

On the Optimality of an Index Policy for Bandwidth Allocation with Delayed State Observation and Differentiated Services

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 the optimality of an index policy for a bandwidth allocation problem, where a single server is allocated among N queues in a slotted system 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 Bernoulli arrival processes with different probabilities of arrival, as well as different buffering/holding costs to differentiate heterogeneous classes of traffic/service. The objective is to minimize the expected total discounted holding cost over a finite or infinite horizon. We introduce an index policy with indices defined as functions of the state of a queue. We first show that when the state of the system is away from the boundary, i.e., no empty queues, the index policy is optimal. When there are empty queues, we show that under sufficient separation of the indices the index policy is still optimal. We show by example that if the separation does not hold, the index policy is not necessarily optimal. We then formulate the optimal bandwidth allocation as a restless bandit problem and show under what conditions the index policy calculated using Whittle's heuristics, which in general is only asymptotically optimal, is optimal for the finite case.

Index Terms—Optimal bandwidth allocation, resource allocation, optimization, index policy, restless bandit, delayed state observation, differentiated services

I. INTRODUCTION

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. In particular, we will examine the optimality of an index policy for one special case of such problems.

Consider the problem of N users/queues competing for a common channel to transmit packets to a single server. The channel consists of time frames of a fixed number of time slots, say M . 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 channel 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 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.

This optimal bandwidth allocation problem is primarily motivated by wireless communication systems that either have large propagation delay, e.g., a satellite data communication scenario, or where resource allocation is done relatively infrequently compared to packet transmission time, due to cost or design constraint. In the case of a satellite network, users/terminals transmit packets to the Network Operating Center (NOC) via the common satellite channel. The data communication link from users to the satellite, also known as the *return channel*, follows a dynamic TDMA schedule. Each user is assigned/allocated a certain number of slots within a TDMA frame that consists of a fixed number of slots. A user can only transmit within its assigned slots during every frame. A user informs the NOC of its current queuing situation (e.g., number of backlogged packets) carried either in packet headers or in a special packet at the beginning of its transmission. The assignment/allocation could be determined either by the satellite or by the NOC, and is broadcast to the users over a *forward channel*, which is separate from (noninterfering with) the return channel. An allocation specifies which slot in the upcoming frame is reserved and to be used by which user. Under such a scenario, due to the long propagation delay of the satellite channel (250 ms from ground/user to satellite and back, or 500 ms from ground/user to ground/NOC via satellite

This work is partially supported by Hughes Network Systems, Germantown, MD.

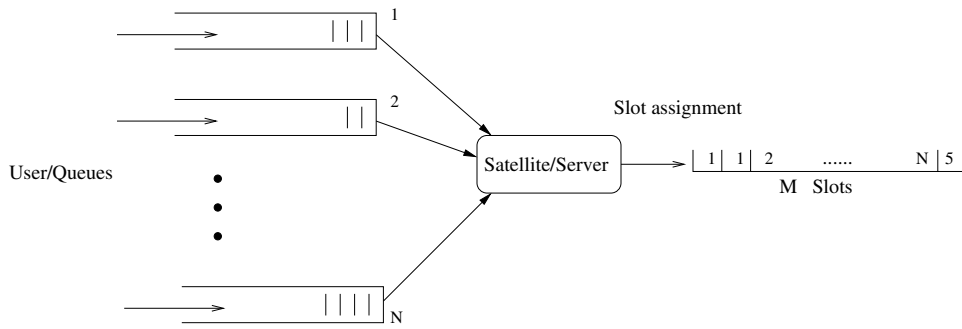


Fig. 1. The bandwidth allocation problem

and back), the allocation decision for a particular frame is made based on the backlog information collected during the *previous* frame, which is delayed and partially “obsolete” by the time the allocation is used since by that time the backlog situation may have changed.

Optimal bandwidth 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 [1] the problem of parallel queues with different holding cost and a single server was considered, where packet transmissions are successful with a certain probability (or equivalently the transmission time follows a geometric distribution) and that we have perfect state information on queue backlogs. It was found that the simple $c\mu$ 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. [2], [3], [4] considered the server allocation problem to multiple queues with varying connectivity probability but of the same service class. Each of them determined policies that maximize throughput over an infinite horizon. In particular, [2] derived the sufficient condition for stability and has shown that the Longest Connected Queue (LCQ) policy stabilizes the system if 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 if not. [5] further considered a similar problem but with differentiated service classes where different queues have different holding cost, 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 only holds when the indices are sufficiently separated. The intuition, as pointed out in [5] is that due to different holding costs, allocation to shorter but more costly queues (which then runs the risk of emptying the queue) is only justified (or compensated) if it is sufficiently more expensive than a longer, less costly queue. All the above cases considered either random queue connectivity or transmission success probability or both, but the state of the system, i.e., connectivity and the number of packets in each queue, is always precisely known when server allocation

is made. This is a major difference between the above cited work and the problem considered here.

[6], [7] considered a server allocation problem with the assumption that the transmission times are asynchronous. [8] 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 multi-armed bandit* problem [9] with multiple plays, where the passive projects undergo state transitions even when they are not selected. This is because in our case the backlog of each queue continuously changes as packets arrive. [9] and [10] studied the asymptotic behavior of this class of problems when the number of arms/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 arms) case and asymptotically optimal in the infinite case.

In [11] we considered a problem similar to the one studied here, with the difference that all queues have the same holding cost and arbitrary but iid arrival processes. We were able to define a class of optimal allocation policies. Considering different holding costs and different arrival processes significantly complicates the situation and it is not clear at this point if a general solution exists. As a first step, in this paper we will only focus on a special case of the outlined problem, with Bernoulli arrivals, i.e., binary, though each queue may have different arrival probabilities, and with only one slot in every frame ($M = 1$), resulting in a single server allocation scenario for every allocation period. We will introduce an index policy of server allocation for this case and examine the conditions under which it is optimal. Extension of work reported in this paper to the more general case of arbitrary arrivals and multiple slot assignment is an important aspect of our ongoing research.

The rest of the paper is organized as follows. In the next section we formulate the problem and state our assumptions. In Section III, we study the optimality of an index policy over a finite horizon. In Section IV we consider the infinite horizon

case and further formulate the problem as a restless bandit problem. We also examine the optimality of indices calculated based on Whittle's heuristic. In Section V we discuss the intuition behind our results and show by examples that if the conditions derived earlier are not satisfied, the index policy is not necessarily optimal. Section VI concludes the paper.

