Optimal Bandwidth Allocation With Delayed State Observation and Batch Assignment

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 consider the problem of allocating bandwidth/server to multiple user transmitters/queues with identical but arbitrary arrival processes, to minimize the total expected holding cost of backlogged packets in the system over a finite horizon. The special features of this problem are that (1) packets continuously arrive at these queues regardless of the allocation decision, (2) there is a large delay between when the allocation decision is made and when the allocation is used, thus decisions can be viewed as based on delayed or obsolete state information/observations; and (3) the bandwidth allocation is in batches of time slots so that each queue can be assigned any number of slots not exceeding the total number in a batch. This problem is motivated by channel allocation in a communication system involving large propagation delay, e.g., a typical satellite data communication scenario. This problem can be cast as a special case of the restless bandit problem with multiple plays with the additional properties that the state observation is imperfect and that each queue can be served multiple times in between two decision epochs. We identify two sufficient conditions which jointly define a class of policies that are optimal. We further show that under very mild assumptions on the arrival process these two conditions are also necessary for the optimality of a policy. Without these assumptions on the arrival process the two conditions are not in general necessary, which we show via two examples.

Index Terms

Resouce allocation, multi-armed bandit, restless bandit, optimal policy

I. Introduction

In this paper we study a problem of optimally allocating bandwith (or servers) to parallel queues with identically but arbitrarily distributed arrival processes. Special features of this problem are that servers can be assigned *in batches*, i.e., multiple servers can be allocated to the same queue at a time, and that there is a signifi cant delay between when the allocation decision is made and when the queues are being served, i.e., allocation decision is based on obsolete or delayed state observations.

This optimal bandwidth allocation problem is primarily motivated by communication systems that have large propagation delay, e.g., a typical data communication scenario in a satellite network as illustrated in Fig. 1. Users/terminals transmit packets to the Network Operating Center (NOC) via the satellite. The data communication link from users to the satellite, also known as the *return channel*, follows a dynamic TDMA schedule. Each user is assgined/allocated a certain number of *slots* within a *frame* that consists of a fi xed number of slots. A user can only transmit within its assigned slots during every frame. A user also inform 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 and back), the allocation decision for a particular frame is made based on the backlog information collected during the 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. This results in possible over-allocation or under-allocation. Therefore in this case the allocation needs to take into account unknown random arrivals that occur in between observations or state information updates.

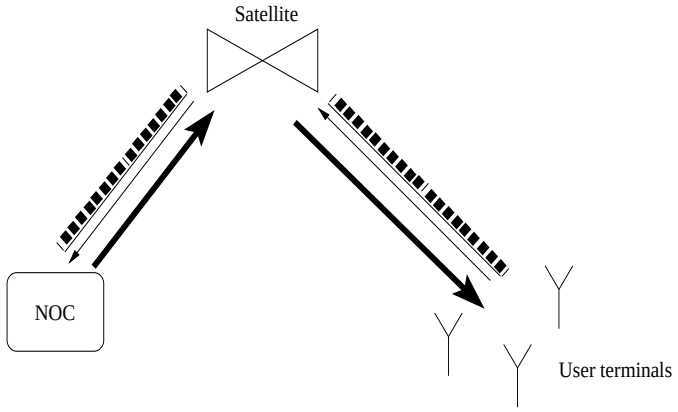

Fig. 1. The saatellite communication scenario

Similar resource allocation problems also arise in systems where resource allocation is done relatively infrequently compared to packet transmission time, due to cost or design constraints.

In this paper we assume backlogged packets incur a holding cost, and formulate an optimal bandwidth allocation problem for the above scenario, with the objective of minimizing the expected total packet holding cost over a finite time horizon. This objective aims at removing as many packets as possible from the queues since packets remaining in the queue incur a cost. Alternatively it can also be viewed as trying to maximize the total throughput (or utilization) of the channel. We identify two conditions that charaterize a class of allocation policies, and prove their sufficiency for any policy within this class to be optimal given the above objective. In addition, we show that under very mild conditions on the arrival process, these two conditions are also necessary for a policy to be optimal. Without such conditions we show via two examples that in general these two conditions are not necessary for a policy to be optimal.

Bandwidth allocation problems have been extensively studied in the literature under various scenarios. Here we review studies most relevant to the one investigated in this paper. Dynamic bandwidth allocation can often be formulated as a dynamic programming problem, the solution of which can always be obtained via numerical methods. However, to obtain a qualitative optimal strategy/policy there is in general no unified framework. The derivation of specific strategies is often highly dependent on the specific scenario under given assumptions.

In [1] the problem of parallel queues with different holding cost and a single server was considered, and the simple $c\mu$ rule was shown to be optimal.

[2], [3], [4] considered the server allocation problem to multiple queues with varying connectivity 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 serving the Longest Connected Queue (LCQ) policy stabilizes the system if system is stablizable. The same policy minimizes the delay in the special case of symmetric queues. [5] further consider 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 the index rule holds when the indices are sufficiently separated. All the above considered random connectivity of queues, but the state of the system, i.e., connectivity and the number of packets in each queue, is always known before server allocation is made. In addition it is also commonly assumed that no more than one server can be allocated to a queue at a time. [6], [7] considered the 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 different from the one examined in this paper and they lead to different solutions.

Another set of related problems are the *bandit* problems. The *multi-armed bandit* problem is concerned with the dynamic allocation of a single server to multiple projects/arms, stated in discrete time as follows. Consider a system of N independent projects/porocesses, where at time t the state of the system is given. At any time step, we need to activate one of the N bandit processes. The activated process will yield a reward as a function of its current state and it will change state according to a Markov process. The goal is to minimize the discounted cost over an infinite horizon. Gittins [9] proved that an index policy is optimal for this problem, where the indices are calculated for each bandit individually as a function of their current states. The optimality of the Gittins index rule has been shown using various techniques, see for example [9], [10], [11], [12]. For a survey see [13] and the references therein. It is shown in [14] that the Gittins index rule is not in general optimal for the case with multiple servers/plays, known as the multi-armed bandit problem with *multiple plays*. In [15] it is shown that the index policy is optimal under certain conditions on the reward process.

The problem studied in this paper can be cast as a special case of the *restless multi-armed bandit* problem [16] 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 change as packets arrive. [16] and [17] studied the asymptotic behavior of this class of problems when the number of arms/projects and servers are infinite with their ratio fixed. Their results do not apply here as the number of queues and servers are finite. A general optimal solution is not known for this class of problems. Our problem is a special case of this class of problems in that

1) The state of the system is not known at the time when the decision is made. This is due to the large propagation delay of the satellite channel;
2) Multiple servers can be assigned to the same project/queue at one decision epoch. That is, the allocation decision not only needs to specify which project/queue to serve (i.e., which queues to assign slots to), but also "how much" it plays/serves (i.e., how many slots to assign to those queues).

The rest of the paper is organized as follows. In the next section we describe our network model and formulate the related optimization problem. In Section III we define a class of policies called the *residual max-min fair* policies and prove their optimality. We also derive conditions on the arrival process under which it is necessary for an optimal policy to be residual max-min fair. In Section IV we show via two examples that in general it is not necessary for an optimal policy to be residual max-min fair. Section V discusses extensions and generalizations of the problem studied here and concludes the paper.

II. Network Model and Problem Formulation

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. The Network Model

Consider N queues that need to transmit packets to a single server/receiver and compete for shares of a common channel that consists of time slots. Packets arrive at each queue according some abitrary random process. We will assume that queues have identical arrival processes. Packets from these queues are of equal length and one packet transmission time occupies one slot time. The allocation of this channel is done once for M slots (M may or may not be greater than N), i.e., assignment is done in batches. These M slots together constitute a frame. Within each allocation decision, a queue may be assigned any number of slots not exceeding M. An illustration is given in Fig. 2. Alternatively, the above model can also be viewed as one where N queues are being served by M servers. Different from most of the prior work, here multiple servers can be assigned to a single queue. When this happens, multiple packets will be served.

