Throughput Loss in Task Scheduling due to Server State Uncertainty

Carri W. Chan
Stanford University
350 Serra Mall
Stanford, CA 94305
cwchan@stanford.edu

Nick Bambos
Stanford University
350 Serra Mall
Stanford, CA 94305
bambos@stanford.edu

ABSTRACT

We consider a dynamic scheduling system where a single controller selects 'tasks' to service over U 'servers' of fluctuating quality/speed. The quality/speed of each server determines the likelihood of successful service should a task be assigned to that server. The goal is to maximize the total expected number of tasks successfully served over a fixed time horizon (aggregate throughput) given only one server can be used in each time slot. However, the state of the servers are not known to the scheduler with certainty; at best, only statistical distributions (estimates) of the realized server states are available. We consider how the uncertainty of server state information compromises the expected aggregate throughput compared to a 'clairvoyant' scheduler which has instantaneous, perfect information about the realized server states.

The issue of operating in uncertain environments arises in a number of scheduling applications of interest from wireless applications to computing networks to revenue management systems. The results presented in this paper provide a framework for gauging the loss due to uncertainty in such scheduling systems.

First, it is shown that opportunistic scheduling (on the server of current *expected* best state) is throughput optimal, under uncertain (unknown) server states. Then, the throughput of the 'clairvoyant' scheduler is found to be upper-bounded (in general) by U times the throughput under uncertain server states; this bound is tight. Third, for bimodal and uniform server qualities/speeds better bounds are obtained–down to a factor of 2. Of course, actual throughput loss due to server state uncertainty depends on the server state distributions which are available as partial information to the scheduler. Finally, via numerical experiments we evaluate the throughput loss in various operational scenarios for wireless packet scheduling applications.

1. INTRODUCTION

A number of interesting scheduling problems have been studied under the assumption of perfect server state information. Utilizing information of each server's state can significantly improve the performance of many scheduling schemes. For example, if the speed of one server is vastly faster than that of others, it may make sense to use the server with the best speed in order to achieve the highest instantaneous rate. We consider a general model for scheduling problems where an infinite backlog of 'tasks', which we will also referred to as 'items', are to be served by a set of U servers. Only one server can be used in each time slot and some (limited) information about the speed/quality of each server is available. The goal is to determine a policy which dynamically selects a server to employ in each time slot in order to maximize the total number of tasks completed.

Our main motivation is throughput maximization for wireless packet scheduling. Wireless scheduling is one such application which has been extensively studied under the assumption of perfect channel/link state information (e.g. [1, 2, 3, 4]). In wireless packet scheduling, 'items' correspond to packets and 'servers' correspond to the designated communication channels for each of the U users. Unfortunately, acquiring channel state information is a physical process that is susceptible to errors and uncertainty. Better estimations of channel quality may be attainable by expending more energy and/or time. Unfortunately, in many applications, it may be simply too costly (or even physically impossible) to establish perfect knowledge of channel state information. Therefore, the best information practically available to a wireless packet scheduler may be some estimated distribution on the potential channel states, as opposed to the exact realized channel state.

A similar application which falls into this framework is in Internet communications where network congestion makes successful packet transmission random. Again, items correspond to packets and servers correspond to communications links from a single router (where the control is executed) to multiple intermediary nodes. As congestion on each link varies, packet level routing and scheduling can improve the network utilization. Many algorithms assume that congestion information is embedded in packets so that rate-control can be determined based on this knowledge [5]. A significant amount of effort has been expended to estimate Internet traffic in order to utilize this information for packet scheduling [6]. This begs the question, how much is lost due to unknown congestion?