II. PROBLEM FORMULATION AND ASSUMPTIONS

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

A. Problem Formulation

Consider N queues that need to transmit packets to a single server/receiver and compete for shares of a common channel bandwidth. Time is slotted. Packets are of equal length and one packet transmission time equals one slot time. Transmissions are always successful. Packets arrive at each queue as a Bernoulli process. The probability of having a single arrival within a slot at queue i is p_i and the probability of having no arrivals is $1 - p_i$.

As mentioned before, for the rest of this paper we will only consider the case where allocation is done once for each slot, i.e., each frame consists of exactly one slot. 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, with $M = 1$.

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 $b_i(t), i = 1, \dots, N$ 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 . Since there is only one slot, $x_i(t+1) \in \{0, 1\}, i = 1, \dots, N$ and $\sum_{i=1}^N x_i(t+1) = 1$. This procedure starts from $t = 0$ and ends at $t = T$, the finite time horizon. Note that in this scenario during the first frame queues do not have a slot allocated 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 $t = T$.

We assume that the holding cost for queue/user i is c_i . The objective is to find an allocation policy π that minimizes the

following cost function.

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

where \mathcal{F}_0 summarizes all the information available at time $t = 0$, and $\beta < 1$ 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 future study.
- 2) We assume that if for some i and t , $x_i(t) > b_i(t)$ then the one slot allocation cannot be used to transmit the possible packet arrival 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. The arrival probabilities within each slot are also known to the server in making the allocation decision.
- 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 the simplicity of our 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 $t = 0, 1, \dots, T$, with each increment representing a frame length. Frame t refers to the frame defined by the time interval $[t, t+1)$. In subsequent discussions we will use terms *frames*, *slots*, *steps* and *stages* interchangeably.

In general we will use arguments to denote the time index and the subscripts to denote a specific user/queue. For example $b_i(t)$ denotes the buffer occupancy at the beginning of time slot t for the i -th queue. All boldface letters represent column vectors and all normal letters represent scalars/random variables. Whenever we need to distinguish two policies, we show the policy as a superscript. For example $b_i^\pi(t)$ means the buffer size of the i -th queue at time t under policy π .

A list of notations are as follows.

$\mathbf{b}(t) = [b_1(t), b_2(t), \dots, b_N(t)]'$: The column vector of all queue occupancies at time t .

$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)]'$: The number of slots (amount of bandwidth) allocated to users, $x_i(t) \in \{0, 1\}, i = 1, \dots, N, t = 1, \dots, T - 1$.

$\mathbf{d}(t) = [\mathbf{b}(t-1) - \mathbf{x}(t-1)]^+$, where $[w]^+$ takes value w or 0 , whichever is greater. This value is completely determined

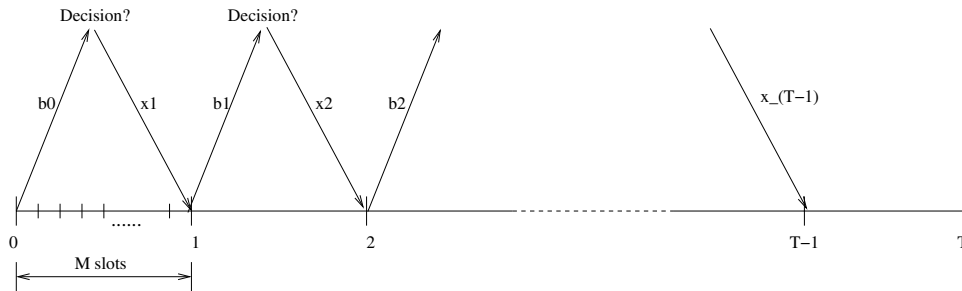


Fig. 2. The bandwidth allocation dynamics

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.

$\mathbf{a}(t) = [a_1(t), a_2(t), \dots, a_N(t)]'$: The number of packet arrivals during frame t , $a_i(t) \in \{0, 1\}$, $i = 1, \dots, N$, $t = 0, \dots, T-1$.

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

p_i : Probability of having one arrival to queue i during each slot. $0 \leq p_i \leq 1$. Probability of having no arrivals is $1 - p_i$.

$C_u = \sum_{t=u}^T \beta^{t-1} \sum_{i=1}^N c_i b_i(t)$: 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 1: 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 $t-1$, therefore we have;

$$\mathbf{b}(t) = \mathbf{d}(t) + \mathbf{a}(t-1) \quad (2)$$

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

III. THE OPTIMALITY OF AN INDEX POLICY

In this section we study the optimality of an index policy for the problem formulated in the previous section. Define the *index* of queue i , denoted by $I_i(t)$, to be:

$$I_i(t) = \begin{cases} c_i & \text{if } d_i(t) > 0, \\ c_i \cdot p_i & \text{if } d_i(t) = 0. \end{cases} \quad (3)$$

Furthermore, we define an *index policy* to be the policy that serves the queue with the highest index at each time step.

The intuition behind this policy seems obvious. It essentially says that when a queue is for sure not empty (if $d_i(t) > 0$ then $b_i(t)$ must be positive), its priority is determined by its unit holding cost, and we should serve the costliest queue

possible. When a queue may be empty (if $d_i(t) = 0$ then $b_i(t) = 0$ with probability $1 - p_i$) on the other hand, its priority is determined by the *expected* holding cost, the original holding cost discounted by the arrival probability. Note here the indices are defined based on the deterministic part of the queue, $d_i(t)$, since we do not know the actual queue size $b_i(t)$.

It turns out that this policy is not always optimal. When the state of the system is away from the boundary, i.e., all queues have non-zero deterministic part, the above index policy is optimal. However, at the boundary, i.e., the deterministic part of some or all of the queues is zero, the index policy is not necessarily optimal. To simplify our presentation, in what follows we will call queue i “empty” at time t if it has a zero deterministic part, i.e., if $d_i(t) = 0$, and “non-empty” otherwise.

Our main result is summarized in the following theorem.

