Utility Optimal Scheduling for General Reward States and Stability Constraint

Prasanna Chaporkar and Saswati Sarkar
Department of Electrical and Systems Engineering
University of Pennsylvania
{prasanna@seas,swati@ee}.upenn.edu

Abstract

We consider a queueing system with n parallel queues, which receives a reward for the service it provides. Our aim is to maximize the expected reward obtained per unit time (*utility*) while ensuring that the mean queue length in each of the queues is bounded (*stability*). We show that the optimal policy has counter intuitive properties because of the general reward states and stability constraint. For example, the greedy policy of serving a customer that fetches maximum reward need not be optimal. In addition, the optimal policy may belong to a class of non work-conserving policies. We obtain two different policies that attain the above optimality goal. The first policy arbitrates service randomly based on the current reward states and probabilities that depend on system statistics. The second policy arbitrates service deterministically based only on the queue lengths and the current reward states, and does not require any knowledge of the system statistics. The proposed policies are optimal in a large class of policies that includes off-line policies, which use knowledge of past, present and even future arrival and reward states in their decision processes.

Index Terms

Queueing theory, utility maximization, stability, randomized algorithms.

I. Introduction

We consider the problem of utility maximization in a queueing system with n parallel queues (Figure 1). Customers arrive at random, and are queued in one of the n buffers depending on the class to which they belong. Time is slotted. Each customer can be served in a single slot. The server S achieves certain reward for the service it provides. The reward depends on *which* class is served and *when* it is served, i.e., if S serves a customer from class k in slot t, then it achieves reward $r_k(t)$. The reward $r_k(t)$ is random and drawn from a finite and bounded sample space for every k. The rewards for different classes can be correlated in and across slots. The server at the queueing system decides

This work was supported by the National Science Foundation under grants ANI-0106984, NCR-0238340 and CNS-0435306.

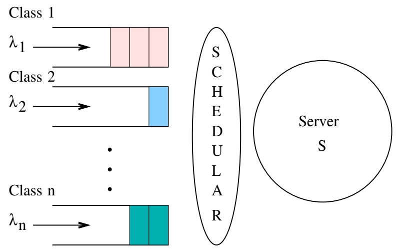

Fig. 1. Figure shows a queueing system under consideration. Each queue corresponds to a particular customer class. In every slot a server chooses a customer class to serve. The server can serve one customer in every slot.

whether to serve, and if it decides to serve, then which class to serve. We allow service to be non work-conserving, i.e., the server may decide not to serve even when the customers are waiting in the queues. *Utility* of a server is the expected reward it achieves per unit time. *Our goal is to design a scheduling policy that maximizes utility among all stable policies*. The system is *stable* if the mean queue length for every class is bounded. Ensuring stability is essential as it guarantees bounded queue lengths, which in turn implies bounded delays. In addition, stability limits loss due to buffer overflow in systems with finite buffer.

In order to achieve the maximum utility, a server has to solve the following decision problems: (1) when to serve, (2) which class to schedule, and (3) how much information it should maintain. To see that the first question is important, we note that if S serves class k when the reward ($r_k(t)$) is small, then the system utility may be small. On the other hand, if S waits for a large value of reward, then the service rate may be small, and hence the system may become unstable. The challenge in solving the second decision problem is that several intuitive policies turn out to be suboptimal. For example, the greedy policy that always serves a class with the largest reward value in a slot may not be optimal. Now, we explain why the third decision problem is important. Note that the arrivals and rewards for various classes can be arbitrarily correlated in and across slots. Thus, the past observations of the arrivals and reward states can potentially be used in the decision process so as to improve the utility. Hence the server has to decide how much information about the past should be retained, and how this information should be used. *Our goal is to resolve these decision problems optimally.*

Now, we describe two scheduling problems in wireless networks which can be modeled as the stochastic control problem stated above. We subsequently outline our contributions.

A. Multicast at Medium Access Layer

Consider a node S with an omni-directional antenna. Let n multicast sessions traverse S (Figure 2). Packets for each session $k \in \{1, \ldots, n\}$ arrive randomly, and are stored in separate queues. The sender

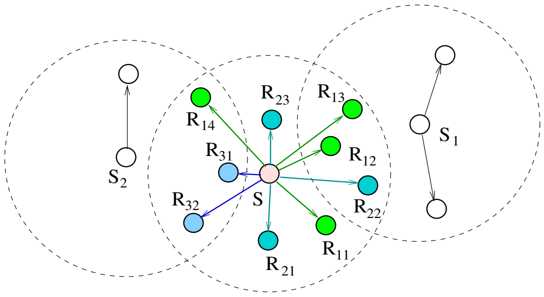

Fig. 2. Figure shows three medium access layer multicast sessions from node S: (1) S to receivers R_{11}, R_{12}, R_{13} and R_{14}, (2) S to receivers R_{21}, R_{22} and R_{23}, and (3) S to receivers R_{31} and R_{32}. The dashed circles indicate transmission ranges of the senders S, S_1 and S_2.

S has to deliver each packet for session k to G_k receivers (k's *multicast group*) in its transmission range. Let S transmit only one packet in a slot. Because of the broadcast nature of wireless communication, a single transmission from S can be intercepted by all the nodes in its transmission range. Hence, if S transmits session k's packet, then the packet can be simultaneously intercepted by all G_k receivers in the multicast group. Though the broadcast nature of wireless medium can be used to improve the bandwidth efficiency of wireless multicast, it also imposes critical challenges. A multicast specific challenge is that *some* but *not all* receivers may be ready to receive. This happens due to interference from other transmissions in the receivers' neighborhood, location dependent channel errors and power saving operations. For example, in Figure 2, R_{12}, R_{13} and R_{22} can not receive a transmission from S whenever S_1 is transmitting a packet as both the transmissions will collide at these receivers. But, all the other receivers can receive the transmission from S if S_2 is not transmitting. Let $r_k(t)$ receivers of session k be ready in slot t, where $r_k(t) \in \{0, \ldots, G_k\}$. Thus, if S transmits session k's packet in slot t, then it will receive reward $r_k(t)$. For example, in Figure 2, if S_1 and S_2 are transmitting in slot t, then the rewards possible for the three sessions are 1, 2 and 0, respectively. When only S_2 is transmitting, the rewards for the sessions are 3, 3 and 0, respectively. Note that the readiness states of the receivers can be arbitrarily correlated in and across slots, e.g., in Figure 2, when S_2 is transmitting, R_{14}, R_{31} and R_{32} are simultaneously not ready. The throughput of a policy is the expected reward it achieves per unit time [1], [2], [3], [4]. Thus, the policy obtained here maximizes throughput subject to stability.

B. Unicast at Medium Access Layer

Consider a node S with n unicast sessions to receivers R_1 to R_n. Packets arrive at S randomly (Figure 3). The packets corresponding to different sessions are queued in separate queues. Node S can transmit only one packet in a slot through a wireless channel. Since the wireless channels experience location dependent fading, the channel capacity varies randomly, and may be different for different

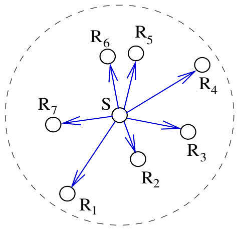

Fig. 3. Figure shows seven unicast sessions from S to receivers R_1 to R_7. The dashed circle indicates the transmission range of S.

sessions in the same slot. The variation in channel capacity is modeled by a Markov chain with finite number of states [5]. A state of this Markov chain corresponds to the probability that the transmission will be successful. Let $r_k(t)$ denote the probability that the transmission for session k in slot t is successful. Note that $r_k(t)$s can be correlated arbitrarily in and across slots depending on the location and mobility of the receivers, e.g., in Figure 3, R_5 and R_6 are likely to suffer from similar fading levels as they are close to each other. Now, if S serves session k in slot t its expected reward is $r_k(t)$ in slot t. Again, throughput is the number of packets delivered successfully per unit time which equals the reward obtained per unit time. Thus, the policy obtained here maximizes throughput among all stable policies.

