the cost function $c_i(b_i) = c_i b_i^2$. We have:

$$\begin{aligned} \hat{c}_i(d_i) &= c_i \sum_{k=0}^{\infty} p_i(k)(d_i+k)^2 \\ &= c_i[d_i^2 + 2\lambda_i d_i + (\lambda^2 + \lambda)] \end{aligned} \tag{38}$$

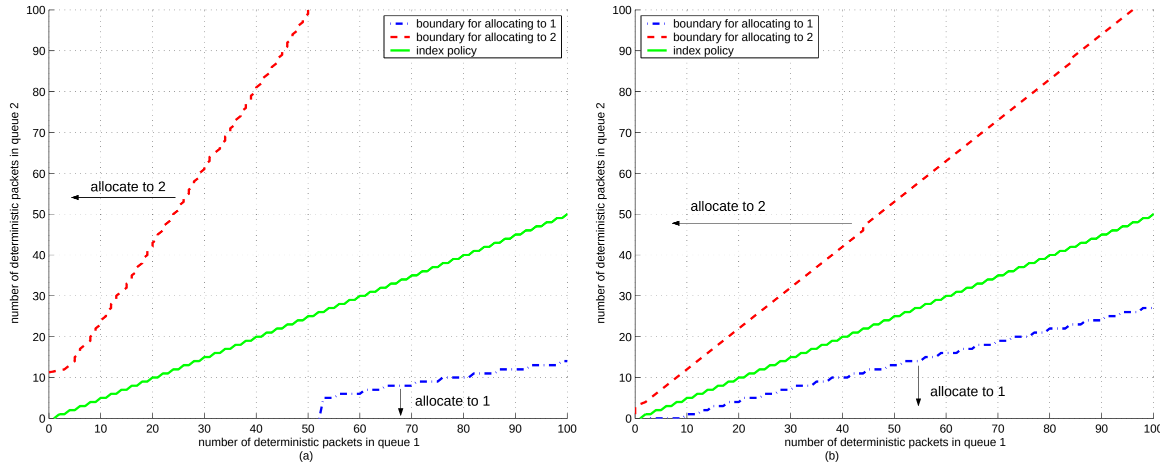

Fig. 5. Required separation between the indices of two queues for Poisson arrivals (a) $\beta = 0.8$, (b) $\beta = 0.6$

The index can be derived as follows. For $d_i \geq 1$ we have

$$
\begin{aligned}
I_i(d_i) &= \sum_{k=0}^{\infty} p_i(k)[\hat{c}_i(d_i+k) - \hat{c}_i(d_i+k-1)] \\
&= c_i \sum_{k=0}^{\infty} p_i(k)(2k + 2d_i + 2\lambda_i - 1) \\
&= c_i[\sum_{k=0}^{\infty} 2kp_i(k) + (2d_i + 2\lambda_i - 1)] \\
&= c_i(4\lambda_i + 2d_i - 1) \ . \qquad (39)
\end{aligned}
$$

For $d_i = 0$ we have

$$
\begin{aligned}
I_i(d_i) &= \sum_{k=1}^{\infty} p_i(k)[\hat{c}_i(d_i+k) - \hat{c}_i(d_i+k-1)] \\
&= c_i \sum_{k=1}^{\infty} p_i(k)(2k + 2d_i + 2\lambda_i - 1) \\
&= c_i[\sum_{k=1}^{\infty} 2kp_i(k) + (2d_i + 2\lambda_i - 1)\sum_{k=1}^{\infty} p_i(k)] \\
&= c_i[2\lambda_i + (2d_i + 2\lambda_i - 1)(1 - e^{-\lambda_i})] \ . \qquad (40)
\end{aligned}
$$

Therefore an index can be defined as $I_i(d_i) = \begin{cases} c_i[4\lambda_i + 2d_i - 1], & \text{if } d_i \neq 0 \\ c_i[2\lambda_i + (2d_i + 2\lambda_i - 1)(1 - e^{-\lambda_i})], & \text{if } d_i = 0 \end{cases}$.