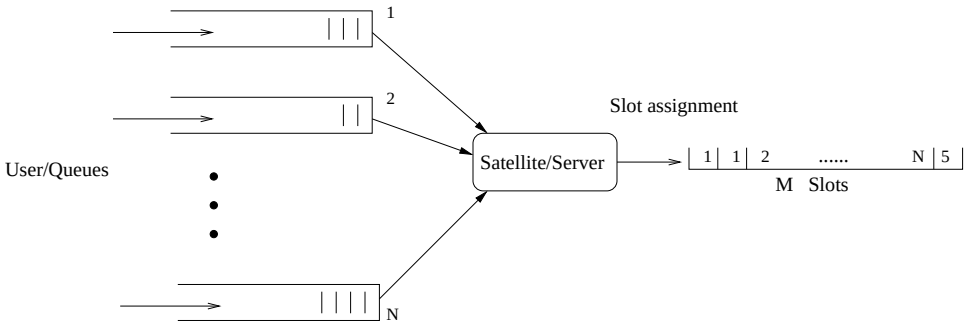

Fig. 2. The network model

The 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/receiver and broadcast to all queues over an 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, which is shown in Fig. 3. Each user advertises its buffer size (denoted by $\mathbf{b}_t$) to the server. The server allocates slots to be used for transmission in the next time frame, denoted by $\mathbf{x}_{t+1}$. This procedure starts from $t = 0$ and ends at $t = T$, the finite time horizon. Note in this scenario during the first frame queues do not have allocated slots 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 keeping a packet in the buffer at the beginning of a frame incurs a unit cost across all queues. Let b_t^i be the queue size of queue/user i at the beginning of frame t and $\mathbf{x}_t$ be

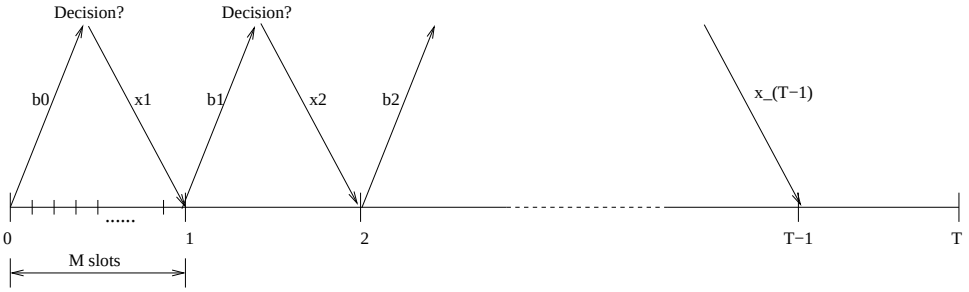

Fig. 3. The bandwidth allocation dynamics

the slot allocation for that frame. The objective is to find an allocation policy that minimizes the following cost function.

$$\begin{aligned} J &= E[C|\mathcal{F}_0], \\ C &= \sum_{t=1}^{T}\sum_{i=1}^{N} b_t^i \end{aligned} \tag{1}$$

Where $\mathcal{F}_0$ summarizes all the information available in the beginning.

B. Assumptions

Below we summarize important assumptions underlying our network model.

1) We assume that each user has an infinite buffer size. Without this assumption we need to introduce penalty for packet dropping/blocking. This is an important extension to work reported in this paper and will be considered in a future study.
2) We assume that if the allocated bandwidth for a user is greater than its buffer occupancy (existing backlog carried over from the previous slot) at the beginning of a frame, the newly-arrived packets during that frame cannot be transmitted using the extra slots during that frame. This assumption is essential because the exact arrival times of the packets arriving in a frame is random (e.g., could be toward the beginning of that frame or the end of that frame). Thus whether an extra slot could be used for a new arrival or not depends on the position of the allocated slot (e.g., the first slot or the last slot of the M slots in the frame), and the arrival time of the packet. Consequently we will have to take into account different ordering of the allocation (i.e, not only that user i is assigned j slots, but also which j slots within the frame). This makes the problem very different and much more complicated. In this paper we will limit our attention to the simpler scenario prescribed by this assumption.
3) We assume that all queues have the same arrival process but mutually independent. Arrivals within each frame are mutually independent and also independent of the queue size. Relaxation of this assumption lead the more general model where different users have different arrival processes. This generalization is discussed in more detail in Section V and is currently being invesigated in a separate study.
4) We do *not* assume that the server knows the arrival process. In other words, the optimal policy does not require the knowledge of the arrival process but it does depend on the fact that queues have identical arrival processes and the previous assumption.
5) We have assumed that all queues have identical holding cost. This precludes differentiation between queues, i.e., all queues are of the same class/priority. Extending this work to consider different service classes is discussed in Section V.

6) The server recalls the latest allocation it has made. Note that the expected cost conditioned on the latest allocation, say $\mathbf{x}_t$ and buffer occupancy $\mathbf{b}_t$ is independent of arrivals that occurred before frame t, (Note that $\mathbf{b}_t$ is a Markov chain with state space $\{1, 2, \cdots\}$ where the transition probabilities depend on the control action $\mathbf{x}_t$).
7) 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 $t = 0, 1, \cdots T$, with each increment representing a frame length. Frame t refers to the frame defined by the interval $[t, t+1)$. In subsequent discussions we will use terms *frames*, *steps* and *stages* interchangeably. We will also use the terms *bandwidth* and *slots* interchangeably.

In general we will use subscripts to denote the time index and the superscripts to denote a specific user/queue. For example b_t^i 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. In additionh, if $\mathbf{v} = [v_1, \cdots, v_N]$ is an N dimensional vector, then $v = \sum_{i=1}^N v_i$. For example $\mathbf{b_t}$ is an $N \times 1$ vector representing the number of packets in each queue at time t, and b_t is a scalar value that represents the total number of packets in all queues at time t: $b_t = \sum_{i=1}^N b_t^i$.

Whenever we need to distinguish two policies, we show the policy as a superscript. For example $b_t^{i,\pi}$ means the buffer size of the i-th queue at time t under policy π.

A list of notations are as follows.

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

$\mathbf{x_t} = [x_t^1, x_t^2, \cdots, x_t^N]'$: The number of packet slots (amount of bandwidth) allocated to users.

$\mathbf{d_t} = [\mathbf{b_{t-1}} - \mathbf{x_{t-1}}]^+$, where $[w]^+$ takes value w or 0, whichever is greater: The number of packets in the i-th queue at time t (i.e., at the end of the $(t-1)^{th}$ frame, or the begining of the t^{th} frame). This value can be completely determined by using 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 frame). Alternatively we will also call this value the amount of *deterministic packets* to be distinguished from the random arrivals occurred during that frame.

$\mathbf{a_t} = [a_t^1, a_t^2, \cdots a_t^N]'$: The number of packet arrivals during frame t.

$\mathbf{y_t} = [\mathbf{x_t} - \mathbf{d_t}]^+$: The excess bandwidth allocated to queue i for random arrivals occurred during frame $t-1$.

λ: The average number of arrivals during each frame for each queue.

p_l: The probability of l packet arrivals during a frame for a single queue. Thus $Pr[a_t^i = l] = p_l,\ \forall t, i$.

$C_t = \sum_{u=t}^T \sum_{i=1}^n b_u^i$: The cost from time t on, or the cost to go.