C. *Our Contributions*

We present our detailed system model in Section II. We state various challenges in designing an optimal policy in Section III. Here, we demonstrate that the optimal policy has the following counter intuitive properties: (a) the optimal policy may belong to the class of non work-conserving policies (b) a policy that serves class k only when the maximum reward for k can be obtained may render the system unstable, and (c) the greedy policy that always serves a class with the largest reward value in a slot may not be optimal. We propose two optimal policies in Section IV. First, we present a randomized policy that serves each class k with a probability w_k that depends on the reward states in the current slot (Section IV-A). We present a linear program that computes the w_k's. The computation of w_k's does not require any information about the correlations in the arrival and reward processes, but only requires first order statistics, namely, the arrival rates and the steady state distribution of the reward process. Note that the first order statistics is often easier to obtain than the correlations, but may still not be available in some systems. Hence, we next propose a statistic oblivious optimal policy that decides the schedule in each slot based on the current queue lengths and the current reward states (Section IV-B). In spite of deciding transmissions using only the current system state, the proposed policies maximize utility in a large class of scheduling policies that includes off-line scheduling policies, which use the knowledge of past, present and even the future arrivals and reward states in their decision process. The optimal policy may not be fair as it may not provide any guarantee on the performance of individual customer classes. In Section V, we generalize the randomized optimal policy to maximize the overall system utility subject to stabilizing the system and attaining certain desired minimum utilities for individual classes. We present the related work in Section VI. We conclude in Section VII We present proofs for all the results in [6].

II. System Model

We consider a queueing system with n-parallel queues at server S (Figure 1). The arriving customers can belong to one of n possible classes. The customers that belong to class k are queued in the k^{th} queue. Each queue has infinite buffer. We assume that time is slotted. The customers arrive as per a random process $\{\mathbf{\Lambda}(t) = (\Lambda_1(t), \ldots, \Lambda_n(t)) : t \geq 0\}$, where $\Lambda_k(t)$ is the number of arrivals in slot t for class k. We assume that $\{\mathbf{\Lambda}(t), t \geq 0\}$ is ergodic and satisfies the following property.

Assumption 1: There exists a vector $\vec{\lambda} = (\lambda_1, \ldots, \lambda_n)$ such that the empirical average of the arrivals in the system converges to $\vec{\lambda}$ at rate faster than $\frac{1}{t^\alpha}$ for some $\alpha > 1$. Mathematically, for every $k \in \{1, \ldots, n\}$ and $\delta > 0$, there exist $\widehat{t_\delta}$ and $\alpha > 1$ such that for every $T \geq \widehat{t_\delta}$,

$$\mathbf{P}\left\{\left|\frac{\sum_{t=1}^{T}\Lambda_k(t)}{T} - \lambda_k\right| > \delta\right\} < \frac{1}{T^\alpha}. \tag{1}$$

We refer to $\vec{\lambda}$ as the arrival rate vector.

We assume that the arrivals occur at the beginning of a slot, and the arriving customer can be served in the same slot. Server S can serve exactly one customer in a slot, and each customer leaves the queue after receiving service once. In every slot, S decides whether to serve, and if it decides to serve then which class to serve. The queueing system achieves reward for the service it provides. Random variable $r_k(t)$ denotes a reward that the system can achieve by scheduling class k in slot t. The reward process $\{\mathbf{R}(t) = (r_1(t), \ldots, r_n(t)) : t \geq 0\}$ is also assumed to be ergodic. Let $\mathcal{S}$ be the set of all possible reward vectors, i.e., $\mathbf{R}(t) \in \mathcal{S}$ for every t. We assume that $\mathcal{S} = \{\mathbf{R}_1, \ldots, \mathbf{R}_m\}$, where $m < \infty$. Now, r_{ik} is the reward that the system achieves for scheduling the k^{th} class in slot t if $\mathbf{R}(t) = \mathbf{R}_i$. We assume that r_{ik} is non-negative and bounded for every i and k. Let $r_k^{\max} = \max_{1\leq i\leq m}\{r_{ik}\}$ and $r^{\max} = \max_{1\leq k\leq n}\{r_k^{\max}\}$. We assume that $\{\mathbf{R}(t), t \geq 0\}$ satisfies the following property.

Assumption 2: There exists a vector $\mathbf{B} = (b_1, \ldots, b_m)$ such that the empirical distribution of the reward process converges to $\mathbf{B}$ at rate faster than $\frac{1}{t^\alpha}$ for some $\alpha > 1$. Mathematically, for every $i \in \{1, \ldots, m\}$

and $\delta > 0$, there exist $\widehat{t_\delta}$ and $\alpha > 1$ such that for every $T \geq \widehat{t_\delta}$,

$$\mathbf{P}\left\{\left|\frac{\sum_{t=1}^{T} \mathbf{1}_{\{\mathbf{R}(t)=\mathbf{R}_i\}}(t)}{T} - b_i\right| > \delta\right\} < \frac{1}{T^\alpha}. \quad (2)$$

Here, $\mathbf{1}_A$ is the indicator of event A.

We refer to $\mathbf{B}$ as the steady state distribution of the reward process. Furthermore, we assume that the reward process does not depend on the queue lengths in the system.

Assumptions 1 and 2 are satisfied by a large class of random processes including the i.i.d and ergodic Markov processes defined on a finite state space. Next, we present some important definitions.

Definition 1 (Utility): Utility of class k under a scheduling policy Δ (denoted as U_k^Δ) is the expected reward received for serving the k^{th} class per unit time. The system utility under scheduling policy Δ (denoted as U^Δ) is the sum of utilities of all the classes under Δ, i.e., $U^\Delta = \sum_{k=1}^{n} U_k^\Delta$.

Definition 2 (System Stability): The queueing system is said to be stable if the mean queue length for every class is finite. A scheduling policy that stabilizes the system is called a *stable policy*.

Definition 3 (Stability Region): Stability region Ω is the set of arrival rates $\vec{\lambda}$ for which some policy can stabilize the system. A policy Δ is said to maximize the stability region if for every $\vec{\lambda} \in \Omega$, Δ is stable.

Definition 4 (Optimality): A stable scheduling policy Δ is said to be optimal if it maximizes the utility among all the stable scheduling policies.

Note that the optimality is defined in a large class of scheduling policies that includes off-line policies that take into account past, present and even future arrivals and reward states in their decision process.

Definition 5 (ϵ-Optimalily): A scheduling policy Δ is said to be ϵ-optimal for some $\epsilon > 0$ if (a) it is stable and (b) no other stable policy can achieve utility more than ϵ plus the utility achieved by Δ.

We seek to obtain an ϵ-optimal policy for any given $\epsilon > 0$. Note that the maximum service rate is one customer per slot. Hence, if the expected arrival rate in the queueing system is greater than one ($\sum_{k=1}^{m} \lambda_k > 1$), then no scheduling policy is stable. So, we assume that $\sum_{k=1}^{m} \lambda_k < 1$.

III. Challenges in Designing Optimal Policy

We demonstrate that the optimal policy has many counter intuitive properties. Let us consider the following example.

Example 1: Let $n = 1$, and the arrival process be Bernoulli with rate $0.5-\epsilon$, i.e., in every slot a customer arrives independently with probability (w.p.) $0.5 - \epsilon$ for arbitrarily small $\epsilon > 0$. Let the probability of achieving rewards 1, 2 and 3 in any given slot be $\frac{1}{3}$ each. A slot in which the queue is non-empty is referred to as a busy slot. We consider three policies: (a) server serves in every busy slot, (b) server serves in every busy slot in which the reward value is 3, and (c) server always serves in a busy slot in which reward is 3 and w.p. 0.5 serves in a busy slot in which reward is 2. Note that policy (a) is the only work-conserving policy in this system. Now, policy (a) maximizes the stability region, and its utility is approximately 1. Policy (b) serves only when maximum reward is achieved and therefore maximizes the expected reward obtained per customer, but it is not stable as the service rate is $\frac{1}{3}$ which is less than the arrival rate. Policy (c) is stable and its utility is approximately $\frac{4}{3}$. This policy, however, is unstable for arrival rates greater than 0.5.

Example 1 demonstrates that *no work-conserving policy may maximize utility*. We now explain why this is the case. If the system serves class k at a rate much higher than λ_k, then the queue for class k will be empty in most of the slots. In this duration, the system cannot take advantage of large $r_k(t)$'s as k's queue is empty. Instead, if the system serves every class k at a rate just higher than λ_k, like Policy (c) in Example 1, then the number of slots in which k's queue is empty will be small, and hence the system can potentially achieve better utility by serving k only when $r_k(t)$ is large.

Moreover, note that the work-conserving policies maximize the stability region. Thus, *some policies that maximize the stability region may not maximize the utility.*

Now, consider a policy that serves a class k only when the highest reward for k can be achieved, e.g., Policy (b) in Example 1. The above example demonstrates that *such policies may render the system unstable even though they maximize the reward obtained per packet.* This is because the maximum reward may arrive at a rate smaller than the arrival rate of the class, and thus this policy will provide a service rate which is less than the arrival rate. Therefore the system will be unstable.