Another application of interest is revenue management for product-line design. In this scenario, a manager must select which product to place on the market given customers' varying demand and willingness to pay. In this case, an item corresponds to a sale of a product and each server corresponds to placing a specific product on the market. An item is successfully 'serviced' if the product placed on the market is purchased by a customer–a sale is completed and revenue is accrued. The likelihood of a product being purchased depends on multiple factors, such as seasonality, price, available substitutes, etc. The goal of the manager is to place products in order to generate the most sales–which is equivalent to 'serving' the

VALUETOOLS 2009, October 20-22, Pisa, Italy

DOI 10.4108/ICST.VALUETOOLS2009.7547

most 'items'. For a more in depth discussion of models for product-line design, see [7, 8]. While much effort, especially by marketing firms, is expended to determine customer demand and willingness to pay, the estimates are often noisy as customers themselves have difficulty assessing their preferences.

In all of these applications, finding accurate information about server quality/speed can be very expensive and sometimes even impossible. The lack of perfect knowledge regarding realized server states can compromise the efficiency of the scheduler and often results in lower throughput when compared to a fictitious 'clairvoyant' scheduler which instantaneously has perfect knowledge of realized server states.

In this paper, we study the impact of uncertain server state information on throughput. We exemplify the issue in the following general scheduling model. A backlog of tasks are to be served by one of U servers of randomly fluctuating quality/speed. In each time slot, the scheduler selects a server to employ, so as to maximize the expected total number of items successfully served over a fixed time interval T (i.e. maximize expected aggregate throughput). The realized server states, however, are *not known* to the scheduler with certainty; only statistical distributions (estimates) of the true/realized server states are available to it. This limits the maximal throughput J^* achieved by this scheduler, compared to the maximal throughput J^o achieved by the 'clairvoyant' one, which instantaneously has perfect knowledge of realized server states. *At worst, how much could the throughput loss be?* We examine this and other related issues below.

In wireless applications, a substantial body of research has investigated the effect of noisy channel estimation on the capacity of wireless communications (e.g. see [9, 10] and references therein). These works study the problem of imperfect channel state information within an information-theoretic framework. In contrast, we quantify the loss in throughput due to the uncertain server state information using a decision-theoretic (dynamic-programming-like) framework, while much of previous work uses an information-theoretic context. Indeed, in the context of wireless scheduling, rather than viewing a wireless channel as having a time-varying bit-rate, we consider lossy packet communication with packet transmission success probability depending on the realized channel quality state. This captures diverse networking scenarios, where data is partitioned into packets of (nearly) equal size and the *key issue is packet scheduling* on communication channels. Furthermore, the scheduling model we examine here encompasses scheduling scenarios beyond the wireless setting to which these information-theoretic results are not directly transferable.

In this paper we focus on understanding throughput loss due to server state uncertainty. This could naturally later lead to designing protocols to allocate limited 'probing' resources to improve acquired/estimated server state information as considered in [11]. We do not investigate the latter here though. In [11], the authors consider how to balance channel probing to enable better channel estimates with packet transmissions. Our work differs from this previous work as we quantify the effects of poor estimates rather than consider how to improve them.

A similar (dual-like) problem to ours is the case where the task sizes are unknown, but the server state is known. In these scenarios, scheduling decisions must be made without full knowledge of the job sizes. A number of adversarial approximation algorithms have been proposed for different scheduling objectives in this context. See [12] for an overview of competitive analysis for online scheduling of varying job processing times.

The rest of this paper is structured as follows. In Section 2, we formally define the scheduling under uncertainty scenario, the expected aggregate throughput, etc. In Section 3, we find the maximum expected aggregate throughput J^* under any server state uncertainty, and the corresponding throughput J^o of the fictitious 'clairvoyant' schedule with perfect server state knowledge, satisfy the *tight* bound $\frac{J^*}{J^o} \leq U$ (where U is the number of servers/channels/ customer classes). This provides a tight characterization of the throughput loss. Under a number of specific server state distributions, this bound is strengthened in subsequent sections. For example, for i.i.d uniform server qualities it is shown that $\frac{J^*}{J^o} \leq 2$. Simulation experiments for applications in wireless packet scheduling provided in Section 6 demonstrate the impact of channel uncertainty on throughput, which is consistent with the theoretical results. Finally, Section 7 presents some conclusions.