Figure 5 illustrates the separation condition given by Theorem 10 in the infinite horizon case. We have assumed that $c_1(b_1) = b_1^2$ and $c_2(b_2) = 2b_2^2$. Other parameters are $\lambda_1 = 0.5, \lambda_2 = 0.3$, and results are shown for $\beta = 0.6$ and $\beta = 0.8$, respectively. Below the dot-dash line is the region where it is optimal to allocate the slot to queue 1 and to the left of the dash line is the region where it is optimal to allocate to the second queue. The solid line shows the boundary determined by the index policy (above the boundary allocate to queue 2 and below the boundary allocate to queue 1).

The results presented in this section can be summarized as follows. If the one step reward for serving a queue is sufficiently larger than the one step reward of the other queues, then it is optimal to serve the former. Note that this separation is sufficient but not necessary. However characterizing the tightness of these bounds can be very complex, since they depend on the specific cost functions, arrival processes and time horizon considered. For the case of linear cost functions, Bernoulli arrivals and infinite horizon the tightness of such bounds has been discussed in [20].

VII. Discussion and Conclusion

In this paper we considered the problem of sharing a single server among multiple queues when the queue backlog information is one step delayed. The goal for the optimal policy is to minimize the total discounted holding cost over a finite or infinite horizon.

We introduced an index policy that is myopic in nature, and derived sufficient conditions under which it is optimal. This is done by bounding the difference in reward/cost between serving one or the other of any two queues. It is shown that the sufficient condition corresponds to having sufficient separation among the indices. This result is then applied to two specific cases where the arrivals are of the batch and Poisson types.

The frame work established here is fairly general and can be applied to a broad range of queueing systems. For example consider the following server allocation problem. N users compete for a single server. If the server is allocated to the i-th user it can transmit with success probability μ_i. Each user may be connected or disconnected at each instant of time. If the user is disconnected it cannot be served. The queue size and the connectivity of each queue is perfectly observed at any time instant. The objective is to minimize the total discounted cost over a finite horizon. This problem has been studied in [10] for the case of linear cost functions ($c_i(b_i) = c_i b_i$) and it has been shown that in this case the policy that serves the non-empty queue with the highest $c\mu$ is optimal if its index ($c\mu$) is sufficiently larger than the index of all other non-empty queues. In [3] the same problem has been solved using the idea of bounding the rewards (but taking advantage of the linearity of the cost function). We can use the method introduced in this paper to further extend those results to the case of non-linear convex cost functions.

As an extension, it would be interesting to derive similar results for the more general case of restless bandit problems. The results would then help characterize sufficient conditions for the existence of an index policy where the index of each queue is based solely on the property and state of that queue.

Appendix - A

Proof of Lemma 1: We can write,

$$
\begin{aligned}
& E^{\pi}[C_t|\mathbf{d}_t,\mathcal{F}_t] - E^{\pi'}[C_t|\mathbf{d}_t,\mathcal{F}_t] \\
= \; & E_{\mathbf{a}_{t-1}}\{E^{\pi}[C_{t+1}|\mathbf{d}_{t+1} = [\mathbf{d}_t + \mathbf{a}_{t-1} - \mathbf{e}_i]^+, \mathcal{F}_{t+1}] \\
& \qquad - E^{\pi'}[C_{t+1}|\mathbf{d}_{t+1} = [\mathbf{d}_t + \mathbf{a}_{t-1} - \mathbf{e}_j]^+, \mathcal{F}_{t+1}]\} \\
\leq \; & E_{\mathbf{a}_{t-1}}\{E^{\pi}[C_{t+1}|\mathbf{d}_{t+1} = \mathbf{d}_t + \mathbf{a}_{t-1}, \mathcal{F}_{t+1}] - S_i(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t) \\
& \qquad + R_j(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t) - E^{\pi'}[C_{t+1}|\mathbf{d}_{t+1} = \mathbf{d}_t + \mathbf{a}_{t-1}, \mathcal{F}_{t+1}]\} \\
= \; & E_{\mathbf{a}_{t-1}}[R_j(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t) - S_i(\mathbf{d}_t + \mathbf{a}_{t-1}, T-t)] \; .
\end{aligned}
$$