Example 2: Let $n = 2$, and let the arrival processes for both the classes be Bernoulli with rate $0.5-\epsilon$. Here, ϵ is a small positive constant. Furthermore, let $m = 2$, and $\mathbf{R}_1 = (20, 15)$ and $\mathbf{R}_2 = (19, 11)$. Let probability that the server observes $\mathbf{R}_1$ in any slot be 0.5. We consider two transmission policies: (a) the greedy policy that serves a non-empty queue with the largest reward value, and (b) a policy that serves class 1 (2, resp.), if reward vector is $\mathbf{R}_2$ ($\mathbf{R}_1$, resp.). Note that both the policies are stable. The greedy policy will always serve class 1 whenever its queue is not empty, and in the remaining slots it will serve class 2. The utility of policy (a) is approximately 16.25, while that of policy (b) is approximately 17.

Example 2 demonstrates that *the greedy policy that selects the class with the largest reward in a slot need not be optimal.*

IV. ϵ-OPTIMAL POLICY

In this section, we design two ϵ-optimal policies. In Subsection IV-A, we propose a statistics dependent randomized policy, and in Subsection IV-B, we propose a statistic oblivious deterministic policy.

A. *Randomized Policy (Δ^*)*

We now describe a randomized policy Δ^* that schedules a class with a probability which depends on the state of the reward process in the current slot. Consider the following linear program.

LP(δ) :- Maximize: $\Phi(\delta) = \sum_{k=1}^{n} \sum_{i=1}^{m} w_{ik} r_{ik}$

Subject to:

1) $\sum_{i=1}^{m} w_{ik} = \lambda_k + \delta$ for every $k = 1, 2, \ldots, n$
2) $\sum_{k=1}^{n} w_{ik} \leq b_i$ for every $i = 1, 2, \ldots, m$
3) $w_{ik} \geq 0$ for every i and k

Note: The linear program **LP(δ)** has a feasible solution, if $\sum_{k=1}^{n} (\lambda_k + \delta) < 1$. Such positive δ can always be found, e.g., $\delta = \left(1 - \sum_{k=1}^{n} \lambda_k\right) / (n+1)$.

Using the optimal solution of **LP(δ)**, $w_{ik}^*(\delta)$, we now describe the scheduling policy $\Delta^*(\delta)$.

- Whenever $\mathbf{R}(t) = \mathbf{R}_i$, choose queue k w.p. $\frac{w_{ik}^*(\delta)}{b_i}$.
- If the chosen queue is non empty, then serve it; otherwise remain idle.

Next, we formally state the stability properties of $\Delta^*(\delta)$.

Theorem 1: For every positive δ that satisfies $\sum_{k=1}^{n} (\lambda_k + \delta) < 1$, $\Delta^*(\delta)$ is stable. Moreover, for every $\vec{\lambda} \in \Omega$, there exists a $\delta > 0$ such that $\Delta^*(\delta)$ is stable.

Intuition: The quantity w_{ik}/b_i is the probability that the k^{th} class is scheduled conditioned on $\mathbf{R}(t) = \mathbf{R}_i$. Constraints 2) and 3) in $\mathbf{LP}(\delta)$ ensure that $w_{ik}(\delta)/b_i$ is a valid probability, i.e., $0 \leq w_{ik}(\delta)/b_i \leq 1$ for every k and i. In steady state, $\mathbf{P}\{\mathbf{R}(t) = \mathbf{R}_i\} = b_i$. Thus, probability that $\mathbf{R}(t) = \mathbf{R}_i$ and the k^{th} class is scheduled is w_{ik}. Now, constraint 1) in $\mathbf{LP}(\delta)$ ensures that the probability that the k^{th} class is scheduled is equal to $\lambda_k + \delta$ for $\delta > 0$ for all k. Thus, the steady state service rate for every class under $\Delta^*(\delta)$ is strictly greater than its arrival rate. Thus, $\Delta^*(\delta)$ is stable. Now, recall that Ω is the set of $\vec{\lambda}$ such that $\sum_{k=1}^{n} \lambda_k < 1$. Thus, for every $\vec{\lambda} \in \Omega$, there exists $\delta > 0$ such that $\sum_{k=1}^{n} (\lambda_k + \delta) < 1$ and clearly for such a δ $\Delta^*(\delta)$ is stable.

The values of δ that ensure stability are functions of arrival rates and steady state distributions of the readiness process only and do not depend on the correlation between the arrivals and that between the readiness states.

Next, we show that $\Delta^*(\delta)$ is ϵ-optimal.

Theorem 2: For any $\epsilon > 0$, if

$$\delta \in \left(0, \min\left\{\frac{1-\sum_{k=1}^{n}\lambda_k}{n}, \frac{\epsilon}{nr^{\max}}\right\}\right), \tag{3}$$

then $\Delta^*(\delta)$ is ϵ-optimal. Moreover, utility of class k is

$$U_k^{\Delta^*(\delta)} \geq \sum_{i=1}^{m} w_{ik}^*(\delta) r_{ik} - \delta r_k^{\max} \qquad \text{w.p. } 1 \tag{4}$$

Intuition: If class k always has a customer waiting in the queue, then from the stationarity of $\Delta^*(\delta)$ and the ergodicity of the reward process $\mathbf{R}(t)$ it follows that the utility of class k under $\Delta^*(\delta)$ is equal to $\sum_{i=1}^{m} w_{ik}^*(\delta) r_{ik}$. Since the service rate is δ more than the arrival rate for class k, intuitively, the fraction of slots in which k is scheduled but has empty queue is equal to δ. In these slots, the server remains idle and zero reward is achieved. Hence, the utility of class k under $\Delta^*(\delta)$ is equal to $\sum_{i=1}^{m} w_{ik}^*(\delta) r_{ik}$ minus the expected reward lost per unit time on account of class k's queue being empty. The expected reward lost per unit time in these intervals is at most $r_k^{\max}\delta$ as the fraction of slots in which k's queue is empty is δ. Thus, (4) follows. Now, the expected reward per unit time of any stable policy can be expressed as $\sum_{k=1}^{n}\sum_{i=1}^{m} w_{ik}(\delta) r_{ik}$ where $w_{ik}(\delta)$ satisfy the constraints in **LP**(δ) for some $\delta > 0$. Furthermore, $w_{ik}(\delta)$ are chosen so as to maximize $\sum_{k=1}^{n}\sum_{i=1}^{m} w_{ik}(\delta) r_{ik}$. Thus, ϵ-optimality of $\Delta^*(\delta)$ follows from (4).

B. *Online Policy (Δ_O)*

The randomized policy Δ^* requires the knowledge of $\vec{\lambda}$ and $\mathbf{B}$ in order to obtain the optimal $\vec{w}$. Such knowledge may not be readily available in the system. We propose a parametrized online policy Δ_O that achieves the optimum utility without requiring the knowledge of $\vec{\lambda}$ or $\mathbf{B}$. The parameter for Δ_O is denoted by V, where V can be appropriately chosen. Here, we require the additional assumption that the arrival process $\mathbf{\Lambda}(t)$ is independent and identically distributed across the slots and the reward process $\mathbf{R}(t)$ is an irreducible and aperiodic Markov chain. We still allow the arrivals and rewards for various classes to be arbitrarily correlated. Let $\vec{Q}(t) = (Q_1(t), \ldots, Q_n(t))$ be the queue length vector at the beginning of the t^{th} slot. Then, a class k is scheduled under policy $\Delta_O(V)$ in slot t if

- $Q_k(t) - V \times (r_k^{\max} - r_k(t)) \geq 0$, and
- $k \in \arg\max_{1\leq j\leq n}\left\{Q_j(t) - V \times (r_j^{\max} - r_j(t))\right\}$.

We have the following performance guarantees for $\Delta_O(V)$.

Theorem 3: For every $V > 0$, $\Delta_O(V)$ is stable. Moreover, for every $\epsilon > 0$, there exists a $V_1 > 0$ such that for every $V > V_1$, $\Delta_O(V)$ is ϵ-optimal.

Intuition: When $\sum_{k=1}^{n} \lambda_k < 1$, then the sum of queue lengths has a negative drift under any work-conserving policy. Thus, every work-conserving policy is stable. Now, if queue length of any class k is greater than $Vr_k^{\max}$, then $\Delta_O(V)$ schedules some queue. Thus, $\Delta_O(V)$ has service rate higher than the arrival rate in system except when the queue lengths in all the queues are small, i.e., when $Q_k(t) < Vr_k^{\max}$ for every k. Thus, sum of the queue lengths has a negative drift when the queue lengths are large. Hence, $\Delta_O(V)$ stabilizes the system for every $V > 0$.