Theorem 1: Let the time horizon be T . Suppose that at time t ($1 \leq t \leq T-1$) for some queue i , $I_i(t) \geq I_j(t) \quad \forall j \neq i$. Then

- 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:

$$I_i(t) \left(\frac{1 - (p_i \beta)^{T-t}}{1 - p_i \beta} \right) \geq I_j(t) \left(\frac{1 - \beta^{T-t}}{1 - \beta} \right) \quad (4)$$

This theorem says that it is always optimal to serve the highest indexed queue, i.e., the index policy is optimal, if this queue is also not empty (deterministically). We consider a system in this state to be *away* from the boundary condition. However, when the state of the system is on the boundary, i.e., queue with the highest index has zero deterministic packets, then the index policy is optimal, if the highest index is sufficiently separated from (larger than) all other indices. This separation is given by (4), and is reminiscent of the separation condition derived in [5]. The intuition behind this sufficient condition is that due to the randomness in packet arrival, which is unobservable at the time of the decision, 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).

Later we show via examples that if this condition is not satisfied then the index policy is not necessarily optimal.

In the following we show these results via a sequence of lemmas.

Lemma 1: Suppose $T = 2$ and $I_i(1) \geq I_j(1)$. Let π be the policy that assigns the slot at time $t = 1$ to queue j and let π' be the policy that assigns the slot to queue i . Then $J_T^\pi \geq J_T^{\pi'}$ a.s..

Proof: Since there is only one allocation period we drop the argument t from $I_i(t)$. Under both policies, the queue size for both queues will be the same at time $t = 1$. Therefore the difference between the costs under the two policies is as follows:

$$\begin{aligned} J_T^\pi - J_T^{\pi'} &= (E^\pi[C_2|\mathcal{F}_1, d_i(1), d_j(1)] \\ &\quad - E^{\pi'}[C_2|\mathcal{F}_1, d_i(1), d_j(1)]) \quad a.s. \\ &= \beta E[c_i b_i^\pi(2) + c_j b_j^\pi(2)] \\ &\quad - \beta E[c_i b_i^{\pi'}(2) + c_j b_j^{\pi'}(2)] \quad a.s. \end{aligned}$$

At time $t = 2$ we have:

$$\begin{aligned} b_i^\pi(2) &= [d_i(1) + a_i(0)] + a_i(1); \\ b_j^\pi(2) &= [d_j(1) + a_j(0) - 1]^+ + a_j(1); \\ b_i^{\pi'}(2) &= [d_i(1) + a_i(0) - 1]^+ + a_i(1); \\ b_j^{\pi'}(2) &= [d_j(1) + a_j(0)] + a_j(1). \end{aligned}$$

We consider the following four cases:

case 1: $d_i(1) > 0, d_j(1) > 0 \Rightarrow I_i = c_i, I_j = c_j$.

$$J_T^\pi - J_T^{\pi'} = \beta \cdot (c_i - c_j) = \beta \cdot (I_i - I_j) \geq 0 \quad a.s. \quad (5)$$

case 2: $d_i(1) = 0, d_j(1) > 0 \Rightarrow I_i = p_i c_i, I_j = c_j$.

$$\begin{aligned} J_T^\pi - J_T^{\pi'} &= p_i \cdot \beta \cdot (c_i - c_j) - (1 - p_i) c_j \beta \\ &= \beta(p_i c_i - c_j) = \beta(I_i - I_j) \geq 0 \quad a.s. \quad (6) \end{aligned}$$

case 3: $d_i(1) > 0, d_j(1) = 0 \Rightarrow I_i = c_i, I_j = p_j c_j$.

$$\begin{aligned} J_T^\pi - J_T^{\pi'} &= p_j \beta (c_i - c_j) + (1 - p_j) \beta c_i \\ &= \beta(c_i - p_j c_j) = \beta(I_i - I_j) \geq 0 \quad a.s. \quad (7) \end{aligned}$$

case 4: $d_i(1) = d_j(1) = 0 \rightarrow I_i = p_i c_i, I_j = p_j c_j$.

$$J_T^\pi - J_T^{\pi'} = \beta(p_i c_i - p_j c_j) = \beta(I_i - I_j) \geq 0 \quad a.s. \quad (8)$$

■

This lemma essentially shows that when the horizon is 2, the index policy is optimal. Under this policy the queue with the highest index should be served, regardless of whether queues are empty or not.

Lemma 2: Let the time horizon be T and suppose $I_i(1) \geq I_j(1)$ and $d_i(1) > 0$. Let π be a policy that serves queue j at time $t = 1$ and then follows an optimal policy given $\mathbf{d}^\pi(2)$ thereafter. Let π' be a policy that serves queue i at time $t = 1$ and then follows an optimal policy given $\mathbf{d}^{\pi'}(2)$ thereafter. Then we have $J_T^\pi \geq J_T^{\pi'}$ a.s..

Proof: Let t' be the first time that π allocates the slot to queue i (let $t' = T$ if π never allocates the slot to queue i).

Define policy $\hat{\pi}$ to be such that it serves queue i at $t = 1$ and serves queue j at $t = t'$ and is the same as policy π everywhere else, i.e., for all $t \neq 1, t'$. We then have:

$$J_T^\pi - J_T^{\hat{\pi}} \geq \begin{cases} \sum_{t=2}^{t'} \beta^{t-1} (c_i - c_j) & \text{if } d_j(1) > 0, \\ \sum_{t=2}^{t'} \beta^{t-1} (c_i - p_j \cdot c_j) & \text{if } d_j(1) = 0. \end{cases} \quad (9)$$

This is because since $d_i(1) > 0$ and queue i is not served in between time slot 1 and t' under both π and $\hat{\pi}$, π costs c_i more than π' for every slot between 2 and t' , both inclusive, for not serving queue i in the first slot. Note that starting $t' + 1$ queue i has the same occupancy under both policies. At the same time, the most π can save on queue j , for serving queue j in the first slot, is c_j for every slot between 2 and t' if queue j starts out non-empty and never empties out between 2 and t' . Thus the equality in (9) in the case $d_j(1) > 0$ is achieved when queue j never becomes empty before t' . If queue j starts out empty, then the most policy π can save is to have an arrival during the first slot and never becomes empty before t' , thus maintaining a difference of $p_j c_j$. Rewriting (9) we have

$$J_T^\pi - J_T^{\hat{\pi}} \geq \sum_{t=2}^{t'} \beta^{t-1} (I_i(1) - I_j(1)) \geq 0 \quad a.s. \quad (10)$$

On the other hand, note that by assumption, after the first slot π' follows the optimal policy. Therefore we must have $J_T^{\hat{\pi}} \geq J_T^{\pi'}$ a.s., thus proving the lemma. ■

This lemma shows that if a non-empty queue has a higher index than other queues (empty or not) at a particular time step, then it is optimal to serve this queue, given that the allocation made for the remaining steps are optimal. Note that we did not specify what the optimal policy is for the remaining steps.