$\mathbf{d_t^{i+}} := \mathbf{d_t} + \mathbf{e_i}$ where $\mathbf{e_i}$ is an N-dimensional vector with all entries zero except for 1 in the i-the position.

$\mathcal{F}_t$: The σ-fields induced by all information through time t.

III. Optimal Policies

In this section we will first establish two conditions that define a class of allocation policies. We will then prove that these two conditions are sufficient for the entire class of policies to be optimal for the objective given in the previous section. In the process of proving this, we will also show that under mild conditions on the arrival process these two conditions are also necessary for a policy to be optimal.

A. Preliminaries

Claim 1: An optimal policy exists for the problem outlined in the previous section.

Proof: The total number of policies (possible allocations) is $(N^M)^{T-1} = N^{M(T-1)}$, which is finite considering a finite horizon T. So the cost function will have a minimum over all possible policies, which means that an optimal policy exists. ∎

Note that the optimal policy need not be unique. Without loss of generality we assume $\mathbf{x_0} = 0$. The buffer occupancy evolves over time as follows:

$$b_t^i = [b_{t-1}^i - x_{t-1}^i]^+ + a_{t-1}^i, \tag{2}$$

where a_{t-1}^i is a random variable that represents the arrivals in the interval $[t-1, t)$ to queue i.

By assumption 3, b_{t-1}^i and a_{t-1}^i are independent. x_t is in general a function of $b_0^i, a_0^i, a_1^i, \cdots, a_{t-1}$ and $x_1^i, \cdots, x_{t-1}^i$. However, by the Markov property of the couple ($\mathbf{b}_t$, $\mathbf{x}_t$), the optimum allocation will be a function of only the last state and the last allocation, b_{t-1}^i, x_{t-1}^i. Equation (2) shows that the buffer occupancy at time t consists of two parts, $d_t^i = [b_{t-1}^i - x_{t-1}^i]^+$, which is deterministic (meaning that it is known when making the decision $\mathbf{x_t}$), and a random part a_{t-1}^i. In subsequent discussions we will call them the *deterministic packets* and *random packets/arrivals*, respectively. d_t^i is the information we have available when making a decision for allocation of frame t. Thus Equation (2) can be re-written as

$$b_t^i = d_t^i + a_{t-1}^i \ . \tag{3}$$

Note that we also have $\mathbf{d_1} = \mathbf{b_0}$, therefore the cost function to be minimized can also be expressed as

$$J = E[C|\mathbf{d_1}, \mathcal{F}_1]. \tag{4}$$

From the definition of $\mathbf{d_{t+1}}$ given earlier, $\mathbf{b_t}, \mathbf{x_t}, \mathbf{d_{t+1}}$ are not independent, i.e., given two the third one is uniquely determined. Therefore we have the following result:

Remark 1: For any allocation policy we have

$$E[C_{t+1}|\mathbf{b_t}, \mathbf{x_t}, \mathcal{F}_t] = E[C_{t+1}|\mathbf{d_{t+1}}, \mathbf{x_t}, \mathcal{F}_t] = E[C_{t+1}|\mathbf{d_{t+1}}, \mathcal{F}_t] \ \ a.s. \tag{5}$$

The first equality is due to the dependency of $\mathbf{b_t}, \mathbf{x_t}, \mathbf{d_{t+1}}$, and the second equality is due to the fact that the expected cost to go is averaged over distribution of $\mathbf{b_{t+1}}$, which is independent of $\mathbf{x_t}$ given $\mathbf{d_{t+1}}$ from Equation (3). Following this result, in all our subsequent discussions we will focus on the conditional cost $E[C_{t+1}|\mathbf{d_{t+1}}, \mathcal{F}_t]$ rather than $E[C_{t+1}|\mathbf{b_t}, \mathbf{x_t}, \mathcal{F}_t]$.

Definition 1: An allocation $\{x_1, x_2, \cdots x_N\}$ is *max-min fair* if $x_i - x_j \leq 1$ for all i, j.
This defines allocations that result in no queue being allocated more than 2 slots than any other queue. This is essentially the discrete version of the more general max-min fairness.

B. Main Results

We define a class of allocation policies $\mathcal{P}$ as follows.

Definition 2: If π is a policy that allocates bandwidth to N queues such that it satisfies the following two conditions

- **(C-1)** all the deterministic packets (or existing backlog) have priority over the random arrivals, i.e. as long as there are known backlog in the queue that have not been scheduled, no slots can be allocated to the unknown, random packets that may or may not be there;
- **(C-2)** if there is more bandwidth/slots than the total number of deterministic packets in the system, then the residual bandwidth/slots will be allocated to different queues in a max-min fair manner (i.e. $\{y_1, \cdots y_N\}$ is max-min fair);

then we call π a *residual max-min fair* (RMF) allocation. We denote by $\mathcal{P}$ the set of all RMF allocations.

Note that strictly speaking our definition of (C-2) implies that (C-1) is satisfied. We state them separately for the ease of our discussions and proofs that follow. Following this definition, we have the following main results of this paper.

Theorem 1: Any policy $\pi \in \mathcal{P}$ minimizes the expected total holding cost given in Equation (1), in that for any other policy $\pi^{'} \notin \mathcal{P}$ we have

$$E^{\pi}[C|\mathcal{F}_0] \leq E^{\pi^{'}}[C|\mathcal{F}_0] \;\; a.s.$$

Theorem 2: Suppose π is an optimal policy that minimizes the expected total holding cost as defined in Equation (1). We have

1) if $p_0 > 0$, then π satisfies C-1;
2) if $p_i > 0$ for all $i \leq \lfloor \frac{M}{N} \rfloor$, then π also satisfies C-2.

Theorem 1 states the sufficient conditions of an optimal policy, the two conditions (C-1) and (C-2) outlined above. Theorem 2 states that these conditions are also necessary for an optimal policy if the arrival process satisfies certain conditions, namely that there has to be a non-zero probability of any number, not exceeding $\lfloor \frac{M}{N} \rfloor$, of packet arrivals during a frame time. (Note that this condition is obviously true in the case of a Poisson arrival process.) From Theorem 1, all $\pi \in \mathcal{P}$ are equally optimal, i.e., incur the same cost given the cost function. This means that given the objective function it does not matter whether we favor one queue over another so long as we allocate for all the existing backlog before handling random arrivals. In other words, since the holding cost is identical across all queues and queues are infinite, any backlogged packets, no matter which queue they are in, are equivalent. So serving one or the other does not make a difference. On the other hand, random arrivals have to be treated fairly, i.e., no queue should be more over-allocated than others so as to achieve a balance between queues.

In the next subsection we will prove these sufficienct and necessary results through a sequence of lemmas.

C. Proof of Optimality

In the following lemma we show that under any policy $\pi \in \mathcal{P}$ the expected cost depends only on the total number of deterministic packets in the system and not on individual queue sizes.

Lemma 1: Under any given policy $\pi \in \mathcal{P}$, $E^{\pi}[C_t|\mathbf{d_t}, \mathcal{F}_t] = E^{\pi}[C_t|d_t, \mathcal{F}_t] = f_t^{\pi}(d_t)$ $a.s.$ for some deterministic function $f_t^{\pi}(\cdot)$. Moreover, the performance of all policies $\pi \in \mathcal{P}$ is the same (i.e. $f_t^{\pi}(d_t) = f_t(d_t) \quad \forall \pi \in \mathcal{P}$).

Note that in general, $f_t(.)$ depends on the set we are averaging on, which is defined by $\mathcal{F}_t$. However, for convenience we will not emphasize this in notation, unless there is an ambiguity.

Proof: We use backward induction on t.