2. SCHEDULING UNDER UNCERTAINTY

We start by defining the general scheduling model. Time is slotted and indexed by $t \in \mathcal{T} = \{1, 2, ..., T\}$. There are U servers, indexed by $u \in \mathcal{U} = \{1, 2, ..., U\}$. Only one server can be used in each time-slot. If a server $u \in \mathcal{U}$ is utilized, it will 'deplete' (or remove) an item from the queue with some probability. Such tasks are never exhausted and are readily available at the controller's queue.

Let c_u^t be the state at time t of server u and let $\mathcal{C}_u$ be the set of all states this server can attain over its evolution. It is assumed that each c_u^t is a random variable that, given distribution $g_u^t = P[c_u^t \in \mathcal{A}]$, is statistically independent of the history $\{c_u^{t'}, t' < t\}$ of the same server u, as well as of the past and current states $\{c_{u'}^{t'}, t' \leq t\}$ of all other servers $u' \in \mathcal{U} - u$. Note that while c_u^t may be dependent over time, g_u^t is a sufficient statistic which allows one to ignore the past given g_u^t.

In each time slot t, the controller decides to assign a task to server $u \in \mathcal{U}$. If server u is used at time t when its state is $c = c_u^t$ then the item is successfully removed with probability

$$s_u^t(c) = P[\text{ successful removal by server } u \mid c_u^t = c\,]. \quad (1)$$

Alternatively, with probability $1 - s_u^t(c)$ the item is not successfully removed. In wireless and Internet communication systems this occurs when a packet is excessively corrupted (e.g. by interference) and cannot be successfully decoded and received, hence, it is dropped. In product-line design, the customers may decide not to purchase the product or to buy from a different vendor and, hence, the sale (item) is not completed and revenue is not received. In particular, let X_u^t be a 1/0 random variable which is 1 if server u successfully serves an item at time t and is 0 otherwise. Then, $P[X_u^t = 1 | c_u^t = c] = s_u^t(c)$ and $P[X_u^t = 0 | c_u^t = c] = 1 - s_u^t(c)$. It is assumed that, given g_u^t, for each $u \in \mathcal{U}$ and $t \in \mathcal{T}$ the random variable X_u^t depends only on c_u^t and is independent of all others. That is, given sufficient statistic g_u^t, successful/failed service events are statistically independent across time slots $t \in \mathcal{T}$ and servers $u \in \mathcal{U}$, except for the state of the server in the current time slot.

Given the probability $g_u^t(\mathcal{A}) = P[c_u^t \in \mathcal{A}]$ of the server state c_u^t being in the (measurable) subset $\mathcal{A}$ of the server state space $\mathcal{C}_u$, we can obtain the probability

$$F_u^t(x) = P\left[s_u^t \leq x\right] = P\left[c \in \mathcal{C}_u : s_u^t(c) \in [0, x]\right]; \quad (2)$$

that is, the statistical distribution of the service success probability by server u at time t. Equivalently, we can obtain the densities $f_u^t(x)$

$$F_u^t(x) = \int_0^x f_u^t(x')dx' \quad (3)$$

viewed in a generalized sense (with delta-spikes) if the distribution

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

has discontinuities.

The server state information available to the scheduler at time $t \in \mathcal{T}$ is the service success densities $\{f_u^t(x), x \in [0,1], u \in \mathcal{U}\}$. These can be viewed as (implicit) estimates of the server states, provided to the scheduler and reflecting uncertainty about the true/realized server states; the more 'spread-out' the distributions the higher the uncertainty. The mapping of g_u^t to f_u^t makes f_u^t a sufficient statistic of the server success probabilities.

A scheduling policy π chooses at each time slot $t \in \mathcal{T}$ a server $u^t = \pi(f^t, t) \in \mathcal{U}$ to employ, given the available server state information

$$f^t = \{f_u^t(x), x \in [0,1], u \in U\}, \tag{4}$$

that is, the current densities (estimates) of the service success probabilities of each server. Let Π be the set of all possible scheduling policies, utilizing information f^t (equivalently F^t) to select the server to use at time $t \in \mathcal{T}$.