The next question is what if the highest indexed queue happens to be empty. We need the following two lemmas to derive sufficient conditions for the optimality of the index policy for this case.

Lemma 3: Let π be an optimal policy given the initial state \mathbf{d} and let π' be an optimal policy given the initial state \mathbf{d}^{i+} . Then

$$\begin{aligned} &E^{\pi'}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}^{i+}] - E^\pi[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}] \\ &\leq \frac{c_i(1 - \beta^T)}{1 - \beta} \quad a.s. \quad (11) \end{aligned}$$

Proof: π is an optimal policy given the initial state \mathbf{d} . Let $\hat{\pi}$ be a policy defined for the initial state \mathbf{d}^{i+} , that schedules the exact same queues as policy π does for \mathbf{d} . We now compare applying π starting with \mathbf{d} with applying $\hat{\pi}$ starting with \mathbf{d}^{i+} . Since they both schedule the same queue every slot, in the worst case, the latter would end up having one more packet in queue i throughout the entire horizon. Therefore,

$$\begin{aligned} &E^{\hat{\pi}}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}^{i+}] - E^\pi[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}] \\ &\leq \sum_{t=1}^T \beta^{t-1} \cdot c_i = \frac{c_i(1 - \beta^T)}{1 - \beta} \quad a.s. \quad (12) \end{aligned}$$

On the other hand policy $\hat{\pi}$ is not necessarily the optimal policy for the initial state \mathbf{d}^{i+} . Therefore,

$$E^{\hat{\pi}}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}^{i+}] \geq E^{\pi'}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}^{i+}] \quad a.s. \quad (13)$$

Combining the two inequalities (12) and (13) proves the lemma. ■

Lemma 4: Let π be an optimal policy given initial state \mathbf{d} and we have $d_i = 0$. Let π' be an optimal policy given initial state \mathbf{d}^{i+} . Then

$$\begin{aligned} & E^{\pi'}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}^{i+}] - E^\pi[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}] \\ & \geq \frac{c_i(1 - (p_i\beta)^T)}{1 - p_i\beta} \quad a.s. \end{aligned} \quad (14)$$

Proof: π' is an optimal policy given the initial state \mathbf{d}^{i+} . Let $\hat{\pi}$ be a policy defined for the initial state \mathbf{d} , that schedules the exact same queues as policy π' does for \mathbf{d}^{i+} . We now compare applying π starting with \mathbf{d}^{i+} and applying $\hat{\pi}$ starting with \mathbf{d} . Since they both schedule the same queue every slot, in the best case (in the sense of minimizing the cost difference), the former might be able to transmit only from queue i till it empties out (the deterministic part). Therefore the least cost difference is

$$\begin{aligned} & E^{\pi'}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}^{i+}] - E^{\hat{\pi}}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}] \geq \\ & \sum_{t=1}^T \beta^{t-1} p_i^{t-1} \cdot c_i = \frac{c_i(1 - (p_i\beta)^T)}{1 - p_i\beta} \quad a.s. \end{aligned} \quad (15)$$

On the other hand, policy $\hat{\pi}$ is not necessarily optimal for initial state \mathbf{d} . Therefore,

$$E^\pi[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}] \leq E^{\hat{\pi}}[C|\mathcal{F}_0, \mathbf{d}(1) = \mathbf{d}] \quad a.s. \quad (16)$$

Combining the two inequalities (15) and (16) proves the lemma. ■

The next two lemmas give sufficient conditions under which a higher indexed but empty queue should be served, given all subsequent allocations are done optimally.

Lemma 5: Let the time horizon be T and suppose we have $I_i(1) \geq I_j(1)$, $d_i(1) = 0$, and $d_j(1) > 0$. Let π be a policy that serves queue j at time $t = 1$ and then follows an optimal policy given $\mathbf{d}^\pi(2)$ thereafter. Let π' be a policy that serves queue i at time $t = 1$ and then follows an optimal policy given $\mathbf{d}^{\pi'}(2)$ thereafter. Then we have;

$$\begin{aligned} & E^\pi[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \\ & \geq E^{\pi'}[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \quad a.s., \end{aligned} \quad (17)$$

if

$$p_i c_i \frac{(1 - (\beta p_i)^{T-1})}{(1 - \beta p_i)} \geq c_j \frac{1 - \beta^{T-1}}{1 - \beta} \quad (18)$$

Proof: Queue sizes are the same at $t = 1$ under both policies. Given that $d_i(1) = 0$, $d_j(1) > 0$, and that π assigns the slot to queue j at $t = 1$ and π' assigns to queue i , at time $t = 2$ we have

$$\begin{aligned} d_i^\pi(2) &= a_i(0); \\ d_j^\pi(2) &= [d_1(j) + a_j(0) - 1]; \\ d_i^{\pi'}(2) &= [a_i(0) - 1]^+; \\ d_j^{\pi'}(2) &= d_1(j) + a_j(0). \end{aligned}$$

We can see that

$$\begin{aligned} d_i^\pi(2) &= d_i^{\pi'}(2) + 1 \text{ with probability } p_i, \text{ and} \\ d_j^\pi(2) &= d_j^{\pi'}(2) - 1 \text{ with probability one.} \end{aligned}$$