Induction basis: Let $t = T$, we have $E^\pi[C_T|\mathbf{d_T}] = d_T + N\lambda = f_T(d_T)$ $a.s.$, where $N\lambda$ is the expected total arrivals during the last frame. Note that the function does not depend on π. Therefore the induction basis is established.

Induction step: Suppose $E[C_{t+1}|\mathbf{d_{t+1}}] = f_{t+1}(d_t + 1)$ $a.s.$ – the induction hypothesis. We want to show $E^\pi[C_t|\mathbf{d_t}] = f_t(d_t)$ $a.s.$.

$E^\pi[C_t|\mathbf{d_t}]$ can be written as follows:

$$E^\pi[C_t|\mathbf{d_t}, \mathcal{F}_t] = d_t + N\lambda + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} E[C_{t+1}|[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+] \ a.s., \tag{6}$$

where $\mathbf{u} = (u_1, u_2, \ldots, u_N)^T$ is the arrivals during the $(t-1)^{th}$ frame. The first term on the right-hand-side (RHS) of (6) is the known backlog in the system at time t. The second term is the expected amount of new arrivals by time t. The third term is the expected cost to go from time $t+1$ on, averaged over all possible arrival patterns. Note that $[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+ = \mathbf{d}_{t+1}$ *a.s.*

By the induction hypothesis we have

$$E[C_{t+1}|[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+, \mathcal{F}_t] = f_{t+1}(\text{sum}[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+) \ a.s. \tag{7}$$

where $\text{sum}(\mathbf{w}) = \sum_i w_i$. We now consider two cases, $d_t > M$ and $d_t \leq M$, respectively.

Case I ($d_t > M$): In this case under C-1, all assigned slots will be for the deterministic part of the queues. So $\text{sum}[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+ = d_t - M + \sum_{j=1}^{n} u_j$ *a.s.*, using Equation (7) we have

$$E[C_{t+1}|[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+, \mathcal{F}_t] = f_{t+1}(d_t - M + \sum_{j=1}^{n} u_j) \ a.s.$$

Thus we can write equation 6 as follows

$$E^\pi[C_t|\mathbf{d_t}, \mathcal{F}_t] = d_t + N\lambda + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} f_{t+1}(d_t - M + \sum_{j=1}^{n} u_j) = f_t(d_t) \ a.s. \tag{8}$$

Note that the M packets chosen for transmission can be from any queue (or any combination of queues) and this will not affect the expectation of the cost. Therefore all policies satisfying C-1 will have the same performance in this case.

Case II ($d_t \leq M$): In this case all the deterministic packets are assigned slots and possibly (when inequality is strict) some random new arrivals will also be scheduled. Let $M - d_t = rN + q$ where r and q are both non-negative integers and $0 \leq q < N$. This is always possible since $q = (M - d_t) \mod N$.

By (C-2) of the policy $\pi \in P$ slots will be allocated to the random part of each queue according to the following specific rule. q users will have $r+1$ slots assigned and $n-q$ users will have r slots assigned. Without loss of generality we will assume that π is a policy that assigns $r+1$ slots to the first q queues and r slots to the remaining $N-q$ queues. We can then write Equation (6) as follows

$$\begin{aligned} E^\pi[C_t|\mathbf{d_t}, \mathcal{F}_t] &= d_t + N\lambda + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} f_{t+1}([u_1 - (r+1)]^+ + \cdots \\ &\quad + [u_q - (r+1)]^+ + [u_{q+1} - r]^+ + \cdots + [u_N - r]^+) \\ &= f_t^\pi(d_t) \ a.s. \end{aligned} \tag{9}$$

if we choose a different policy in $\mathcal{P}$ that assigns $r+1$ slots to q random queues, not necessarily the first q, then by exchanging the sums in Equation (9) we get the same result. Therefore for any other policy in $\mathcal{P}$ the same expectation will hold, i.e. for all policies satisfying (C-1) and (C-2) the expected cost will be the same.

Combining the two cases, the induction step is proved. So Lemma 1 holds for all $1 \leq t \leq T$. ∎

Definition 3: Define $g_t(\mathbf{d_t}^i)$ as follows:

$$g_t(\mathbf{d_t}^i) := E[C_t|\mathbf{d_t^{i+}}, \mathcal{F}_t] - E[C_t|\mathbf{d_t}, \mathcal{F}_t]\,. \tag{10}$$

By Lemma 1, for all policies $\pi \in \mathcal{P}$ we have

$$g_t^\pi(\mathbf{d_t}^i) = E^\pi[C_t|\mathbf{d_t^{i+}}, \mathcal{F}_{t-1}] - E^\pi[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] = f_t(d_t+1) - f_t(d_t) = g_t^\pi(d_t)\ a.s., \tag{11}$$

which is independent of i.

Lemma 2: $g_t^\pi(d_t)$ is a positive and non-decreasing function of d_t almost everywhere, for all $d_t \geq 0$ under any policy $\pi \in \mathcal{P}$.

Proof: We use backward induction on t. Again we will omit the superscript π without causing ambiguity.

Induction basis: Let $t = T$. Then we have

$$E[C_T|\mathbf{d_T^{i+}}, \mathcal{F}_{T-1}] = d_T + 1 + N\lambda = E[C_T|\mathbf{d_T}, \mathcal{F}_{T-1}] + 1 \ \ \Rightarrow \ \ g_T(d_T) = 1\ a.s.$$

So the induction basis is established.

Induction step: Assume $g_{t+1}(d_{t+1})$ is a positive and non-decreasing function of d_{t+1}. We want to show that $g_t(d_t)$ will also be a positive and non-decreasing function of d_t. Let's assume the allocation in the case of $\mathbf{d_t}$ is $\mathbf{x_t}$ and $\hat{\mathbf{x}}_\mathbf{t}$ in the case of $\mathbf{d_t^{i+}}$ under the same policy. Then we have

$$E[C_t|\mathbf{d_t^{i+}}, \mathcal{F}_{t-1}] = N\lambda + d_t + 1 + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} E[C_{t+1}|[\mathbf{d_t^{i+}} + \mathbf{u} - \hat{\mathbf{x}}_\mathbf{t}]^+, fcal_t]\ \ a.s. \tag{12}$$

$$E[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] = N\lambda + d_t + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} E[C_{t+1}|[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+, \mathcal{F}_t]\ \ a.s. \tag{13}$$

So we can write $g_t(d_t)$ as follows:

$$g_t(d_t) = 1 + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} \{E[C_{t+1}|[\mathbf{d_t^{i+}} + \mathbf{u} - \hat{\mathbf{x}}_\mathbf{t}]^+, \mathcal{F}_t] - E[C_{t+1}|[\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+, \mathcal{F}_t]\}\ \ a.s. \tag{14}$$

We now consider two cases, $d_t \geq M$ and $d_t < M$, respectively.

Case I ($d_t \geq M$): In this case, by (C-1) of our policy π, all the slots are assigned to the deterministic packets and there will not be any extra slots allocated to random packets. Thus we can write Equation (14) as follows:

$$g_t(d_t) = 1 + \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} g_{t+1}(d_t - M + \sum_{j=1}^{n} u_j)\ a.s., \tag{15}$$

which is positive and non-decreasing in d_t by the induction hypothesis.

Case II ($d_t < M$): In this case all the deterministic packets will be scheduled and some random packets will be assigned slots as well. Let $M - d_t = rN + q$, where r and q are non-negative integers and $0 \leq q < N$. We further consider two cases $q = 0$ and $q > 0$, respectively.