The scheduler's objective is to maximize the expected aggregate throughput of the system, that is, the expected total number of tasks that are successfully served over the time horizon T. Define a reward function $R(f^t, \pi(f^t, t)) = X_u^t$ which is 1 if an item is successfully removed by server $u^t = \pi(f^t, t)$ chosen by policy π at time t, or is 0 otherwise. The expected aggregate throughput of the system operating under scheduling policy π over the interval $\{t, t+1, ..., T\}$ is

$$J^\pi(f^t, t) = E\left[\sum_{t'=t}^{T} R(f^{t'}, \pi(f^{t'}, t'))\right], \tag{5}$$

starting at t with server state information f^t. Let

$$J^*(f^t, t) = \max_{\pi \in \Pi} J^\pi(f^t, t) \tag{6}$$

and define $\pi^* \in \Pi$ to be a scheduling policy which achieves this maximum.

As shown below, an optimal schedule is the *greedy (opportunistic)* policy $\pi^g \in \Pi$, which schedules the server whose the current expectation of a successful service is maximized. Specifically,

$$\begin{aligned} \pi^g(f^t, t) &= \arg\max_{u \in \mathcal{U}} E[R(f^t, u)] \\ &= \arg\max_{u \in \mathcal{U}} E_f[s_u^t]. \end{aligned} \tag{7}$$

Note that the expectation of successful service by server u at t (hence, reward 1) is $E[R(f^t, u)] = E[X_u^t] = E[E[X_u^t | c_u^t]] = E[s_u^t(c_u^t)] = \int_0^1 s f_u^t(s) ds = E_f[s_u^t]$. In general, we denote below by $E_f[\cdot]$ expectations with respect to the densities $f^t = \{f_u^t(x), u \in \mathcal{U}, t \in \mathcal{T}\}$ (or equivalently the distributions $F^t = \{F_u^t(x), u \in \mathcal{U}, t \in \mathcal{T}\}$) of the server success probabilities, which is the only information revealed to the scheduler about the servers.

2.1 Opportunistic Scheduling on Perfectly Known Servers

There has been a substantial body of research on how to leverage *perfect* server state knowledge to develop throughput optimal schedules. The main premise in these algorithms is that the realizations of the server states c_u^t are perfectly *known* in each time slot t and can be used to determine which task to schedule. The set of admissible schedules is then expanded from Π to Π^o, including fictitious 'clairvoyant' schedules that instantaneously know the actual server state c_u^t realized. We call this set Π^o to denote its 'oracle' or 'omniscience' abilities. Note that unlike standard competitive-type analysis and oracle policies as in [13, 14], the schedules in Π^o are only aware of the realized c_u^t (hence, s_u^t), but *not* X_u^t, so that the actual service result (success/failure) is not known a priori.

Simple opportunistic policies utilizing perfect server state information are known to be throughput maximizing [1, 2, 3, 4]. In the context of our problem, we simply reiterate this result below and refer the reader to the previous literature for the details and proof.

THEOREM 1. *(Optimal Scheduling on Known Servers) The opportunistic scheduling policy*

$$\pi^o(c^t, t) = \arg\max_{u \in \mathcal{U}} s_u^t(c_u^t) \tag{8}$$

or equivalently

$$\pi^o(s^t, t) = \arg\max_{u \in \mathcal{U}} s_u^t \tag{9}$$

achieves maximal expected throughput $J^o(c^t, t) = J^o(s^t, t)$ across $\{t, t+1, ..., T\}$ within the class of schedules Π^o. The latter schedules perfectly know the server states $c^t = \{c_u^t, u \in \mathcal{U}\}$, hence, the service success probabilities $s^t = \{s_u^t, u \in \mathcal{U}\}$ (but not the service outcomes).