Note that in deriving the inequality, we have used Definition 2. For the last equality we have used the fact that both π and π' are optimal from time $t+1$ on.

Therefore if (7) holds then we have,

$$E^{\pi}[C|\mathbf{d}_t, \mathcal{F}_t] \leq E^{\pi'}[C|\mathbf{d}_t, \mathcal{F}_t] \quad \textit{a.s.},$$

thus proving the lemma. ∎

APPENDIX - B

Proof of Lemma 2: Let π be the optimal policy from time $t+1$ on given $\mathbf{d}_{t+1}$. Define policy $\hat{\pi}$ for the starting condition $\mathbf{d}_{t+1}^{i-}$ as follows. Policy $\hat{\pi}$ assigns slots to the same queue as policy π does under the starting condition $\mathbf{d}_{t+1}$ for every slot. The "best case", in the sense of minimizing the difference in cost between the two policies, is if both policies assign to queue i until (the deterministic part of the queue is) empty. If the slot is assigned to queue i in all the subsequent time intervals (starting with $d_{i,t+1}+1$ deterministic packets in the queue), the number of deterministic packets in queue i at any time $t' \geq t+1$ will have the same distribution as the random process $Y_{i,t'}$, conditioned on $Y_{i,t+1} = d_{i,t+1}+1$. This is true until the deterministic part of the queue hits zero. From that point on both policies π and $\hat{\pi}$ will have the same performance and are both optimal (since π is optimal). Therefore we can write:

$$\begin{aligned}
& E^{\pi}[C_{t+1}|\mathbf{d}_{t+1}, \mathcal{F}_{t+1}] - E^{\pi'}[C_{t+1}|\mathbf{d}_{t+1}^{i-}, \mathcal{F}_{t+1}] \geq E^{\pi}[C_{t+1}|\mathbf{d}_{t+1}, \mathcal{F}_{t+1}] - E^{\hat{\pi}'}[C_{t+1}|\mathbf{d}_{t+1}^{i-}, \mathcal{F}_{t+1}] \\
\geq & \sum_{t'=t+1}^{T} \beta^{t'-1} \sum_{k=1}^{\infty} \mathbb{P}(Y_{i,t'} = k | Y_{i,t+1} = d_{i,t+1}+1) \cdot (\hat{c}_i(k) - \hat{c}_i(k-1)) \\
= & \sum_{k=1}^{\infty} (\hat{c}_i(k) - \hat{c}_i(k-1)) \sum_{t'=t+1}^{T} \beta^{t'-1} \mathbb{P}(Y_{i,t'} = k | Y_{i,t+1} = d_{i,t+1}+1) \\
= & \sum_{k=1}^{\infty} (\hat{c}_i(k) - \hat{c}_i(k-1)) \sum_{u=0}^{T-t-1} \beta^{u+t} \mathbb{P}(Y_{i,t+u+1} = k | Y_{i,t+1} = d_{i,t+1}+1) \\
= & \beta^t \sum_{k=1}^{\infty} (\hat{c}_i(k) - \hat{c}_i(k-1)) \sum_{u=0}^{T-t-1} \beta^{u} \mathbb{P}(Y_{i,u} = k | Y_{i,0} = d_{i,t+1}+1) \ .
\end{aligned} \tag{B-1}$$

The first inequality is due to the fact that the policy $\hat{\pi}$ is not necessarily optimal for the initial state $\mathbf{d}_{t+1}^{i-}$ and the second inequality results from the evolution of the random process Y as the best case. The rest of the equalities are simple algebra, thus proving the lemma. ∎