Case II.a ($q = 0$): Since $d_t < M$, $q = 0$ implies $r \geq 1$. By (C-2) of the allocation policy π, we have $y_t^i = r, \ \forall i$, i.e. we assign r extra slots to all queues for random packets in addition to the deterministic part. When we add one more deterministic packet to the i-th queue, then by (C-1), one of the extra slots must be allocated to this additional packet. Without loss of generality suppose we use one of the extra slots of the first queue, leaving $r-1$ extra slots for the first queue. We can then write Equation (14) as follows:

$$g_t(d_t) = 1 + \sum_{u_1 \geq r} \sum_{u_2=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} g_{t+1}((u_1 - r) + [u_2 - r]^+ + \cdots + [u_N - r]^+) \ \ a.s. \quad (16)$$

Note here we are only summing over $u_1 \geq r$. This is because for $u_1 < r$, i.e., the number of new arrivals is below the extra slots assigned, there is no difference between the two scenarios $\mathbf{d_t^{i+}}$ and $\mathbf{d_t^i}$ and the corresponding g_{t+1} would be zero.

Case II.b ($q > 0$): In this case q of the queues (without loss of generality assuming queue 1 through q) have $r+1$ slots assigned for their random arrivals and $n - q$ queues ($q+1$ through N) have r slots assigned for their random arrivals. By (C-2) of policy π, if we have one more deterministic packet, then one of the queues with $r+1$ slots allocated will have to give up one of its slots (again without loss of generality assuming it is the first queue) for the additional deterministic packet. Thus Equation (14) can be written as follows:

$$\begin{aligned} g_t(d_t) \ &= \ 1 + \sum_{u_1 \geq r+1} \sum_{u_2=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} g_{t+1}((u_1 - (r+1)) + \cdots \\ &\quad + [u_q - (r+1)]^+ + [u_{q+1} - r]^+ + \cdots + [u_N - r]^+) \ \ a.s. \end{aligned} \quad (17)$$

Again here we are only summing over $u_1 \geq r+1$ since otherwise there is no difference between the two scenarios $\mathbf{d_t^{i+}}$ and $\mathbf{d_t^i}$.

Note that since $M - d_t = rN + q$, as d_t increases, r is non-increasing. So by increasing d_t, the number of elements in the summations in equations (16) and (17) are both non-decreasing. Therefore, using the induction hypothesis that $g_{t+1}(\cdot)$ is positive, we conclude that $g_t(d_t)$ is positive and non-decreasing in $d_t \geq 0$, completing the induction step. ∎

Corollary 1: $f_t(d_t)$ is a non-negative convex function of d_t almost surely for $d_t \geq 0$.

Lemma 1 shows that under a policy in $\mathcal{P}$, the expected cost will depend only on the total amount of backlog in the system, i.e., the expected cost to go from time t will depend only on the total number of deterministic packets in the system at time t, rather than the vector $\mathbf{b_0}$ or $\mathbf{d_1}$. Using this together with Equation (5) we have

$$E^\pi[C_{t+1} | \mathbf{b_t}, \mathbf{x_t}, \mathcal{F}_t] = E^\pi[C_{t+1} | \mathbf{d_{t+1}}, \mathcal{F}_t] . E^\pi[C_{t+1} | d_{t+1}, \mathcal{F}_t] \ a.s. \quad (18)$$

The next two lemmas establish the optimality of (C-1) and (C-2). The first shows that for any policy that does not satisfy (C-1), we can always construct a policy that satisfies (C-1) and results in at most the cost of the original policy. The second shows that for any policy that satisfies (C-1) but does not satisfy (C-2), we can always construct a policy that satisfies (C-2) and results in at most the cost of the original policy.

Lemma 3: Suppose a policy π violates (C-1), then there exists a policy $\pi^{'}$ such that $\pi^{'}$ does not violate (C-1) and $E^{\pi^{'}}[C|\mathcal{F}_0] \leq E^{\pi}[C|\mathcal{F}_0]$ *a.s.*, moreover, if $p_0 > 0$, the inequality is strict.

Proof: Suppose π violates (C-1) in some frame t, some queue must be under-allocated (i.e., not all deterministic packets are assigned slots) and some queue must be over-allocated (i.e., extra slots are assigned for random arrivals). Therefore under π there exist i, j such that $x_t^i < d_t^i$ and $x_t^j > d_t^j$.

We construct a policy $\hat{\pi}$ as follows. $\hat{\pi}$ is the same as π up to frame t. Therefore cost incurred upto and including time t under π and $\hat{\pi}$ is the same. At time frame t, let $\hat{x}_t^i = x_t^i + 1$ and $\hat{x}_t^j = x_t^j - 1$ be the allocation under $\hat{\pi}$ for i, j, respectively. In words, we re-allocate one of the slots reserved for queue j's random arrivals to queue i's existing backlog, and keep everything else the same. Under policy π there are $y_t^j = x_t^j - d_t^j$ slots reserved for random arrivals at queue j. Under policy $\hat{\pi}$ we denote by $\hat{y}_t^j = x_t^j - d_t^j - 1$ the amount reserved for random arrivals at queue j.

From this point on two situations are possible. Consider the first case where the number of arrivals to queue j during the $t-1$-th time frame (a_{t-1}^j) is less than $x_t^j - d_t^j$, i.e., $a_{t-1}^j < y_t^j$, then π and $\hat{\pi}$ result in the same backlog in queue j at time $t+1$ since both allocations assign more than necessary. As a result, under $\hat{\pi}$ we will have one less backlogged packet in queue i at time $t+1$, while all other queues have the same backlog compared to that under policy π. That is, $\mathbf{d_{t+1}} = \hat{\mathbf{d}}_{\mathbf{t+1}}^{\mathbf{i+}}$ *a.s.*, where $\hat{\mathbf{d}}$ is the backlog for policy $\hat{\pi}$. Therefore at time $t+1$, cost under π is one more than cost under $\hat{\pi}$. For the remaining frames, we let $\hat{\pi}$ give the exact same allocation $\mathbf{x_{t+1}}, \cdots, \mathbf{x_{T-1}}$ as given by π, thus incurring at most the same cost to go as that under π.

Now consider the second case where $a_{t-1}^j \geq x_t^j - d_t^j$. Under policy $\hat{\pi}$, at $t+1$ we have one packet more in queue j and one packet less in queue i compared to that under policy π. Thus cost incurred up to and including $t+1$ is the same for both policies. At $t+1$, we will let $\hat{\pi}$ assign one more slot to queue j and one less to i compared to that given by π, which then results in the exact same backlog situation at time $t+2$. From this point on, $\hat{\pi}$ will give the exact same allocation as that given by π, incurring the same cost to go as that under π.

Combining the above two cases, we have constructed a policy $\hat{\pi}$, that incurs at most the cost as incurred by π, expressed as follows:

$$\begin{aligned}
E^{\hat{\pi}}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] &= p(a_{t-1}^j < x_t^j - d_t^j) E^{\hat{\pi}}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] + p(a_{t-1}^j \geq x_t^j - d_t^j) E^{\hat{\pi}}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] \\
&\leq p(a_{t-1}^j < x_t^j - d_t^j)(E^{\pi}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] - 1) + p(a_{t-1}^j \geq x_t^j - d_t^j) E^{\pi}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] \\
&= E^{\pi}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}] - p(a_{t-1}^j < x_t^j - d_t^j) \\
&\leq E^{\pi}[C_t|\mathbf{d_t}, \mathcal{F}_{t-1}]
\end{aligned}$$

Where all the equalities and inequalities hold almost surely.

Note that the last inequality becomes strict if $p(a_{t-1}^j < x_t^j - d_t^j)$ is positive. Since $x_t^j - d_t^j > 0$, it suffices to have $p_0 > 0$ in order for the strict inequality to hold. Since the cost upto t is the same under both policies, we have $E^{\hat{\pi}}[C|\mathbf{d_1}, \mathcal{F}_0] \leq E^{\pi}[C|\mathbf{d_1}, fcal_0]$. Moreover, if $p_0 > 0$, the inequality is strict.