We now explain why $\Delta_O(V)$ maximizes the utility. For simplicity, we consider a special case in which the reward values are integer and belong to $\{1, \ldots, r_k^{\max}\}$ for class k. To attain optimality subject to stability, a scheduling policy should obtain the maximum possible reward for every packet while maintaining the service rate greater than the arrival rate. But, $\vec{\lambda}$ is not known. Hence, one option is to adjust the service rate based on the queue lengths. Thus, when Q_k is small, the scheduling policy can wait longer without violating stability, and hence it should schedule the k^{th} queue only if the achievable reward is high. On the other hand, if Q_k is large, in order to preserve stability, the policy should schedule the k^{th} queue at a higher rate even at the cost of achieving low reward.

Now, Δ_O obtains the schedule as per the above intuition. For example, when $Q_k \leq V$, $Q_k - V(r_k^{\max} - r_k) > 0$ only when achievable reward (r_k) is equal to $r_k^{\max}$. Thus, $\Delta_O(V)$ will schedule the k^{th} queue only if the maximum possible reward is achievable. Now, if $Q_k \in \{V+1, \ldots, 2V\}$, then $Q_k - V(r_k^{\max} - r_k) \geq 0$ only when the achievable reward is greater than or equal to $r_k^{\max} - 1$. Thus, Δ_O will schedule the k^{th} queue only if the achievable reward is greater than or equal to $r_k^{\max} - 1$. Similarly, if $Q_k \in \{(r_k^{\max} - u)V + 1, \ldots, (r_k^{\max} - u + 1)V\}$, then Δ_O will schedule the k^{th} queue only if the achievable reward is greater than or equal to u. Thus, intuitively, Δ_O achieves the largest possible reward per packet constrained to stability, and hence it is optimal.

V. Maximizing Utility among Fair Policies

We now generalize our optimization goal so as to maximize the overall system utility subject to (a) stabilizing the system and (b) providing every class k a minimum utility F_k. We show that the randomized scheduling policy Δ^* can be extended to attain this goal. We first introduce the following notation.

Let $\vec{F} = \{F_1, \ldots, F_n\}$. Let $\mathcal{C}_{\vec{F}}$ denote the set of stable scheduling policies Δ that guarantee $U_k^{\Delta} \geq F_k$ for every $k \in \{1, \ldots, n\}$. Let $\vec{\epsilon}$ be the n dimensional vector whose every element is ϵ.

Definition 6 (ϵ-fair): A policy Δ is said to be ϵ-fair if (a) Δ is stable, (b) $\Delta \in \mathcal{C}_{\vec{F}-\vec{\epsilon}}$ and (c) no policy in $\mathcal{C}_{\vec{F}}$ can achieve utility more than ϵ plus the utility of Δ.

Now, consider a policy $\Delta_1^*(\delta)$ which is similar to $\Delta^*(\delta)$ except that the linear program for computing the optimal values of $w_{ik}(\delta)$ has the following additional constraints (henceforth referred as *fairness constraints*):

$$\sum_{i=1}^{m} r_{ik} w_{ik} \geq F_k \text{ for every } k = 1, 2, \ldots n.$$

We refer to $\mathbf{LP}(\delta)$ with the above fairness constraints as $\mathbf{LP_1}(\delta)$. Now, $\Delta_1^*(\delta)$ attains the following performance guarantee.

Theorem 4: If $\mathcal{C}_{\vec{F}}$ is not empty, then for every $\epsilon > 0$ there exists a $\widehat{\delta} > 0$ such that for every $\delta \in (0, \widehat{\delta})$, $\Delta_1^*(\delta)$ is ϵ-fair.

The policy Δ^* can also be extended to maximize the minimum utility obtained by a class among all stable policies.

VI. Related Work

Existing research in the area of maximizing utility in a system with n parallel queues can be broadly classified into two categories. In the first category, the premise is that every customer fetches unit reward, but the number of customers that can be served for a class in a slot is different for different classes and varies randomly with time [7], [8], [9], [10], [11], [12], [13]. In this scenario, the utility of any stable scheduling policy is the sum of arrival rates over all the classes. Since every stable policy provides the same utility, the focus in this area has been to obtain policies that maximize the stability region. Typically, the proposed policies do not require statistics of arrival and reward processes, but assume the knowledge of instantaneous reward states and queue lengths in every slot. Tassiulas *et. al.* have characterized the stability region of the system, and have obtained scheduling policies that maximizes the stability region [13], [14]. In [7], [10], [11], [12], [15], authors have obtained policies that provide certain delay characteristics in addition to maximizing the stability region.

Recently, considerable research efforts have been directed towards designing a policy that maximizes utility when different packets achieve different rewards [16], [17], [18], [19], [20], [21], [22]. In this area, the premise is that the system is saturated, i.e., every class always has a customer waiting in the queue. In saturated systems, a greedy policy of serving a customer that fetches maximum reward in a slot is clearly optimum. But, the greedy policy may provide poor quality of service to some customer classes. Hence, the utility maximization with various fairness constraints have been studied.

Our framework considers utility maximization in systems with arrivals and general reward states. Thus, our work provides a bridge between the above two different sets of problems.

Now, we describe the work that is most closely related to our work. In our prior work, we have considered the similar problems in the special case that the system has only one class (i.e., when

$n = 1$). We have shown that in such systems a policy that maximizes the stability region need not maximize the utility [1], [2], [4]. We have also shown that threshold based policies maximize utility subject to stability [1], [4]. A threshold based policy chooses threshold T in every slot, and transmits only if the achievable reward is greater than or equal to T. Recently, Neely [23] has considered a queueing system in which in each slot different queues can be simultaneously served at different rates. The rate vector can be selected among some given choices, and different selections have different costs. In this scenario, Neely has proposed a scheduling policy that minimizes the cost while stabilizing the system. In our case, scheduled queues must be served at the same rate, but receive different rewards that depend on the state of readiness process. We maximize the total reward achieved per unit time subject to stability. Thus, in some sense, we study the dual of the problem studied in [23].

VII. Conclusion

We have considered the problem of utility maximization constrained to stability for general reward states, and proposed two utility optimal policies. In the course of this exposition we, however, have made some simplifying assumptions, which we elaborate on next. Our simplifying assumption was that the reward process does not depend on the scheduling policy. But, in many practical applications this may not be the case. For example, in multicast at medium access level (Section I-A), it may be possible to design a scheduling policy that generates favorable receiver readiness states and thereby improve throughput. Designing such a policy is possible if the senders coordinate their transmissions. In [24], we propose utility optimal policies in a closed loop dynamical system where the reward process and scheduling policy affect each other.

Another simplifying assumption we made was that each customer can be served in a single slot. All the results in the paper can be generalized to accommodate any independent and identically distributed service times.

Appendix

First, we introduce additional notation that will be used throughout this section. We denote a sample path by ω. Let $\eta_i(t)$ denote the number of slots until t with reward vector $\mathbf{R}_i$. Note that $\sum_{i=1}^{m} \eta_i(t) = t$ for every t. Recall that a customer class k is served only if k is scheduled and its queue is non-empty. We define $\xi_{ik}^{\Delta}(\mathcal{B}, t)$ to be the number of slots until t such that the reward vector is $\mathbf{R}_i$, class k is scheduled under Δ and its queue is non-empty. Similarly, we define $\xi_{ik}^{\Delta}(\mathcal{I}, t)$ to be the number of slots until t such that the reward vector is $\mathbf{R}_i$, class k is scheduled under Δ but its queue is empty. Next, we derive some properties of these variables.