APPENDIX - C

Proof of Lemma 3: Let π' be the optimal policy given $\mathbf{d}_{t+1}^{i-}$. Define policy $\hat{\pi}$ for the initial state $\mathbf{d}_{t+1}$ as follows. Policy $\hat{\pi}$ assigns the slot to the same queue as policy π' does under the starting condition $\mathbf{d}_{t+1}^{i-}$ at time $t+1$. The "worst case", in the sense of maximizing the difference between the cost of two policies, is that queue i is never served again. Therefore the process for the number of packets in queue i at any time $t' \geq t+1$ (given the initial deterministic packets $d_{i,t+1}+1$) has the same distribution as $X_{i,t'}$ given that $X_{i,t+1} = d_{i,t+1}$. Therefore we can write:

$$\begin{aligned}
& E^{\pi}[C_{t+1}|\mathbf{d}_{t+1},\mathcal{F}_{t+1}] - E^{\pi'}[C_{t+1}|\mathbf{d}^{i-}_{t+1},\mathcal{F}_{t+1}] \leq E^{\hat{\pi}}[C_{t+1}|\mathbf{d}_{t+1},\mathcal{F}_{t+1}] - E^{\pi'}[C_{t+1}|\mathbf{d}^{i-}_{t+1},\mathcal{F}_{t+1}] \\
\leq\ & \sum_{t'=t+1}^{T} \beta^{t'-1} \sum_{k=d_{i,t+1}}^{\infty} \mathbb{P}[X_{i,t'}=k|X_{i,t+1}=d_{i,t+1}] \cdot (\hat{c}(k)-\hat{c}(k-1)) \\
=\ & \sum_{t'=t+1}^{T} \beta^{t'-1} \sum_{l=0}^{\infty} \mathbb{P}[X_{i,t'}=l+d_{i,t+1}|X_{i,t+1}=d_{i,t+1}] \cdot (\hat{c}(l+d_{i,t+1})-\hat{c}(l+d_{i,t+1}-1)) \\
\leq\ & \sum_{t'=t+1}^{T} \beta^{t'-1} \sum_{l=0}^{\infty} \mathbb{P}[X_{i,t'}=l|X_{i,t+1}=0] \cdot (1+\hat{\alpha}_i(d_{i,t+1}))^l (\hat{c}_i(d_{i,t+1})-\hat{c}_i(d_{i,t+1}-1)) \\
\leq\ & \Delta\hat{c}_i(d_{i,t+1}) \sum_{t'=t+1}^{T} \beta^{t'-1} \sum_{l=0}^{\infty} \mathbb{P}[X_{i,t'}=l|X_{i,t+1}=0] \cdot (1+\hat{\alpha}_i(d_{i,t+1}))^l \\
=\ & \beta^t \Delta\hat{c}_i(d_{i,t+1}) \sum_{u=0}^{T-t-1} \beta^u \sum_{l=0}^{\infty} \mathbb{P}[X_{i,u}=l|X_{i,0}=0] \cdot (1+\hat{\alpha}_i(d_{i,t+1}))^l \ .
\end{aligned} \tag{C-1}$$

The first inequality is obtained since $\hat{\pi}$ is not necessarily the optimal policy. The second inequality is due to the fact that we are considering the worst case when policy $\hat{\pi}$ is being used. The next equality is just a change of variables $l = k - d_i$. The third inequality is obtained by using the definition of $\hat{\alpha}_i$, by which we have: $\hat{c}_i(d_i+l)-\hat{c}_i(d_i+l-1) \leq (1+\hat{\alpha}_i(d_i))^l(\hat{c}(d_i)-\hat{c}(d_i-1))$ for $d_{i,t+1} > 0$. ∎

References

[1] D. A. Levine, I. F. Akyildiz, and M. Naghshineh, "The shadow cluster concept for the resource allocation and call admission in atm-based wireless networks," *IEEE/ACM Transaction on Networking*, vol. 5, no. 1, February 1997.