Starting from any policy π that does not satisfy (C-1), we can construct a policy $\pi^{'}$ by repeatedly using the same construction outlined above. More specifically, starting from the first time when (C-1) is violated under π, we can construct a policy that assigns more slots to deterministic packets compared to π and results in at most the cost of π. Repeating for a finite number of times we will obtain a policy that satisfies (C-1) at this time frame. We then move on to the next time when (C-1) is violated and so on. Repeating this procedure for a finite number of times, we obtain a policy $\pi^{'}$ that results in at most the cost of policy π and does not violate (C-1). Moreover, if $p_0 > 0$, then $\pi^{'}$ is strictly better than π. ∎

Lemma 4: For a policy π that satisfies (C-1) but not (C-2), there exists a policy π' that satisfies both (C-1) and (C-2), and $E^{\pi'}[C|\mathcal{F}_0] \leq E^{\pi}[C|\mathcal{F}_0]$ *a.s.*

Proof: We use backward induction on the time t when (C-2) is violated under π to show that we can always find a policy that satisfies (C-2) with no increased cost for $t = T-1, T-2, \cdots, 1$.

Induction basis: Suppose π is not residual max-min fair in the last step, i.e., there exist i, j such that, $y^i_{T-1} \geq y^j_{T-1} + 2$. We show that if we subtract one from x^i_{T-1} and add one to user x^j_{T-1}, the expected cost will be no more under the new allocation. Define policy $\hat{\pi}$ to be the same as π up to time frame $T-1$. At $T-1$ let $\hat{y}^i_{T-1} = y^i_{T-1} - 1$ and $\hat{y}^j_{T-1} = y^j_{T-1} + 1$ be the allocation for random arrivals under π for i, j, respectively. For all the other queues set $\hat{y}^k_{T-1} = y^k_{T-1},\ k \neq i, j$. Since the cost incurred in all the previous time frames remain the same we only need to consider the expected cost for the last frame $T-1$. Since the allocation for all the other queues except i, j is the same, the expected queue sizes for those queues remain the same. Thus we only need to consider the difference between the sum of the expected values of queues i, j under these two policies:

$$E^{\pi}[C|\mathcal{F}_0] - E^{\hat{\pi}}[C|\mathcal{F}_0] = (E^{\pi}[b^i_T] + E^{\pi}[b^j_T]) - (E^{\hat{\pi}}[b^i_T] + E^{\hat{\pi}}[b^j_T]) \ \ a.s. \tag{19}$$

We have $E[b^i_T|\mathcal{F}_{T-1}] = \lambda + d^i_T$ *a.s.*and by definition, $d^i_T = [b^i_{T-1} - x^i_{T-1}]^+ = [a^i_{T-2} - y^i_{T-1}]^+$. Under policy π we have (using l to represent the number of arrivals during frame $T-2$):

$$\begin{aligned} E^{\pi}[b^i_T] &= \lambda + \sum_{l=1}^{\infty} l \cdot p_{y^i_{T-1}+l} \ \ a.s.; \\ E^{\pi}[b^j_T] &= \lambda + \sum_{l=1}^{\infty} l \cdot p_{y^j_{T-1}+l} \ \ a.s. \end{aligned}$$

Under policy $\hat{\pi}$ we have:

$$\begin{aligned} E^{\hat{\pi}}[b^i_T] &= \lambda + \sum_{l=1}^{\infty} l \cdot p_{\hat{y}^i_{T-1}+l} = \lambda + \sum_{l=1}^{\infty} l \cdot p_{y^i_{T-1}+l-1} \ \ a.s.; \\ E^{\hat{\pi}}[b^j_T] &= \lambda + \sum_{l=1}^{\infty} l \cdot p_{\hat{y}^j_{T-1}+l} + \lambda + \sum_{l=1}^{\infty} l \cdot p_{y^j_{T-1}+l+1} \ \ a.s. \end{aligned}$$

Using the above equations we get:

$$(E^{\pi}[b^i_T] + E^{\pi}[b^j_T]) - (E^{\hat{\pi}}[b^i_T] + E^{\hat{\pi}}[b^j_T]) = \sum_{l=1}^{y^i_{T-1} - y^j_{T-1} - 1} p_{y^j_{T-1}+l} \geq 0 \ \ a.s., \tag{20}$$

which means that the expected cost for policy $\hat{\pi}$ is at most the expected cost for policy π. It is also clear from the above expression that if $p_{y^j_{T-1}+1} > 0$ then the inequality is strict. $\hat{\pi}$ may or may not be residual max-min fair at time $T-1$, but $\hat{\pi}$ is "closer" to being RMF than π. If we repeat this procedure a finite number of times we can obtain a policy π' that satisfies (C-2) at time $T-1$ and results in at most the cost incurred by π. Induction basis is thus established.

Induction step: Now assume that π satisfies (C-2) for all time frames $t+1, t+2, \cdots, T-1$, but violates (C-2) at t. This means that there exists i, j such that $y^i_t \geq y^j_t + 2$. Define policy $\hat{\pi}$ to be the same as π up to time t. At time t we let $\hat{x}^i_t = x^i_t - 1$ and $\hat{x}^j_t = x^j_t + 1$, i.e., subtract one slot from queue i and allocate it to queue j. Thus $\hat{y}^i_t = y^i_t - 1$ and $\hat{y}^j_t = y^j_t + 1$. Starting $t+1$, both policies satisfy (C-2). We show that policy $\hat{\pi}$ incurs at most the cost incurred by policy π.

Since the policies are the same for all frames before t, we only need to consider the cost to go from t on, $E[C_{t+1}|\mathcal{F}_t]$. The number of deterministic packets in the queues at time t, $\mathbf{d_t}$, serves as an initial condition. Since from frame $t+1$ on (C-2) is satisfied, by Lemma 1 the expected cost of all those frames will only depend on the sum of all the packets in the system. We have

$$\begin{aligned} &E^{\pi}[C|\mathcal{F}_0] - E^{\hat{\pi}}[C|\mathcal{F}_0] = \\ E^{\pi}[C|\mathcal{F}_{t-1}] - E^{\hat{\pi}}[C|\mathcal{F}_{t-1}] == \quad &\sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} p_{u_1} \cdots p_{u_N} \{E^{\pi}[C_{t+1}|d_{t+1} = \text{sum}([\mathbf{d_t} + \mathbf{u} - \mathbf{x_t}]^+), \mathcal{F}_t] - \\ &E^{\hat{\pi}}[C_{t+1}|d_{t+1} = \text{sum}([\mathbf{d_t} + \mathbf{u} - \mathbf{\hat{x}_t}]^+), \mathcal{F}_t]\} \;\; a.s., \end{aligned} \tag{21}$$

where $[u_1, \cdots, u_N]$ is the arrival to the queues during the $(t-1)^{th}$ frame. We now partition all the possible transitions from step $t-1$ to step t as follows. The probability that we have u_i new arrivals in queue i and u_j new arrivals in queue j during frame $t-1$ is $p_{u_i} p_{u_j}$. We partition all pairs $\{(u_i, u_j) | u_i, u_j \geq 0\}$ into the following four sets:

$R_1 = \{(u_i, u_j) | u_i \geq y_t^i \;\&\; u_j \geq y_t^j + 1\}$;
$R_2 = \{(u_i, u_j) | u_i \geq y_t^i \;\&\; u_j < y_t^j + 1\}$;
$R_3 = \{(u_i, u_j) | u_i < y_t^i \;\&\; u_j \geq y_t^j + 1\}$;
$R_4 = \{(u_i, u_j) | u_i < y_t^i \;\&\; u_j < y_t^j + 1\}$.