Property 1: For every $i \in \{1, \ldots, m\}$,

$$\lim_{t \to \infty} \frac{\eta_i(t)}{t} = b_i \quad \text{w.p. } 1.$$

Proof: Fix any $\delta > 0$. Let an event E_t be defined as follows for every $t \in \{1, 2, \ldots, \}$.

$$E_t = \left\{ \omega : \left| \frac{\eta_i(t)}{t} - b_i \right| > \delta \right\}.$$

Now, by (2), it follows that

$$\sum_{t=1}^{\infty} \mathbf{P}\{E_t\} < \infty.$$

Thus, by Borel-Cantelli Lemma,

$$\mathbf{P}\left\{E_t \text{ infinitely often}\right\} = 0. \tag{5}$$

Since, (5) holds for every $\delta > 0$, the result follows. ∎

Property 2: Let $\xi_{ik}^{\Delta}(t) \stackrel{\text{def}}{=} \xi_{ik}^{\Delta}(\mathcal{B}, t) + \xi_{ik}^{\Delta}(\mathcal{I}, t)$. Then,

$$\lim_{t \to \infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(t)}{t} = w_{ik}(\delta) \quad \text{w.p. } 1$$

Proof: We note that

$$\lim_{t \to \infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(t)}{t} = \lim_{t \to \infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(t)}{\eta_i(t)} \lim_{t \to \infty} \frac{\eta_i(t)}{t}. \tag{6}$$

Under $\Delta^*(\delta)$, class k is scheduled with probability $w_{ik}(\delta)/b_i$ in every slot independent of the previous scheduling decisions whenever the reward vector is $\mathbf{R}_i$. Thus, by Kolmogorov's Strong Law of Large Numbers (SLLN),

$$\lim_{t \to \infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(t)}{\eta_i(t)} = \frac{w_{ik}(\delta)}{b_i} \quad \text{w.p. } 1. \tag{7}$$

Thus, the result follows from Property 1, (6) and (7). ∎

Property 3: If $\Delta^*(\delta)$ is stable, then for every $k \in \{1, \ldots, n\}$

$$\lim_{t \to \infty} \frac{\sum_{i=1}^{m} \xi_{ik}^{\Delta^*(\delta)}(\mathcal{B}, t)}{t} = \lambda_k \quad \text{w.p. } 1 \tag{8}$$

$$\lim_{t \to \infty} \frac{\sum_{i=1}^{m} \xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t)}{t} = \delta \quad \text{w.p. } 1. \tag{9}$$

Proof: Note that $\sum_{i=1}^{m} \xi_{ik}^{\Delta^*(\delta)}$ is the number of class k's customers served until t under $\Delta^*(\delta)$. Since, the system is stable, (8) follows. Now, for every $1 \le i \le m$, $1 \le k \le n$ and $t \ge 1$,

$$\xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t) = \xi_{ik}^{\Delta^*(\delta)}(t) - \xi_{ik}^{\Delta^*(\delta)}(\mathcal{B}, t).$$

Thus,

$$\lim_{t \to \infty} \frac{\sum_{i=1}^{m} \xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t)}{t} = \lim_{t \to \infty} \frac{\sum_{i=1}^{m} \xi_{ik}^{\Delta^*(\delta)}(t)}{t} - \lim_{t \to \infty} \frac{\sum_{i=1}^{m} \xi_{ik}^{\Delta^*(\delta)}(\mathcal{B}, t)}{t}.$$

From Property 2, it follows that

$$\begin{aligned}\lim_{t\to\infty}\frac{\sum_{i=1}^{m}\xi_{ik}^{\Delta^*(\delta)}(t)}{t} &= \sum_{i=1}^{m} w_{ik}(\delta) \quad \text{w.p. } 1\\ &= \lambda_k+\delta. \end{aligned} \tag{10}$$

Thus, (9) follows from (8) and (10). ■

Next, we prove Theorem 1.

Proof of Theorem 1

Proof: To prove Theorem 1, it suffices to show that the length of busy period is finite for every class k irrespective of the state of reward process at the beginning of the busy period and the queue lengths of the other classes given that $\sum_{k=1}^{n}\lambda_k+\delta<1$. The *Busy period* for a class is defined as the largest contiguous time interval for which its queue length is greater than 0.

Note that if $\sum_{k=1}^{n}\lambda_k+\delta<1$, then **LP**($\delta$) has a feasible solution, and hence $\Delta^*(\delta)$ is well defined. Let $\mathcal{BP}_k$ be a random variable denoting the length of busy period of k^{th} class under $\Delta^*(\delta)$. Without loss of generality, we assume that a typical busy period for class k starts in slot 1. We observe that for every $T\geq 1$, $\mathcal{BP}_k\geq T$ if and only if the number of customers arrived until t is greater than the number of times k is scheduled under $\Delta^*(\delta)$ until t for every $t\leq T$. Thus,

$$\begin{aligned}
\mathbf{P}\{\mathcal{BP}_k\geq T\} &= \mathbf{P}\left\{\bigcap_{l=1}^{T}\left\{\left(\sum_{t=1}^{l}\Lambda_k(t)-\xi_k^{\Delta^*(\delta)}(l)\right)>0\right\}\right\}\\
&\leq \mathbf{P}\left\{\left(\sum_{t=1}^{T}\Lambda_k(t)-\xi_k^{\Delta^*(\delta)}(T)\right)>0\right\}\\
&= \mathbf{P}\left\{\left(\frac{1}{T}\sum_{t=1}^{T}\Lambda_k(t)-\frac{\xi_k^{\Delta^*(\delta)}(T)}{T}\right)>0\right\}\\
&= 1-\mathbf{P}\left\{\left(\frac{1}{T}\sum_{t=1}^{T}\Lambda_k(t)-\frac{\xi_k^{\Delta^*(\delta)}(T)}{T}\right)\leq 0\right\}\\
&\leq 1-\mathbf{P}\left\{\left\{\frac{1}{T}\sum_{t=1}^{T}\Lambda_k(t)\leq\lambda_k+\frac{\delta}{2}\right\}\bigcap\left\{\frac{\xi_k^{\Delta^*(\delta)}(T)}{T}\geq\lambda_k+\frac{\delta}{2}\right\}\right\}\\
&\leq \mathbf{P}\left\{\frac{1}{T}\sum_{t=1}^{T}\Lambda_k(t)>\lambda_k+\frac{\delta}{2}\right\}+\mathbf{P}\left\{\frac{\xi_k^{\Delta^*(\delta)}(T)}{T}<\lambda_k+\frac{\delta}{2}\right\}\\
&\qquad\text{(by De Morgan's law and union bound)}\\
&\leq \mathbf{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T}\Lambda_k(t)-\lambda_k\right|>\frac{\delta}{2}\right\}+\mathbf{P}\left\{\left|\frac{\xi_k^{\Delta^*(\delta)}(T)}{T}-\lambda_k-\delta\right|>\frac{\delta}{2}\right\}.
\end{aligned} \tag{11}$$

Now, from (1),

$$\sum_{T=1}^{\infty}\mathbf{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T}\Lambda_k(t)-\lambda_k\right|>\frac{\delta}{2}\right\}<\infty. \tag{12}$$

Furthermore, since the scheduling decisions are taken independently in every reward state, we obtain the following relation using (2) and Property 3.

$$\sum_{T=1}^{\infty} \mathbf{P}\left\{\left|\frac{\xi_k^{\Delta^*(\delta)}(T)}{T} - \lambda_k - \delta\right| > \frac{\delta}{2}\right\} < \infty. \tag{13}$$

Now, $\mathbf{E}[\mathcal{BP}_k] = \sum_{T=1}^{\infty} \mathbf{P}\{\mathcal{BP}_k \geq T\}$. Thus, it follows from (11), (12) and (13) that the expected busy period length for session k is finite. Since k is arbitrary, the result follows.

Now, we show that if any policy can stabilize the system, then there exists $\delta > 0$ such that $\Delta^*(\delta)$ is also stable. First, we note that if $\sum_{k=1}^{n} \lambda_k \geq 1$, then no policy can stabilize the system. Now, for any $\vec{\lambda}$ such that $\sum_{k=1}^{n} \lambda_k < 1$, $\Delta^*(\delta)$ is stable for every $\delta \in \left(0, \frac{1-\sum_{k=1}^{n}\lambda_k}{n+1}\right)$. ■

PROOF OF THEOREM 2

First, we prove a supporting lemma (Lemma 1) that shows that $\Phi(0)$ is an upper bound on the utility of any stable policy. Recall that $\Phi(0)$ is the value of the objective function of **LP**(δ) with parameter $\delta = 0$. Using Lemma 1, we prove Theorem 2.

Lemma 1: For any stable policy Δ, $U^{\Delta} \leq \Phi(0)$.

Proof: Here, we present the proof for a simple case. Specifically, we assume that $\lim_{t\to\infty} \frac{\xi_{ik}^{\Delta}}{t}$ exists with probability 1. We note that the result holds even when this condition is not satisfied. Consider any non-trivial sample path ω, and let

$$\gamma_{ik}^{\Delta}(\omega) = \lim_{t\to\infty} \frac{\xi_{ik}^{\Delta}(\mathcal{B}, t; \omega)}{t}.$$

Note that $\gamma_{ik}^{\Delta}(\omega) \geq 0$. The quantity $\gamma_{ik}^{\Delta}(\omega)$ has an interpretation of the fraction of customers from class k that received reward r_{ik}. Moreover, since Δ is stable, the rate of departure (due to service completion) is equal to the arrival rate λ_k w.p. 1 for every class k. Thus,

$$\sum_{i=1}^{m} \gamma_{ik}^{\Delta}(\omega) = \lambda_k.$$

Furthermore, the fraction of slots in which $\mathbf{R}(t) = \mathbf{R}_i$ is b_i w.p. 1. Hence,

$$\sum_{k=1}^{n} \gamma_{ik}^{\Delta}(\omega) \leq b_i.$$

Now, the utility of Δ along ω is given as follows.

$$U^{\Delta}(\omega) = \sum_{i=1}^{m}\sum_{k=1}^{n} \gamma_{ik}^{\Delta}(\omega) r_{ik}.$$

Now, note that $\gamma_{ik}^{\Delta}(\omega)$'s satisfy all the constraints of **LP**(0). Thus, $\gamma_{ik}^{\Delta}(\omega)$'s are the feasible solution of **LP**(0). Moreover, utility of Δ has the same form as the objective function of **LP**(0). Thus, $\Phi(0) \geq U^{\Delta}(\omega)$. Since ω is any non-trivial sample path, the result follows. ■

Now, we prove ϵ-optimality of $\Delta^*(\delta)$.