We have

$$\begin{aligned} & E^\pi[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] - \\ & E^{\pi'}[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \\ & = E_{\mathbf{d}^\pi, \mathbf{d}^{\pi'}}\{E^\pi[C_2|\mathcal{F}_1, \mathbf{d}^\pi(2)] - \\ & E[C_2|\mathcal{F}_1, \mathbf{d}^{\pi'}(2)]|d_i(1) = 0, d_j(1) \neq 0\} \quad a.s., \\ & = E_{\mathbf{d}^\pi, \mathbf{d}^{\pi'}}\{p_i \cdot (E^\pi[C_2|\mathcal{F}_1, \mathbf{d}^{\pi'}(2) + e_i] \\ & - E^{\pi'}[C_2|\mathcal{F}_1, \mathbf{d}^{\pi'}(2)]) \\ & + (E^\pi[C_2|\mathcal{F}_1, \mathbf{d}^{\pi'}(2) - e_j] \\ & - E^{\pi'}[C_2|\mathcal{F}_1, \mathbf{d}^{\pi'}(2)])|d_i(1) = 0, d_j(1) \neq 0\} \end{aligned} \quad (19)$$

where $E_{\mathbf{d}^\pi, \mathbf{d}^{\pi'}}$ is the expectation over all values of $\mathbf{d}(2)$ under policies π, π' conditioned on $d_i(1) = 0, d_j(1) \neq 0$ and $E_{\mathbf{d}^{\pi'}}$ is the same expectation for values of $\mathbf{d}(2)$ under policy π' . Now using Lemmas 3 and 4 we have

$$\begin{aligned} & E^\pi[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \\ & - E^{\pi'}[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \\ & \geq \beta(p_i c_i \frac{(1 - (\beta p_i)^{T-1})}{(1 - \beta p_i)} - c_j \frac{1 - \beta^{T-1}}{1 - \beta}) \quad a.s. \end{aligned} \quad (20)$$

It can be seen that if equation (18) is satisfied, then we have;

$$\begin{aligned} & E^\pi[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \\ & \geq E^{\pi'}[C|\mathcal{F}_0, d_i(1) = 0, d_j(1) \neq 0] \quad a.s. \end{aligned}$$

■

Lemma 6: Let the time horizon be T and suppose we have $I_i(1) \geq I_j(1)$ and $d_i(1) = d_j(1) = 0$. Let π be a policy that serves queue j at time $t = 1$ and then follows an optimal policy given $\mathbf{d}^\pi(2)$ thereafter. Let π' be a policy that serves queue i at time $t = 1$ and then follows an optimal policy given $\mathbf{d}^{\pi'}(2)$ thereafter. Then we have

$$\begin{aligned} & E^\pi[C|\mathcal{F}_0, d_i(1) = d_j(1) = 0] \\ & \geq E^{\pi'}[C|\mathcal{F}_0, d_i(1) = d_j(1) = 0] \quad a.s., \end{aligned} \quad (21)$$

if

$$p_i c_i \frac{(1 - (\beta p_i)^{T-1})}{(1 - \beta p_i)} \geq p_j c_j \frac{1 - \beta^{T-1}}{1 - \beta} \quad (22)$$

Proof: The proof proceeds in a similar way to that with Lemma 5. All queue sizes are the same at $t = 1$ under both policies. Since π assigns the slot at time $t = 1$ to queue j and π' assigns it to queue i , at time $t = 2$ we have

$$\begin{aligned} d_i^\pi(2) &= a_i(0); \\ d_j^\pi(2) &= [a_j(0) - 1]^+; \\ d_i^{\pi'}(2) &= [a_i(0) - 1]^+; \\ d_j^{\pi'}(2) &= a_j(0). \end{aligned}$$

Specifically, if there is an arrival to queue i , then $d_i^\pi(2) = 1$ and $d_i^{\pi'}(2) = 0$. If there is an arrival to queue j , then $d_j^\pi(2) = 0$ and $d_j^{\pi'}(2) = 1$.

These combined with Lemmas 3 and 4 gives us

$$\begin{aligned}
& E^\pi[C|\mathcal{F}_0, d_i(1) = d_j(1) = 0] - \\
& E^{\pi'}[C|\mathcal{F}_0, d_i(1) = d_j(1) = 0] \\
= & E_{\mathbf{d}^\pi, \mathbf{d}^{\pi'}}\{E^\pi[C_2|\mathcal{F}_1, \mathbf{d}^\pi(2)] - \\
& E[C_2|\mathcal{F}_1, \mathbf{d}^{\pi'}(2)]|d_i(1) = d_j(1) = 0\} \quad a.s. \\
\geq & \beta p_i c_i \frac{(1 - (\beta p_i)^{T-1})}{(1 - \beta p_i)} - \beta p_j c_j \frac{1 - \beta^{T-1}}{1 - \beta} \quad a.s.,
\end{aligned}$$

where $E_{\mathbf{d}^\pi, \mathbf{d}^{\pi'}}$ is the expectation over all values of $\mathbf{d}(2)$ under policies π, π' conditioned on $d_i(1) = d_j(1) = 0$.

Therefore if Equation (22) is satisfied we have

$$\begin{aligned}
& E^\pi[C|\mathcal{F}_0, d_i(1) = d_j(1) = 0] \\
\geq & E^{\pi'}[C|\mathcal{F}_0, d_i(1) = d_j(1) = 0] \quad a.s.. \quad (23)
\end{aligned}$$

We note the following facts from the above results:

- 1) If (18) is satisfied, then (22) is also satisfied, i.e., (18) is a stronger condition than (22).
- 2) If (18) is satisfied for some horizon T then it is also satisfied for any horizon $T' \leq T$, i.e., the condition becomes weaker and weaker as T decreases. Same applies to (22).
- 3) All the above lemmas remain valid for the allocation of an arbitrary time slot t within the horizon ($1 \leq t \leq T-1$ as opposed to $t = 1$), given $\mathbf{d}(t)$, when T is replaced by $T-t+1$ in Equations (11), (14), (18), and (22). This is because in all these lemmas what matters is the time to go, which is $T-t+1$ for an allocation made at t and a horizon of T .

Combining Lemmas 1, 2, 5, and 6 we are now able to prove Theorem 1 as follows.

Proof of Theorem 1: The case of $T = 2$ is directly given by Lemma 1.

If $d_i(t) > 0$ then it follows directly from Lemma 2 that it is optimal to serve queue i .

Suppose $d_i(t) = 0$ and its index satisfies (4) with respect to all other queues. Consider now some other queue j such that $d_j(t) \neq 0$, and queue k such that $d_k(t) = 0$. Then by Lemma 5 the policy that serves queue i at time t and then follows an optimal policy is at least as good as any policy that serves queue j at time t . Similarly by Lemma 6 the policy that serves queue i at time t and then follows an optimal policy is at least as good as any policy that serves queue k at time t . Thus a policy that serves queue i at time t and then follows an optimal policy is at least as good as any other policy. Therefore serving queue i under (4) when $d_i(t) = 0$ is optimal. ■

Remark 2: Theorem 1 gives a sufficient condition for the optimality of the index rule for one step when the state of the system is on the boundary (the queue with the highest index is empty). A straightforward induction argument shows that if the conditions of Theorem 1 hold in every time step (this requires the indices to be separated as defined by condition (4)), then the index policy is optimal for the entire horizon.