For all pairs $(u_i, u_j) \in R_1$ and $(u_i, u_j) \in R_4$, the sum of deterministic packets in queues i, j is the same under both policies π and $\hat{\pi}$ since R_1 results in under-allocation under both policies, and R_4 results in over-allocation under both policies. For all pairs $(u_i, u_j) \in R_2$ the sum of deterministic packets under policy $\hat{\pi}$ is one more than that under policy π. For all pairs $(u_i, u_j) \in R_3$ the situation is the opposite.

For simplicity, we will use the notation $\sum_{(u_i,u_j) \in R}$ to denote summing over all possible arrivals to all queues such that $(u_i, u_j) \in R$. Using this notation, Equation (21) can be written as

$$\begin{aligned} E^{\pi}[C|\mathbf{d_1}, \mathcal{F}_t] - E^{\hat{\pi}}[C|\mathbf{d_1}, \mathcal{F}_t] \;\; = \;\; &\sum_{(u_i,u_j) \in R_1} p_{u_1} \cdots p_{u_N} \cdot 0 + \sum_{(u_i,u_j) \in R_4} p_{u_1} \cdots p_{u_N} \cdot 0 \\ + \;\; &\sum_{(u_i,u_j) \in R_2} p_{u_1} \cdots p_{u_N} \cdot \{E^{\pi}[C_{t+1}|d_{t+1} = u_i - y_t^i + \sum_{k \neq i,j} [y_t^k - u_k]^+ \\ - \;\; &E^{\hat{\pi}}[C_{t+1}|d_{t+1} = u_i - (y_t^i - 1) + \sum_{k \neq i,j} [y_t^k - u_k]^+\} \\ + \;\; &\sum_{(u_i,u_j) \in R_3} p_{u_1} \cdots p_{u_N} \cdot \{E^{\pi}[C_{t+1}|d_{t+1} = u_j - y_t^j + \sum_{k \neq i,j} [y_t^k - u_k]^+ \\ - \;\; &E^{\hat{\pi}}[C_{t+1}|d_{t+1} = u_j - (y_t^j + 1) + \sum_{k \neq i,j} [y_t^k - u_k]^+\} \;\; a.s., \end{aligned} \tag{22}$$

However, note that

$$(u_i, u_j) \in R_2 \;\Rightarrow\; u_j < y_t^j + 1 < y_t^i \;\&\; u_i \geq y_t^i > y_t^j + 1 \;\Rightarrow\; (u_j, u_i) \in R_3 \,. \tag{23}$$

Define the set R_2' to be

$$R_2' = \{(u_i, u_j) | (u_j, u_i) \in R_2\},$$

and we have $R_2' \subset R_3$ following Equation (23). Equation (21) can thus be written as

$$\begin{aligned}
E^{\pi}[C|\mathbf{d_1},\mathcal{F}_t] - E^{\hat{\pi}}[C|\mathbf{d_1},\mathcal{F}_t] &= \sum_{(u_i,u_j)\in R_2} p_{u_1}\cdots p_{u_N}\cdot\{E^{\pi}[C_{t+1}|d_{t+1}=u_i-y_t^i+\sum_{k\neq i,j}[y_t^k-u_k]^+] \\
&- E^{\hat{\pi}}[C_{t+1}|d_{t+1}=u_i-(y_t^i-1)+\sum_{k\neq i,j}[y_t^k-u_k]^+] \\
&+ E^{\pi}[C_{t+1}|d_{t+1}=u_i-y_t^j+\sum_{k\neq i,j}[y_t^k-u_k]^+] \\
&- E^{\hat{\pi}}[C_{t+1}|d_{t+1}=u_i-(y_t^j+1)+\sum_{k\neq i,j}[y_t^k-u_k]^+\} \\
&+ \sum_{(u_i,u_j)\in R_3-R_2'} p_{u_1}\cdots p_{u_N}\cdot\{E^{\pi}[C_{t+1}|d_{t+1}=u_j-y_t^j+\sum_{k\neq i,j}[y_t^k-u_k]^+] \\
&- E^{\hat{\pi}}[C_{t+1}|d_{t+1}=u_j-(y_t^j+1)+\sum_{k\neq i,j}[y_t^k-u_k]^+]\}\ \ a.s. \qquad (24)
\end{aligned}$$

Thus we have

$$\begin{aligned}
&E^{\pi}[C|\mathbf{d_1},\mathcal{F}_t] - E^{\hat{\pi}}[C|\mathbf{d_1},\mathcal{F}_t] = \\
&\sum_{(u_i,u_j)\in R_2} p_{u_1}\cdots p_{u_N}\cdot\{g_{t+1}(u_i-(y_t^j+1)+\sum_{k\neq i,j}[y_t^k-u_k]^+) - g_{t+1}(u_i-y_t^i+\sum_{k\neq i,j}[y_t^k-u_k]^+)\} \\
&+ \sum_{(u_i,u_j)\in R_3-R_2'} p_{u_1}\cdots p_{u_N}\cdot g_{t+1}(u_i-(y_t^j+1)+\sum_{k\neq i,j}[y_t^k-u_k]^+)\ \ a.s. \qquad (25)
\end{aligned}$$

Note that we have left out the superscripts π and $\pi^{'}$ in $g_{t+1}(\cdot)$. This is because both π and $\pi^{'}$ satisfy (C-1) and (C-2) from time $t+1$ on by the induction hypothesis, and are thus indistinguishable given the initial condition d_{t+1}. Using Lemma 2 and Corollary 1, the first term in Equation (25) is non-negative due to the convexity of $g_{t+1}(\cdot)$ and the assumption $y_t^i > y_t^j + 1$. The second term is also non-negative since $g_{t+1}(\cdot)$ is positive by Lemma 2. Thus we have shown that $\hat{\pi}$ incurs at most the cost incurred by π, and similar to the induction basis, $\hat{\pi}$ is "closer" to being RMF than π. If we repeat this procedure a finite number of times we can obtain a policy $\pi^{'}$ that satisfies (C-2) at time t and results in at most the cost incurred by π. The induction step is thus proved. Therefore for any policy π that satisfies (C-1) but not (C-2) we can use the above procedure going backwards in time to construct a policy $\pi^{'}$ that satisfies (C-2) and has a cost of at most that incurred by π. ∎

Note that in the above proof $R_3 - R_2' \neq \emptyset$ because $y_t^i \geq y_t^j + 2$. Indeed it can be easily verified that

$$R_3 - R_2^{'} = \{(u_i,u_j)|y_t^j + 1 \leq u_i, u_j < y_t^i\}\ .$$

Thus if $p_{y_t^j+1} > 0$, then the second term in Equation (25) is guaranteed to be strictly positive. In this case policy $\hat{\pi}$ is strictly better than policy π. This leads to the following corollary.

Corollary 2: If the arrival process has the property that $p_i > 0$ for all $i \leq \lfloor \frac{M}{N} \rfloor$, then a residual max-min fair policy $\pi^{'}$ (i.e., $\pi^{'}$ satisfies (C-2)) is strictly better than one that violates (C-2).