A. Proof of Theorem 2

Proof: We first prove (4). Fix $0 < \delta < (1 - \sum_{k=1}^{n} \lambda_k)/n$. Note that **LP**$(\delta)$ is feasible and $\Delta^*(\delta)$ is stable. Now, the total reward received by k^{th} class under $\Delta^*(\delta)$ until t is $\sum_{i=1}^{m} r_{ik}\xi_{ik}^{\Delta^*(\delta)}(\mathcal{B}, t)$. Thus, the utility of class k under $\Delta^*(\delta)$ is

$$\begin{aligned}
U_k^{\Delta^*(\delta)} &= \lim_{t\to\infty} \frac{\sum_{i=1}^{m} r_{ik}\xi_{ik}^{\Delta^*(\delta)}(\mathcal{B}, t)}{t} \\
&= \lim_{t\to\infty} \frac{\sum_{i=1}^{m} r_{ik}\left(\xi_{ik}^{\Delta^*(\delta)}(t) - \xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t)\right)}{t} \\
&= \sum_{i=1}^{m} r_{ik}\left[\lim_{t\to\infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(t)}{t} - \lim_{t\to\infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t)}{t}\right] \\
&= \sum_{i=1}^{m} r_{ik}w_{ik}^*(\delta) - \lim_{t\to\infty} \sum_{i=1}^{m} r_{ik}\frac{\xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t)}{t} \quad \text{w.p. 1 (by Property 2)} \\
&\geq \sum_{i=1}^{m} r_{ik}w_{ik}^*(\delta) - r_k^{\max} \sum_{i=1}^{m} \lim_{t\to\infty} \frac{\xi_{ik}^{\Delta^*(\delta)}(\mathcal{I}, t)}{t} \\
&= \sum_{i=1}^{m} r_{ik}w_{ik}^*(\delta) - r_k^{\max}\delta \quad \text{w.p. 1 (by Property 3).}
\end{aligned}$$

Thus, (4) follows. Now, a lower bound on $U^{\Delta^*(\delta)}$ can be obtained as follows.

$$\begin{aligned}
U^{\Delta^*(\delta)} &= \sum_{k=1}^{n} U_k^{\Delta^*(\delta)} \\
&\geq \sum_{k=1}^{n} \left[\sum_{i=1}^{m} r_{ik}w_{ik}^*(\delta) - r_k^{\max}\delta\right] \quad \text{w.p. 1 (by (4))} \\
&\geq \sum_{k=1}^{n} \sum_{i=1}^{m} r_{ik}w_{ik}^*(\delta) - nr^{\max}\delta \\
&= \Phi(\delta) - nr^{\max}\delta.
\end{aligned} \tag{14}$$

Since, $r_{ik} \geq 0$ for every i and k, we note that $\Phi(\delta)$ is clearly a monotonically increasing function of δ. Thus, $\Phi(\delta) - \Phi(0) \geq 0$. Now, consider

$$\begin{aligned}
\Phi(0) - U^{\Delta^*(\delta)} &\leq \Phi(0) - \Phi(\delta) + nr^{\max}\delta \quad \text{w.p. 1 (by (14))} \\
\Rightarrow \quad U^{\Delta} - U^{\Delta^*(\delta)} &\leq nr^{\max}\delta \quad \text{w.p. 1 (by Lemma 1).}
\end{aligned}$$

Now, by the choice of δ in Theorem 2, $nr^{\max}\delta \leq \epsilon$. Since Δ is an arbitrary stable policy ϵ-optimality of $\Delta^*(\delta)$ follows. ■

PROOF OF THEOREM 3

We first introduce some notation that we use throughout this section. Let $\vec{\ell} = (\ell_1, \ldots, \ell_n)$ denote n-dimensional binary vector. Let $\mathcal{L}$ be the set of vectors $\vec{\ell}$ such that at most one of its element is 1. We refer to $\vec{\ell} \in \mathcal{L}$ as the *valid schedule* and $\mathcal{L}$ is the set of valid schedules. Note that $|\mathcal{L}| = n + 1$. We index the allowable schedules as $\vec{\ell}_0$ to $\vec{\ell}_n$. A valid schedule $\vec{\ell}_i = (\ell_{i1}, \ldots, \ell_{in})$ has $\ell_{ii} = 1$ for $i > 0$ and $\vec{\ell}_0 = \vec{0}$, where $\vec{0}$ is the n-dimensional vector of all zeroes. Let $\vec{\ell}^{\Delta}(t)$ denote a schedule chosen by policy Δ in slot t. If $\vec{\ell}^{\Delta}(t) = \vec{\ell}_i$ for $i > 0$, then i^{th} class is scheduled, and if $\vec{\ell}^{\Delta}(t) = \vec{\ell}_0$, then the none of the classes is scheduled. With this notation, clearly,

$$\vec{\ell}^{\Delta_O}(t) \in \arg\max_{\vec{\ell} \in \mathcal{L}} \left\{ \sum_{k=1}^{n} (Q_k(t)\ell_k - V(r_k^{\max} - r_k(t))\ell_k) \right\}. \tag{15}$$

For brevity, let

$$W(\vec{Q}, \vec{\ell}, \mathbf{R}(t)) \stackrel{\text{def}}{=} \sum_{k=1}^{n} (Q_k(t)\ell_k - V(r_k^{\max} - r_k(t))\ell_k). \tag{16}$$

Let $Y(t) \stackrel{\text{def}}{=} (\vec{Q}(t), \mathbf{R}(t))$. Note that $\{Y(t)\}_{t \geq 1}$ is a Markov chain. We define a Lyapunov function as follows.

$$f(Y(t)) = \sum_{k=1}^{m} (Q_k(t))^2. \tag{17}$$

Now, the dynamic queue length equation for the k^{th} class under policy Δ is as follows.

$$Q_k(t+1) = \max\left\{0, Q_k(t) + \Lambda_k(t) - \ell_k^{\Delta}(t)\right\}. \tag{18}$$

From (17) and (18), it follows that

$$\begin{aligned}
f(Y(t+1)) &\leq \sum_{k=1}^{n} \left[(Q_k(t))^2 + (\Lambda_k(t))^2 + (\ell_k^{\Delta}(t))^2 + 2Q_k(t)\Lambda_k(t) - 2Q_k(t)\ell_k^{\Delta}(t) - 2\Lambda_k(t)\ell_k^{\Delta}(t)\right] \\
&\leq \sum_{k=1}^{n} \left[(Q_k(t))^2 + (\Lambda_k(t))^2 + (\ell_k^{\Delta}(t))^2 + 2Q_k(t)\Lambda_k(t) - 2Q_k^{\Delta}(t)\ell_k^{\Delta}(t)\right] \\
&\leq n(A_{\max}^2 + 1) + \sum_{k=1}^{n} \left[(Q_k(t))^2 + 2Q_k(t)\Lambda_k(t) - 2Q_k(t)\ell_k^{\Delta}(t)\right].
\end{aligned}$$

Let, $M \stackrel{\text{def}}{=} n(A_{\max}^2 + 1)$. Thus, it follows that

$$f(Y(t+1)) - f(Y(t)) \leq M + \sum_{k=1}^{n} \left[2Q_k(t)\Lambda_k(t) - 2Q_k(t)\ell_k^{\Delta}(t)\right].$$

Now, we consider Lyapunov drift.