Remark 3: How restrictive the sufficient condition (4) may be, is not immediately clear, as it depends on the horizon T , the discount factor β and the arrival probabilities. It can be evaluated for a specific system with known values of these parameters.

IV. INFINITE HORIZON AND THE RESTLESS BANDIT FORMULATION

A. Infinite Horizon

The optimality condition given in Theorem 1 is time and horizon dependent. A fairly straightforward extension of this result to the infinite horizon case gives us the following theorem as $T \rightarrow \infty$.

Theorem 2: Consider an infinite horizon. Suppose that at time t for some queue i , we have $I_i(t) \geq I_j(t) \quad \forall j \neq i$. Then

- 1) if $d_i(t) > 0$, then it is optimal to allocate the slot at time t to queue i .
- 2) if $d_i(t) = 0$, then it is optimal to allocate the slot at time t to queue i if for all $j \neq i$,

$$\frac{I_i(t)}{1 - p_i \beta} \geq \frac{I_j(t)}{1 - \beta}. \quad (24)$$

Theorem 2 gives a sufficient condition for the optimality of the index policy over an infinite horizon.

B. Calculating the Indices for a Restless Bandit Problem

In what follows we will consider the infinite horizon problem and formulate it as a restless bandit [9], [10]. Consider the problem of allocating M slots to N queues where $M < N$ and each queue can be allocated at most one slot per allocation period (frame). We want to find the optimal policy that minimizes the total expected discounted cost (equation (1)) over an infinite horizon. The problem studied in previous sections is a special case ($M = 1$) of this more general formulation. Whittle in [9] showed that by relaxing the condition of serving exactly M queues within each frame to that of serving an *average* of M queues per frame over the horizon, the problem can be separated into N one-dimensional problems and an index can be calculated for each queue, if the indices satisfy some indexability condition. The policy then would be to serve the queue with the highest index. For more details see [9], [12], [13].

It can be shown that our problem satisfies Whittle's indexability property and the index is calculated as follows. Each queue will be considered individually. A *subsidy* is given to taking the passive action on queue i (not allocating the slot to the queue). In our problem this is equivalent to charging user i an amount γ if a slot is allocated. The index then is the amount of subsidy/charge such that allocating and not allocating the slot to queue i are equally optimal. Below we proceed to calculate the index for an individual user as a function of its state d . Since only one queue is under consideration, we will use similar notations as before but suppress all subscripts.

Again arrivals will be assumed to be of a Bernoulli type within each frame, with an arrival probability p . In each time

frame we have the option to either allocate a single slot to the queue or not. If we choose to allocate a slot, the cost for this allocation is γ . The unit holding cost for the queue for each packet during a time frame is c . The information available when the decision is being made for allocation at time t is $d(t)$, the number of deterministic packets in the queue. Let $v(t) = 1$ if the slot is allocated for time frame t and let $v(t) = 0$ otherwise. The objective is to find a policy π that minimizes the the following cost function over an infinite horizon.

$$J_{\infty}^{\pi} = E^{\pi} \left[\sum_{t=1}^{\infty} \beta^{t-1} \cdot R(b(t), v(t)) | \mathcal{F}_0 \right] \quad (25)$$

where

$$R(b(t), v(t)) = \begin{cases} c \cdot b(t) + \gamma & v(t) = 1, \\ c \cdot b(t) & v(t) = 0. \end{cases} \quad (26)$$

Define $d(t)$ to be the state of the system at time t . The dynamic programming formulation to this problem is as follows:

$$\begin{aligned} & V(0) \\ = & cp + \min\{\gamma + \beta V(0), \beta(pV(1) + (1-p)V(0))\}; \\ & V(d) \\ = & c(d+p) + \min\{\gamma + \beta(pV(d) + (1-p)V(d-1)), \\ & \beta(pV(d+1) + (1-p)V(d))\} \quad d \geq 1, \end{aligned} \quad (27)$$

where $V(d)$ is the value function, or cost-to-go given state d .

Definition 1: An index at state d , $\gamma^*(d)$ is defined as follows. $\gamma^*(d)$ is the supremum of all γ for which it is optimal to allocate the slot when there are d deterministic packets in the queue. Alternatively, it is also the γ value for which it is equally optimal to assign and not assign the slot.

Using the two lemmas from the Appendix we can obtain the following result:

$$\begin{aligned} \gamma^*(0) &= \frac{\beta pc}{1-p\beta}; \\ \gamma^*(d) &= \frac{\beta c}{1-\beta}, \quad d \geq 1. \end{aligned} \quad (28)$$

Whittle's heuristic index policy is to allocate the slot to the queue with the highest index. This index policy is asymptotically optimal for N and M going to infinity with fixed ratio, and is only sub-optimal in general for finite cases ($M = 1$ in this case).

Here we point out the relationship and differences between the Whittle's heuristic index policy and our index policy for infinite horizon in Theorem 2.

- 1) Comparing (28) with (3) we see that when the state of the system is away from the boundary ($d_i(t) \neq 0, \forall i$), Whittle's index policy coincides with our index policy, and is optimal by Theorem 2. The optimal policy is simply to serve the queue with the highest cost.
- 2) When $d_i(t) = 0$ for some queue i , then under Whittle's index policy the slot will be allocated to i if and only if $\gamma_i \geq \gamma_j, j \neq i$. As we will show in the next section,

in this case Whittle's index policy is not necessarily optimal.

V. EXAMPLES AND DISCUSSIONS

In this section we show via two examples, both over infinite horizon, that the Whittle's index policy is not necessarily optimal for the problem under consideration, and that if Equation (24) is not satisfied, our index policy is also not necessarily optimal.

Example 1: Suppose we only have two queue, with $c_1 = 10, c_2 = 7, p_1 = 0.8, p_2 = 1, \beta = 0.9$. Suppose $d_1(t) = 0, d_2(t) \neq 0$. We need to determine whether the slot should be allocated to queue 1 or 2 for slot t , given all subsequent slots will be allocated optimally.