Proof: First note that if under a certain policy we have $y_t^i \geq \lfloor \frac{M}{N} \rfloor$ for all i and t, then this policy is residual max-min fair. Suppose policy π violates (C-2) in some time frame t, then there exist i, j such that $y_t^i \geq y_t^j + 2$ and $y_t^j < \lfloor \frac{M}{N} \rfloor$. We know that $p_{y_t^j+1} > 0$ will guarantee Equation (25) to

be strictly positive, i.e., $p_{y_i^j+1} > 0$ means that we can find a policy that is strictly better than π. By repeating the procedure used in Lemma 4, $p_i > 0$ for all $i \leq \lfloor \frac{M}{N} \rfloor$ means that we can find a residual max-min fair policy π' that is strictly better than any π that is not residual max-min fair. This in turns shows that given the property $p_i > 0$ for all $i \leq \lfloor \frac{M}{N} \rfloor$, an optimal policy is necessarily residual max-min fair, i.e., an optimal policy has to satisfy (C-2). ∎

This corollary implies that if the arrival process has the property $p_i > 0$ for all $i \leq \lfloor \frac{M}{N} \rfloor$, then (C-2) or residual max-min fair is necessary for a policy to be optimal.

Combining Lemmas 3 and 4 we obtain Theorem 1. Lemma 3 and Corollary 2 lead to Theorem 2.

IV. Sufficient But Not Necessary – Two Examples

In the previous section, we proved that any policy $\pi \in \mathcal{P}$ minimizes the cost function and therefore is optimal. However, under certain conditions on the arrival process there may be other policies that are not residual max-min fair, but are nevertheless optimal. In other words, (C-1) and (C-2) are not in general necessary for a policy to be optimal. We derived two assumptions on the arrival process under which (C-1) and (C-2) become necessary for optimality. In this section we provide two examples to illustrate that without these assumptions on the arrival process, an optimal policy need not be residual max-min fair.

Example 1: Consider two users ($N = 2$) and a time horizon $T = 2$ (i.e. the only time an allocation is made is at $t = 1$). Let $M = 5, b_0^1 = 3, b_0^2 = 2$. Suppose the arrival process is a weighted poisson process, where all probabilities are weighted by $\frac{1}{p_0}$. Thus all probabilities p_l are equal to the pmf of a poisson process conditioned on $p_0 \neq 0$ or $p_0 > 0$.

A residual max-min fair policy would assign slots to all the deterministic packets and not allocate any slots for random arrivals, i.e. $\mathbf{x} = [3, 2]$. However, one can easily check that any other policy that allocates one slot to the random arrivals for one queue and the rest of the slots to deterministic packets will have the same expected cost. For example $\mathbf{x} = [2, 3]$ and $\mathbf{x} = [4, 1]$ will both have the same expected cost as the RMF policy, although they do not satisfy (C-1). This is due to the fact that the probability of having no arrival in a frame is zero, with the weighted poisson process.

Example 2: Consider two users ($N = 2$) and a time horizon $T = 2$. Let $M = 6, b_0^1 = 2, b_0^2 = 2, p_0 = 0.5, p_1 = 0$, and $p_2 = 0.5$. A residual max-min fair policy is to let $\mathbf{x} = [3, 3]$ (i.e for each queue allocate one slot for random arrivals). Thus we have

$$E[C|\mathbf{d_0}] = E[b_1|\mathbf{d_0}] + E[b_2|\mathbf{d_0}] = 2 \times 3 + 2 \times \{0.5 \cdot (0.5 \times 2) + 0.5 \cdot (0.5 \times 1 + 0.5 \times 3)\} = 9 \ .$$

Now suppose we have an allocation $\mathbf{x} = [2, 4]$. It can be easily shown that under this policy the expected cost is also 9, although this allocation violates (C-2) and thus is not residual max-min fair.

V. Discussion and Conclusion

The main result obtained in the previous sections are highly intuitive. In essence, since queues are infinite and the only cost is packet holding cost, we should clear up the existing packets as quickly as possible in order to minimize the total cost. (C-1) implies that due to the delay of state information, we don't know for sure how many new arrivals are in the system. Therefore so long as there are known backlog in the system that need to be allocated, we should never allocate for a new arrival that may or may not be there. Since holding cost is constant and identical for all queues, all

existing backlog become equivalent. Consequently it does not matter which queue we serve first and for how much so long as we serve the deterministic backlog first. (C-2) implies that if we have to allocate for the unknown part of the queue, the best things to do is to allocate them fairly among all queues. This prevents any particular queue from becoming too large and incur extra holding cost.

It is worth mentioning that we could also discount the cost over time and define

$$C = \sum_{t=1}^{T} \beta^t \sum_{i=1}^{N} b_t^i \ ,$$

where $\beta < 1$ is the discount factor. All our lemmas and theorems still hold in this case with simple modifications to the proofs.

Below we discuss some interesting and important extensions and generalizations of the problem studied here, by relaxing some of the assumptions we have made. These are part of our on-going study.

(Assumption 1): An important extension is to drop the infinite queue assumption, and to place a penalty on dropping/blocking packets. Intuitively, the longest queue first type of policy should be optimal in this case. Note that longest queue first policy is a subset of $\mathcal{P}$ defined by (C-1) and (C-2).

(Assumption 3): If arrival processes are different for different queues, then (C-1) should still hold given the holding costs are the same. (C-2) no longer holds. Instead the allocation for random arrivals may be some form of weighted max-min fair taking into account different arrival processes.

(Assumption 5): Another interesting extension is to account for differentiated service classes represented by different holding costs, as considered in [5]. The objective is again to minimize the total expected holding cost, or the weighted sum of number of packets in the system weighted by different holding costs. Higher packet holding cost implies a higher-priority queue. In this case the allocation will have to take the arrival process into account. (Note that for the problem studied in this paper we do not need arrival statistics to determine the optimal allocation, under the assumption that all arrivals are identically distributed). We conjecture that some type of index policy may be optimal in this case.

Similarly we could consider queue connectivity probabilities and the probability of packet transmission success/failure. One may also consider the total expected discounted holding cost over an infinite horizon as the objective function to be minimized. Although not immediately obvious from the proofs shown here, our conjecture is that the same set of policies will still be optimal for the infinite horizon case.

To conclude, in this paper we considered the problem of finding an optimal policy to minimize the total number of packets/jobs in the system over a finite horizon. The features of this problem include (1) there is a large delay between when the allocation decision is made and when the allocation is used, thus decisions can be viewed as based on delayed or obsolete state information/observations; and (2) the bandwidth allocation is in frames of time slots so that each queue can be assigned any number of slots not exceeding the total number in a frame. We established two conditions under which a policy will be optimal. These two conditions define a set of policies among all admissible policies. We also argued that in general these conditions need not be necessary for the optimality of a policy. We proved that under some mild conditions on the arrival process, these two conditions are also necessary for the optimality of a policy.

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] J. C. Gittins, "Bandit processes and dynamic allocation indices," *J. Royal Statistical Society Series*, vol. B14, pp. 148–167, 1972.

[10] R. R. Weber, "On the gittins index for multi-armed bandits," *The Annals of Applied Probability*, vol. 2, pp. 1024–1033, 1994.

[11] P. Whittle, "Multi-armed bandits and the gittins index," *Journal of the Royal Statistical Society*, vol. 42, no. 2, pp. 143–149, 1980.

[12] D. Bertsimas and J. Nino-Mora, "Conversion laws, extended polymatroids and multi-armed bandit problems," *Mathematics of Operations Research*, vol. 21, pp. 257–306, 1996.

[13] E. Frostig and G. Weiss, "Four proofs of gittins' multi-armed bandit theorem," *Applied Probability Trust*, November 1999.

[14] T. Ishikida, "Informational aspects of decentralized resource allocation," *PhD. Thesis, University of California, Berkeley*, 1992.

[15] D. G. Pandelis and D. Teneketzis, "On the optimality of the gittins index rule for multi-armed bandits with multiple plays," *Mathematical Methods of Operations Research*, vol. 50, pp. 449–461, 1990.

[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.