[2] Y. Iraqi and R. Boutaba, "A dynamic bandwidth allocation algorithm for mpeg video sources in wireless networks," *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pp. 86 – 92, 1999.

[3] N. Ehsan and M. Liu, "Server allocation in wireless networks: The use of index policies," in *Combinatorial Optimization in Communication Networks*. 2005, Maggie Cheng, Yingshu Li and Ding-Zhu Du (Eds.), Kluwer Academic Publishers.

[4] J. S. Baras, A. J. Dorsey, and A. M. Makowski, "Two competing queues with linear costs and geometric service requirements: The μc rule is often optimal," *Adv. Appl. Prob.*, vol. 17, pp. 186–209, 1985.

[5] C. Buyukkoc, P. Varaiya, and J. Warland, "The $c\mu$-rule revisited," *Advances in Applied Probability*, vol. 17, pp. 237–238, 1985.

[6] J. G. Shanthikumar and D. D. Yao, "Multi class queueing systems: Polymatroid structure and optimal scheduling control," *Oper. Res.*, vol. 40, pp. 293 – 299, 1992.

[7] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, March 1993.

[8] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1067–73, May 1997.

[9] N. Bambos and G. Michailidis, "On the stationary dynamics of parallel queues with random server connectivities," *Proc. 43th Conference on Decision and Control (CDC)*, pp. 3638–43, 1995, New Orleans, LA.

[10] C. Lott and D. Teneketzis, "On the optimality of an index rule in multi-channel allocation for single-hop mobile networks with multiple service classes," *Probability in the Engineering and Informational Sciences*, vol. 14, no. 3, pp. 259–297, July 2000.

[11] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power allocation and routing in multibeam satellites with time-varying channels," *IEEE/ACM Transactions on Networking, Vol. 11, N0. 1*, pp. 138–152, 2003.

[12] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power and server allocation in a multibeam satellite with time-varying channels," *in Proc. IEEE INFOCOM*, vol. 3, pp. 1451–1460, 2002.

[13] L. Tassiulas and S. Papavassiliou, "Optimal anticipative scheduling with asynchronous transmission opportunities," *IEEE Transactions on Automatic Control*, vol. 40, no. 12, pp. 2052–62, December 1995.

[14] M. Carr and B. Hajek, "Scheduling with asynchronous service opportunities with applications to multiple satellite systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 12, pp. 1820–33, December 1993.

[15] P. Sparaggis, D. Towsley, and C. Cassandras, "Extremal properties of the shortest/longest non-full queue policies in finite-capacity systems with state-dependent service rates," *Journal of Applied Probability*, vol. 30, pp. 233–236, 1993.

[16] P. Whittle, "Restless bandits: Activity allocation in a changing world," *A Celebration of Applied Probability, ed. J. Gani, Journal of applied probability*, vol. 25A, pp. 287–298, 1988.

[17] R. Weber and G. Weiss, "On an index policy for restless bandits," *Journal of Applied Probability*, vol. 27, pp. 637–648, 1990.

[18] J. Nino-Mora, "Restless bandits, patial conservation laws, and indexability," *Advances in Applied Probability, Vol. 33, no. 1*, pp. 76–98, 2001.

[19] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queueing network control," *Mathematics of Operations Research, Vol. 24, No. 2*, pp. 293–305, May 1999.

[20] N. Ehsan and M. Liu, "On the optimality of an index policy for bandwidth allocation with delayed state observation and differentiated services," *Proc. IEEE INFOCOM*, April 2004.

[21] N. Ehsan and M. Liu, "Properties of optimal resource sharing in a satellite channel," *to appear in Proc. IEEE Conference on Decision and Control (CDC)*, December 2004.