In this paper, we are primarily concerned with comparing

1. the maximum expected aggregate throughput $J^*(f^t, t)$ achieved in $\{t, t+1, ..., T\}$ by schedules in Π with partial (uncertain) knowledge of the server state provided by $f^t = \{f_u^t(x), x \in [0,1]\}$ to
2. the maximum expected aggregate throughput $J^o(t)$ achieved by schedules in Π^o with perfect knowledge of the server state $c^t = \{c_u^t, u \in \mathcal{U}\}$, hence, of the service success probabilities $s^t = \{s_u^t, u \in \mathcal{U}\}$.

Define a similar reward function $R^o(c^t, \pi^o(c^t, t)) = X_u^t$ when the 'clairvoyant' optimal opportunistic schedule $u^t = \pi^o(c^t, t)$ is used.

We note again that in the context of wireless packet scheduling, a line of information-theoretic research has examined the effect of channel estimation errors on capacity [9, 10, 15, 16]. Recall that channel estimation errors corresponds to server state estimation errors in our formulation. This prior research, however, focuses on channel capacity and coding rather than on packet scheduling, as we do in this paper. In [16], the authors propose a back-off mechanism to generate error free codebooks when the channel state information is noisy. In order to achieve the capacity limits established in these works, long transmission sessions may be necessary. Alternatively, one can view wireless channels as having time-varying probabilities of successful transmission rather than time-varying bitrates. It is this packetized view that our model encompasses. Moreover, as discussed in Section 1, this model includes many applications beyond wireless scheduling.

3. THROUGHPUT LOSS DUE TO UNCERTAINTY

In this section, we compare the maximum throughput $J^*(f_t, t)$ under server uncertainty to the throughput $J^o(t)$ of the 'clairvoyant' schedule π^o. The server uncertainty induces a throughput loss; however, we show that $J^o(t)/J^*(f_t, t) \leq U$ under any f^t and the bound is actually tight. We start by determining the maximum throughput schedule under server uncertainty.

THEOREM 2. *(Optimal Scheduling on Uncertain Servers) Given densities $f^t = \{f_u^t(x), x \in [0,1], u \in \mathcal{U}\}$ of the service success probabilities as available server state information, the* greedy (opportunistic) schedule

$$\pi^*(f^t, t) = \pi^g(f^t, t) = \arg\max_{u \in \mathcal{U}} E_f[s_u^t] \tag{10}$$

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

is optimal, *maximizing the expected throughput of the system, hence,* $J^*(f^t,t) = J^g(f^t,t)$.

PROOF. We argue by contradiction. Assume there exists some time $\tau \leq T$, such that the optimal policy and the greedy policy do not coincide. Define the set of greedy servers at time τ as $U^*_\tau = \{\arg\max_u E_f[s^\tau_u]\}$. So, by assumption $\pi^*(f^\tau,\tau) \notin U^*_\tau$. Let's consider a policy $\tilde{\pi}$ which coincides with π^* for all $t \neq \tau$; in time slot τ, $\tilde{\pi} \in U^*_\tau$ schedules a greedy server. Since there are an infinite number of items to be depleted and the server states are independent of the past given f^t_u, in time slots $t \neq \tau$ the expected reward earned by the $\tilde{\pi}$ policy, $R(f^t,\tilde{\pi}(f^t,t))$ is *identical* to the expected reward earned by the π^* policy, $R(f^t,\pi^*(f^t,t))$. By definition of the greedy set: $E_f[s^\tau_{\pi^*(f^\tau,\tau)}] < E_f[s^\tau_{\tilde{\pi}(f^\tau,\tau)}]$. Therefore,

$$\begin{aligned} J^*(f^t,t) &= E\big[\sum_{t'=t}^{T} R(f^{t'},\pi^*(f^{t'},t'))\big] \\ &= E\big[R(f^\tau,\pi^*(\tau)) + \sum_{t'\neq\tau} R(f^{t'},\tilde{\pi}(f^{t'},t'))\big] \\ &< E\big[R(f^\tau,\tilde{\pi}(\tau)) + \sum_{t'\neq\tau} R(f^{t'},\tilde{\pi}(f^{t'},t'))\big] \\ &= E\big[\sum_{t'=t}^{T} R(f^{t'},\tilde{\pi}(f^{t'},t'))\big] = J^{\tilde{\pi}}(f^t,t) \end{aligned}$$