$$
\begin{aligned}
&\mathbb{E}\left[f(Y(t+1)) - f(Y(t)) \mid Y(t)\right] \\
&\leq \; M + \mathbb{E}\left[\sum_{k=1}^{n}\left[2Q_k(t)\Lambda_k(t) - 2Q_k(t)\ell_k^{\Delta}(t)\right] \mid Y(t)\right] \\
&= \; M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - \mathbb{E}\left[\sum_{k=1}^{n} 2Q_k(t)\ell_k^{\Delta}(t) \mid Y(t)\right] \qquad (19)\\
&= \; M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - \mathbb{E}\left[\sum_{k=1}^{n} 2Q_k(t)\ell_k^{\Delta}(t) \mid Y(t)\right] \\
&\quad +2V\mathbb{E}\left[\sum_{k=1}^{n}[r_k^{\max} - r_k(t)]\ell_k^{\Delta}(t) \mid Y(t)\right] - 2V\mathbb{E}\left[\sum_{k=1}^{n}[r_k^{\max} - r_k(t)]\ell_k^{\Delta}(t) \mid Y(t)\right] \\
&= \; M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - 2\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta}(t), \mathbf{R}(t))|Y(t)\right] - 2V\mathbb{E}\left[\sum_{k=1}^{n}[r_k^{\max} - r_k(t)]\ell_k^{\Delta}(t)|Y(t)\right] \qquad (20)
\end{aligned}
$$

Note that (20) obtains upper bound on the drift under any scheduling policy Δ. Now, consider Lyapunov drift under policy Δ_O. From (20),

$$
\begin{aligned}
&\mathbb{E}\left[f(Y(t+1)) - f(Y(t)) \mid Y(t)\right] \\
&\leq \; M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - 2\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta_O}(t), \mathbf{R}(t))|Y(t)\right] - 2V\mathbb{E}\left[\sum_{k=1}^{n}[r_k^{\max} - r_k(t)]\ell_k^{\Delta_O}(t)|Y(t)\right] \qquad (21)
\end{aligned}
$$

By definition of Δ_O (equation (15)), for every Δ and t,

$$
\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta_O}(t), \mathbf{R}(t)) \mid Y(t)\right] \geq \mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta}(t), \mathbf{R}(t)) \mid Y(t)\right]. \qquad (22)
$$

Now, let Δ_1 denote a randomized policy that schedules class k w.p. $\lambda_k + \delta$ in every slot irrespective of $Y(t)$ for any $\delta \in (0, (1 - \sum_{k=1}^{n} \lambda_k)/n)$. Clearly, Δ_1 is stable for every $\vec{\lambda} \in \Omega$. Thus, from (21), for every $\vec{\lambda} \in \Omega$,

$$
\begin{aligned}
&\mathbb{E}\left[f(Y(t+1)) - f(Y(t)) \mid Y(t)\right] \\
&\leq \; M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - 2\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta_1}(t), \mathbf{R}(t)) \mid Y(t)\right] \\
&\quad -2V\mathbb{E}\left[\sum_{k=1}^{n}[r_k^{\max} - r_k(t)]\ell_k^{\Delta_O}(t) \mid Y(t)\right] \\
&\leq \; M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - 2\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta_1}(t), \mathbf{R}(t)) \mid Y(t)\right]. \qquad (23)
\end{aligned}
$$

Now, let us evaluate $\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta_1}(t), \mathbf{R}(t)) \mid Y(t)\right]$. First, consider

$$
\mathbb{E}[\ell_k^{\Delta_1}(t) \mid Y(t)] = \mathbb{E}[\ell_k^{\Delta_1}(t)] = \lambda_k + \delta. \qquad (24)
$$

Now, note that

$$\begin{aligned}
\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta_1}(t), \mathbf{R}(t)) \mid Y(t)\right] &\geq \mathbb{E}\left[\sum_{k=1}^{n} Q_k(t) l_k^{\Delta_1}(t) \mid Y(t)\right] \quad \text{(since } V > 0\text{)} \\
&= \sum_{k=1}^{n} Q_k(t) \mathbb{E}\left[l_k^{\Delta_1}(t) \mid Y(t)\right] \\
&= \sum_{k=1}^{n} Q_k(t)(\lambda_k + \delta) \quad \text{(from (24)).} \qquad (25)
\end{aligned}$$

By substituting (25) in (23) we obtain

$$\mathbb{E}\left[f(Y(t+1)) - f(Y(t)) \mid Y(t)\right] \leq M - \delta \sum_{k=1}^{n} Q_k(t).$$

Thus, for sufficiently large $\sum_{k=1}^{n} Q_k(t)$ the Lyapunov drift is negative and bounded away from 0. Thus, by Foster's theorem process $\{Y(t)\}_{t \geq 1}$ is positive recurrent. This implies that the policy $\Delta_O(V)$ is stable for every $V > 0$.

Now, we show that for sufficiently large V the policy $\Delta_O(V)$ is ϵ-optimal. By (21) and (22), it follows that for every $\vec{\lambda} \in \Omega$ and δ such that $\Delta^*(\delta)$ is stable

$$\begin{aligned}
&\mathbb{E}\left[f(Y(t+1)) - f(Y(t)) \mid Y(t)\right] \\
&\leq M + \sum_{k=1}^{n} 2Q_k(t)\lambda_k - 2\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta^*(\delta)}(t), \mathbf{R}(t)) \mid Y(t)\right] \\
&\quad -2V\mathbb{E}\left[\sum_{k=1}^{n} [r_k^{\max} - r_k(t)] \ell_k^{\Delta_O}(t) \mid Y(t)\right]. \qquad (26)
\end{aligned}$$

Taking expectation with respect to the stationary distribution of $Y(t)$ on both sides of (26), we obtain

$$0 \leq M + \sum_{k=1}^{n} 2\mathbb{E}[Q_k(t)]\lambda_k - 2\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta^*(\delta)}(t), \mathbf{R}(t))\right] - 2V\mathbb{E}\left[\sum_{k=1}^{n} [r_k^{\max} - r_k(t)] \ell_k^{\Delta_O}(t)\right]. \quad (27)$$

Now, let us evaluate $\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta^*(\delta)}(t), \mathbf{R}(t))\right]$. First, consider

$$\begin{aligned}
\mathbb{E}[\ell_k^{\Delta^*(\delta)}(t)] &= \sum_{i=1}^{m} \mathbb{E}[\ell_k^{\Delta^*(\delta)}(t) \mid \mathbf{R}(t) = \mathbf{R}_i] \mathbb{P}\left\{\mathbf{R}(t) = \mathbf{R}_i\right\} \\
&= \sum_{i=1}^{m} w_{ik}^*(\delta) = \lambda_k + \delta. \qquad (28)
\end{aligned}$$

Next, consider

$$\begin{aligned}
\mathbb{E}[r_k(t)\ell_k^{\Delta^*(\delta)}(t)] &= \sum_{i=1}^{m} \mathbb{E}[r_k(t)\ell_k^{\Delta^*(\delta)}(t) \mid \mathbf{R}(t) = \mathbf{R}_i] \mathbb{P}\left\{\mathbf{R}(t) = \mathbf{R}_i\right\} \\
&= \sum_{i=1}^{m} r_{ik} w_{ik}^*(\delta). \qquad (29)
\end{aligned}$$

Thus, from (28) and (29)

$$\mathbb{E}\left[W(\vec{Q}(t), \vec{\ell}^{\Delta^*(\delta)}(t), \mathbf{R}(t))\right] = \sum_{k=1}^{n} [Q_k(t)(\lambda_k + \delta) - V r_k^{\max}(\lambda_k + \delta)] + V\Phi(\delta). \tag{30}$$

Now, substituting (30) in (27), we obtain

$$0 \leq M + 2V \sum_{k=1}^{n} r_k^{\max}(\lambda_k + \delta) - 2V\Phi(\delta) - 2V\mathbb{E}\left[\sum_{k=1}^{n} [r_k^{\max} - r_k(t)]\ell_k^{\Delta_O}(t)\right]. \tag{31}$$

Since $Y(t)$ is positive recurrent and $\Delta_O(V)$ takes transmission decisions based only on the current state of $Y(t)$, we conclude the following by stationarity and ergodicity of $Y(t)$.

$$\begin{aligned} \mathbb{E}[\ell_k^{\Delta_O}(t)] &= \lim_{u\to\infty} \frac{1}{u} \sum_{v=1}^{u} \ell_k^{\Delta_O}(v) = \lambda_k \quad \text{w.p. } 1, \\ \mathbb{E}[r_k(t)\ell_k^{\Delta_O}(t)] &= \lim_{u\to\infty} \frac{1}{u} \sum_{v=1}^{u} r_k(v)\ell_k^{\Delta_O}(v) = U^{\Delta_O} \quad \text{w.p. } 1. \end{aligned}$$

Thus, from (31), it follows that

$$U^{\Delta_O} \geq \Phi(\delta) - \delta \sum_{k=1}^{n} r_k^{\max} - \frac{M}{2V}.$$

Now, we can choose δ and V such that $\delta \sum_{k=1}^{n} r_k^{\max} + \frac{M}{2V} < \epsilon$. Moreover, since $\Phi(\delta) \geq \Phi(0)$ and $\Phi(0)$ is an upper bound on the utility of any stable policy (Lemma 1), the result follows. ■

PROOF OF THEOREM 4

Proof: We note that $\mathcal{C}_{\vec{F}}$ is a subset of Ω. Thus, for every $\vec{\lambda} \in \mathcal{C}_{\vec{F}}$ there exists $\delta' > 0$ such that for every $0 < \delta < \delta'$ policy $\Delta^*(\delta)$ is stable (Theorem 1).