One can easily check that in this example we have $\gamma_1 < \gamma_2$, and $I_1 > I_2$. Therefore, under the Whittle's index policy, the slot should be assigned to queue 2, whereas under our index policy the slot should be assigned to queue 1. Also note that $\frac{I_1}{1-p_1\beta} < \frac{I_2}{1-\beta}$, thus the sufficient condition (24) is not satisfied. As we show below, the optimal decision is indeed to allocate the slot to queue 2. Therefore our index policy is not optimal in this example where (24) is not satisfied.

Let π be the policy that allocates the slot at time t to queue 1 and follows the optimal policy afterwards. Let π' be the policy that allocates the slot at time t to queue 2 and follows the optimal policy afterwards. We show that $J_{\infty}^{\pi'} \leq J_{\infty}^{\pi}$ a.s. as follows.

Policy π allocates the slot at t to queue 1 and follows an optimal policy afterward. Let t' be the first (random) time that policy π allocates to queue 2 (let $t' = \infty$ if policy π never serves queue 2 a.s.).

Define $\hat{\pi}$ to be the following policy. At time t it allocates the slot to queue 2. Afterwards we have one of the following cases:

1. If $a_1(t-1) = 0$, then from time $t+1$, $\hat{\pi}$ allocate to the same queue as policy π . In this case, since $p_2 = 1$, there will always be one more packet in queue 2 when policy π is used compared to policy $\hat{\pi}$.

2. If $a_1(t-1) = 1$, then $\hat{\pi}$ starts allocating subsequent slots to queue 1. If $d_1(t) = 0$ for some time $t_1 < t'$, then from t_1+1 , $\hat{\pi}$ starts allocating to the same queue as policy π (note since there are only two queues, $t_1 < t'$ necessarily means that π allocates all slots to queue 1 as well until t' , therefore the two policies essentially have the same allocations). In this case after t_1 there will always be one more packet in queue 2 when π is used. If $d_1(t)$ has not become empty by time t' , then π' allocates the slot at t' to queue 1, and follow the same allocation as π does afterwards (note that from $t'+1$ on queues will be in the same state under both policies).

We then have

$$\begin{aligned} J_\infty^\pi - J_\infty^{\hat{\pi}} &= \beta^t(1-p_1) \cdot \frac{c_2}{1-\beta} - \\ &\quad \beta^t p_1 \cdot \left(\sum_{u=t+1}^{t'} \beta^{u-t-1} p_1^{u-t-1} (c_1 - c_2) \right) \\ &\geq \beta^t \left[\frac{(1-p_1)c_2}{1-\beta} - \frac{p_1(c_1-c_2)}{1-\beta p_1} \right]. \end{aligned} \quad (29)$$

The first term on the right hand side corresponds to $a_1(t-1) = 0$ and the second term to $a_1(t-1) = 1$. If we have

$$\frac{(1-p_1)c_2}{1-\beta} \geq \frac{p_1(c_1-c_2)}{1-\beta p_1}, \quad (30)$$

then $J_\infty^\pi \geq J_\infty^{\hat{\pi}}$. It can be easily verified that the data in Example 1 satisfies (30). On the other hand policy $\hat{\pi}$ is not necessarily optimal after t . So we have $J_\infty^{\pi'} \leq J_\infty^{\hat{\pi}}$.

Therefore in Example 1 it is optimal to serve queue 2.

In the next example we consider a variation of Example 1 which has a different solution.

Example 2: Consider the same two queues as above, and suppose there are $N-2$ other queues with $c_i = 11$, $p_i = 0.1$. Also assume that at time t we have $d_1(t) = 0$, $d_2(t) \neq 0$, and $d_i(t) = 0$ $i \neq 1, 2$. We need to determine which queue to allocate for slot t .

In this example, we can easily verify that $I_1, I_2 > I_i, i \neq 1, 2$, $\gamma_1, \gamma_2 > \gamma_i, i \neq 1, 2$, and that I_1 and I_2 satisfies (24) w.r.t all other queues, respectively. Therefore under both index policies we should allocate the slot to either queue 1 or queue 2. In particular, since the parameters on queue 1 and queue 2 did not change from Example 1, under the Whittle's index policy the slot should be assigned to queue 2.

However, as $N \rightarrow \infty$ the probability of having a packet in one of the queues $i \neq 1, 2$ approaches one. Since their cost is greater than c_1 and c_2 , the probability that either queue 1 or queue 2 being served again approaches zero. This is because with probability 1 there will be a non-empty queue $i \neq 1, 2$ having the highest index, and by Theorem 2 it is optimal to serve this queue. Thus in this case, it is optimal to allocate the slot to the first queue since $p_1 c_1 \geq c_2$.

To summarize, Example 1 shows that when the sufficient condition (24) is not satisfied, our index policy is not necessarily optimal. Although the example is for infinite horizon, one can construct a similar finite horizon example to show the same (e.g., by having very large T in Example 1). Example 2 shows that the Whittle's index policy is also not necessarily optimal. In addition, combining these two examples we see that when the queue with the highest index is empty and (24) is not satisfied, then the optimal policy may not be determined by indexing a queue based on its own state (i.e., an "index", if exists, may depend on all other queues in the system).

Note that the above examples are constructed to show that the two index policies are not necessarily optimal. These examples may or may not reflect certain practical scenarios.

VI. CONCLUSION AND FUTURE WORK

In this paper we studied the optimality of an index policy for allocating a single server to N parallel queues, when the queue size is not perfectly observed, the arrivals are Bernoulli and the services are differentiated. We derived sufficient conditions for the index policy to be optimal, for both the finite horizon and infinite horizon cases. We also show by examples that when the sufficient condition is not satisfied, this index policy is not necessarily optimal.

We then compared our results to Whittle's heuristic index policy derived for the same problem over an infinite horizon. Although Whittle's index heuristic is asymptotically optimal when number of allocation slots and the number of queues go to infinity, it is not necessarily optimal for the finite case. We showed when the Whittle's index policy is optimal in this problem, and showed via examples that it is not necessarily optimal when the sufficient condition is not satisfied.

Our future work will focus on generalizing these results into the case of arbitrary arrivals and multiple allocation slots.

APPENDIX

The following two lemmas demonstrate that the indices shown in Equation (28) are indeed the Whittle's indices for the restless bandit problem over an infinite horizon. Consider the single queue scenario described in section IV.B with the holding cost c , arrival probability p and a charge for allocating a slot γ .