We present the proof for ϵ-fairness of Δ in a special case. Specifically, we assume that $\lim_{t\to\infty} \frac{\xi_{ik}^{\Delta}}{t}$ exists w.p. 1 for $\Delta \in \mathcal{C}_{\vec{F}}$. We note that the result holds even when this condition is not satisfied. Consider any non-trivial sample path ω, and let

$$\gamma_{ik}^{\Delta}(\omega) \stackrel{\text{def}}{=} \lim_{t\to\infty} \frac{\xi_{ik}^{\Delta}(\mathcal{B}, t; \omega)}{t}.$$

Furthermore, the fraction of slots in which $\mathbf{R}(t) = \mathbf{R}_i$ is b_i w.p. 1. Hence,

$$\sum_{k=1}^{n} \gamma_{ik}^{\Delta}(\omega) \leq b_i.$$

Moreover, since Δ is stable, the rate of departure (due to service completion) is equal to the arrival rate λ_k w.p. 1 for every class k. Thus,

$$\sum_{i=1}^{m} \gamma_{ik}^{\Delta}(\omega) = \lambda_k.$$

Now, the utility of k^{th} class under Δ along ω is given as follows.

$$U_k^{\Delta}(\omega) = \sum_{i=1}^{m} \gamma_{ik}^{\Delta}(\omega) r_{ik}.$$

Since $\Delta \in \mathcal{C}_{\vec{F}}$, $U_k^{\Delta}(\omega) \geq F_k$.

Moreover, the utility of Δ along ω is given as follows.

$$U^{\Delta}(\omega) = \sum_{i=1}^{m} \sum_{k=1}^{n} \gamma_{ik}^{\Delta}(\omega) r_{ik}.$$

Now, note that $\gamma_{ik}^{\Delta}(\omega)$'s satisfy all the constraints of $\mathbf{LP_1}(0)$. Thus, $\gamma_{ik}^{\Delta}(\omega)$'s are the feasible solution of $\mathbf{LP_1}(0)$. Moreover, utility of Δ has the same form as the objective function of $\mathbf{LP_1}(0)$. Thus, $\Phi_1(0) \geq U^{\Delta}(\omega)$, where $\Phi_1(\delta)$ denotes the optimal value of the objective function of $\mathbf{LP_1}(0)$. Since ω is any non-trivial sample path, we conclude that $\Phi_1(0)$ is the upper bound on the utility of every policy in $\mathcal{C}_{\vec{F}}$ w.p. 1. Moreover, since r_{ik}'s are non-negative $\Phi_1(\delta)$ is a monotone non-decreasing function of δ. Thus, $\Phi_1(\delta) \geq \Phi_1(0)$, for every $\delta \geq 0$ such that $\mathbf{LP_1}(\delta)$ is feasible.

Let $\widehat{w}_{ik}^{*}(\delta)$ denote the optimal values of variables w_{ik} in $\mathbf{LP_1}(\delta)$. Then as shown in Theorem 2, here also we have the following lower bound on the utility k^{th} class under $\Delta_1^{*}(\delta)$.

$$\begin{aligned} U_k^{\Delta_1^{*}(\delta)} &\geq \sum_{i=1}^{m} \widehat{w}_{ik}^{*}(\delta) r_{ik} - \delta r_k^{\max} \quad \text{w.p. 1} && (32) \\ &\geq F_k - \delta r_k^{\max} \quad \text{w.p. 1 (because of the fairness constraints in } \mathbf{LP_1}(\delta)). && (33) \end{aligned}$$

Thus, we have the following lower bound on the utility of $\Delta_1^{*}(\delta)$.

$$\begin{aligned} U^{\Delta_1^{*}(\delta)} &\geq \sum_{k=1}^{n} \sum_{i=1}^{m} \widehat{w}_{ik}^{*}(\delta) r_{ik} - n\delta r^{\max} \quad \text{w.p. 1} \quad \text{(from (32))} \\ &\geq \Phi_1(\delta) - n\delta r^{\max} \quad \text{w.p. 1.} && (34) \end{aligned}$$

Now, note that for $\delta \in \left(0, \min\left\{\frac{1-\sum_{k=1}^{n}\lambda_k}{n}, \frac{\epsilon}{nr^{\max}}\right\}\right)$, $\Delta_1^{*}(\delta)$ is ϵ-fair. ■

REFERENCES

[1] P. Chaporkar, A. Bhat, and S. Sarkar, "An adaptive transmission strategy to maximize throughput in mac layer wireless multicast," in *5th ACM Intl. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Tokyo, Japan, May 2004, pp. 256–267.

[2] P. Chaporkar and S. Sarkar, "Stochastic control techniques for throughput optimal wireless multicast," in *42nd IEEE Conf. on Decision and Control (CDC)*, Maui, Hawaii, Dec 2003.

[3] ——, "On-line optimal wireless multicast," in *2nd Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Nwtwoeks (WiOpt)*, Univ. of Cambridge, UK, Mar 2004, pp. 282–291.

[4] ——, "Wireless multicast: Theory and approaches," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1954–1972, Jun 2005.

[5] Q. Zhang and S. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 47, pp. 1688–1692, Nov 1999.

[6] P. Chaporkar and S. Sarkar, "Utility optimal scheduling for general reward states and stability constraint," University of Pennsylvania, http://www.seas.upenn.edu/∼prasanna, Tech. Rep., 2005.

[7] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, Feb 2001.

[8] S. Borst, "User-level performance of channel aware scheduling algorithms in wireless data networks," in *IEEE INFOCOM*, San Francisco, CA, Apr 2003.

[9] M. Goyal, A. Kumar, and V. Sharma, "Power constrined and delay optimal policies for scheduling transmission over a fading channel," in *IEEE INFOCOM*, San Francisco, CA, Apr 2003.

[10] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *American Mathematical Society Translations, Series 2, A volume in memory of F. Karpelevich*, 2002.

[11] A. Stolyar and K. Ramanan, "Largest weighted delay first scheduling: Large deviations andoptimality," *Annals of Applied Probability*, vol. 11, no. 1, pp. 1–48, 2001.

[12] A. Stolyar, S. Shakkottai, and R. Srikant, "Pathwise optimality of the exponential scheduling rule for wireless channels," *Advances in Applied Probability*, 2004.

[13] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Info. Theory*, vol. 30, no. 2, pp. 466–478, Mar 1993.

[14] ——, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, Dec 1992.

[15] M. Neely and E. Modiano, "Capacity and delay tradeoffs for ad-hoc mobile networks," *IEEE Trans. on Information Theory*, vol. 51, no. 6, pp. 1917–1937, Jun 2005.

[16] S. Borst and P. Whiting, "Dynamic rate control algorithms for hdr throughput optimization," in *IEEE INFOCOM*, Anchorage, Alaska, Apr 2001.

[17] S. Kulkarni and C. Rosenberg, "Opportunistic scheduling policies for wireless systems with short term fairness constraints," in *IEEE Globecom*, San Francisco, CA, Dec 2003.

[18] ——, "Opprotunistic scheduling in wireless systems with multiple interfaces and multiple constraints," in *6th ACM Intl. Workshop on Modeling, Analysis and Simulations of Wireless and Mobile Systems (MSWim)*, San Diego, CA, Sep 2003.

[19] X. Liu, E. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2064, Dec 2001.

[20] ——, "A framework for opportunistic scheduling in wireless networks," *Computer Networks Journal (Elsevier)*, vol. 41, no. 4, pp. 451–474, 2003.

[21] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *IEEE INFOCOM*, San Francisco, CA, Apr 2003.

[22] M. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *IEEE INFOCOM*, Miami, FL, Mar 2005.

[23] M. Neely, "Energy optimal control for time varying wireless networks," in *IEEE INFOCOM'05*, Mar 2005.

[24] P. Chaporkar, "Wireless multicast: Theory, approaches and protocols," Ph.D. dissertation, Univ. of Pennsylvania, Philadelphia, PA, 2005, http://www.seas.upenn.edu/∼prasanna/Disseration.pdf.