Lemma A-1: Suppose $d(t) \neq 0$. If $\gamma \leq \frac{\beta c}{1-\beta}$, then it is optimal to allocate the slot for the next time frame and if $\gamma \geq \frac{\beta c}{1-\beta}$ then it is optimal not to allocate the slot for the next time frame. In case of equality it is equally optimal to allocate or not to allocate the slot.

Proof: Let π be the policy that allocates a slot for time t and then allocates optimally. Let π' be the policy that does not allocate the slot for t and then allocates optimally. Let $t' > t$ be the first time that π' allocates a slot.

Define $\hat{\pi}$ to be the following policy. $\hat{\pi}$ allocates a slot at t and then does not allocate any slots for all time frames less than or equal t' . At $t'+1$ the queue will be in the same state whether under $\hat{\pi}$ or under π' . Therefore,

$$\begin{aligned} J_\infty^{\pi'} - J_\infty^{\hat{\pi}} &= \beta^{t'-1} \gamma - \beta^{t-1} \gamma + \sum_{u=t+1}^{t'} \beta^{u-1} c \\ &= \beta^{t-1} \left\{ \frac{\beta c (1 - \beta^{t'-t})}{1-\beta} - \gamma (1 - \beta^{t'-t}) \right\} \end{aligned} \quad (A-1)$$

It can be seen that if $\gamma \leq \frac{\beta c}{1-\beta}$, then we have $J_\infty^{\pi'} \geq J_\infty^{\hat{\pi}}$. Also note that we have $J_\infty^\pi \leq J_\infty^{\hat{\pi}}$. Thus if $\gamma \leq \frac{\beta c}{1-\beta}$ then we have $J_\infty^{\pi'} \geq J_\infty^\pi$, which means that it is optimal to allocate the slot at time t .

Now we redefine policy $\hat{\pi}$ to be the following. Policy $\hat{\pi}$ does not allocate a slot for time t . After time t it allocates exactly the same as policy π . Therefore in the worse case under policy

$\hat{\pi}$ there will always have one more packet in the queue. So we have

$$J_{\infty}^{\pi} - J_{\infty}^{\hat{\pi}} \geq \beta^{t-1} \gamma - \frac{\beta^t c}{1-\beta} \quad (\text{A-2})$$

Therefore if $\gamma \geq \frac{\beta c}{1-\beta}$, then we have $J_{\infty}^{\pi} \geq J_{\infty}^{\hat{\pi}}$. On the other hand since $J_{\infty}^{\pi'} \leq J_{\infty}^{\hat{\pi}}$, we conclude that if $\gamma \geq \frac{\beta c}{1-\beta}$, then $J_{\infty}^{\pi} \geq J_{\infty}^{\pi'}$. Therefore it is optimal not to allocate the slot for the next time frame.

When $\gamma = \frac{\beta c}{1-\beta}$, from the above two arguments we know that $J_{\infty}^{\pi} = J_{\infty}^{\pi'}$, meaning allocating and not allocating are equally optimal. ■

Lemma A-2: Suppose $d(t) = 0$. If $\gamma \leq \frac{\beta p c}{1-\beta p}$, then it is optimal to allocate the slot for the next time frame and if $\gamma \geq \frac{\beta p c}{1-\beta p}$ then it is optimal not to allocate the slot for the next time frame. In case of equality it is equally optimal to allocate or not to allocate the slot.

Proof: If $\gamma \geq \frac{\beta c}{1-\beta}$, then with a method similar to that in Lemma A-1 it can be shown that it is optimal not to allocate a slot for the next time frame (note that if $d(t) = 0$, (A-2) holds for π and $\hat{\pi}$ defined in Lemma A-1).

Now consider the case $\gamma \leq \frac{\beta c}{1-\beta}$.

Let π be the policy that allocates a slot when $d(t) = 0$ and π' be the policy that does not allocate a slot when $d(t) = 0$. Both policies allocate the slot when $d(t) \neq 0$, since this is optimal by the previous lemma and the assumption that $\gamma \leq \frac{\beta c}{1-\beta}$.

Let $d(t) = 0$. Then, policy π allocates a slot and policy π' does not allocate a slot for time t .

If $a(t-1) = 0$, then both policies will be in the same state at time $t+1$. If $a(t-1) = 1$, then both policies continuously allocate a slot until the deterministic part becomes zero again, (note that policy π allocates the slot by definition and policy π' allocates the slot because $\gamma \leq \frac{\beta c}{1-\beta}$). After π' has transmitted all the deterministic packets, then both queues will again be in the same state. Therefore we have $J_{\infty}^{\pi'} \geq J_{\infty}^{\pi}$ iff

$$\begin{aligned} & \sum_{u=t+1}^{\infty} \beta^{u-1} p^{u-t} c - \beta^{t-1} \gamma \geq 0 \\ \iff & \beta^{t-1} \left(\frac{\beta p c}{1-\beta p} - \gamma \right) \geq 0 \end{aligned} \quad (\text{A-3})$$

Thus, If $\gamma \leq \frac{\beta p c}{1-\beta p}$, then $J_{\infty}^{\pi'} \geq J_{\infty}^{\pi}$, thus it is optimal to allocate the slot for the next time frame. If $\gamma \geq \frac{\beta p c}{1-\beta p}$ then $J_{\infty}^{\pi'} \leq J_{\infty}^{\pi}$ and it is optimal not to allocate the slot for the next time frame. In case of equality it is equally optimal either to allocate or not to allocate the slot. ■

REFERENCES

- [1] C. Buyukkoc, P. Varaiya, and J. Warland, "The $c\mu$ -rule revisited," *Advances in Applied Probability*, vol. 17, pp. 237–238, 1985.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] R. Weber and G. Weiss, "On an index policy for restless bandits," *Journal of Applied Probability*, vol. 27, pp. 637–648, 1990.
- [11] N. Ehsan and M. Liu, "Optimal bandwidth allocation with delayed state observation and batch assignment," *EECS Technical Report CGR 03-11, University of Michigan, Ann Arbor*, March 2003.
- [12] J. Nino-Mora, "Restless bandits, patial conservation laws, and indexability," *Advances in Applied Probability*, Vol. 33, no. 1, pp. 76–98, 2001.
- [13] M. Raissi-Dehkordi and J. S. Baras, "Broadcast scheduling in information delivery systems," *Proc. IEEE GLOBECOM*, 2002.