This contradicts the optimality of J^*. Therefore, there exists an optimal policy which, in each time slot, schedules the server with the highest expected probability of successful service. ■

By Theorem 1 and 2, we see the optimal schedules are opportunistic both in case 1) of perfect server information given by c^t (or s^t) and in case 2) of partial server information given by the f^t densities of the service success probabilities. However, in case 1) where the server state is known items are scheduled on the server with the *best* state, but in case 2) to the one with the *best expected* state. Note that at the limit where the server state uncertainty is squeezed out (the densities become single delta-spikes) the schedule of case 2) actually degenerates to that of case 1). Throughput loss due to server state uncertainty is induced by scheduling based on expected rather than true state realizations.

3.1 Tight Upper-Bound on $J^o(t)/J^*(f_t,t)$

We begin with a bound regarding the maximization of the random service success probabilities s^t_u on the servers.

LEMMA 1. *For the success probabilities (1),*

$$E_f[\max_{u\in\mathcal{U}} s^t_u] \leq U \max_{u\in\mathcal{U}} E_f[s^t_u] \tag{11}$$

Recall that U is the number of servers.

Sketch of Proof: We prove here the result in the (special) case where the distributions $F^t_u(x)$ of success probabilities are smooth, hence, $f^t_u(x) = dF^t_u(x)/dx$ for all $u \in \mathcal{U}$ and $t \in \mathcal{T}$. For notational simplicity, we suppress the dependence on time. Let us define $s_y = \max_{u\in\mathcal{U}} s_u$, so that

$$F_y(x) = P(\max_u s_u \leq x) = \prod_{u=1}^{U} P(s_u \leq x) = \prod_{u=1}^{U} F_u(x).$$

Then $f_y(x) = \frac{d}{dx}F_y(x) = \sum_{u=1}^{U} f_u(x)\prod_{k\neq u} F_k(x)$. We can now define the expectation of s_y.

$$\begin{aligned} E_f[s_y] &= E_f[\max_{u\in\mathcal{U}} s_u] \\ &= \int_0^1 x f_y(x)dx \\ &= \sum_{u=1}^{U}\int_0^1 x f_u(x)\prod_{k\neq u} F_k(x)dx \\ &\leq \sum_{u=1}^{U}\int_0^1 x f_u(x)dx \\ &= \sum_{u=1}^{U} E_f[s_u] \\ &\leq U \max_u E_f[s_u] \end{aligned}$$

where the first inequality is because $F_u(x) \leq 1$ for all x and u. The second inequality comes from the definition of the maximization function. This proves our bound. ■

This Lemma allows derivation of a tight bound on the throughput loss due to server state information uncertainty.

THEOREM 3. *(Tight Bound on Throughput Loss) For any distribution of server state, we have*

$$\frac{J^o(t)}{J^*(f^t,t)} \leq U, \tag{12}$$

where U is the number of users. The bound is tight, as Example 1 below demonstrates.

Hence, the throughput gain of 'clairvoyant' schedule π^o over $\pi^* = \pi^g$ is upper-bounded by U. Correspondingly, this reflects a throughput loss of schedule π^* compared to π^o.

PROOF. This is a direct consequence of Lemma 1. In the case of unknown server state, the expected reward earned in each time slot is taken over the distribution for the server state. Recall by Theorem 2, under unknown server state, $E[R(f^{t'},\pi^*(f^{t'},t'))] = \max_u E_f[s^{t'}_u]$. In the case of known server state, the expected reward earned in each time slot is taken over the probability of successful service of each item. Recall by Theorem 1, under known server state, $E[R^o(c^{t'},\pi^o(c^{t'},t'))] = E_f[\max_u s^{t'}_u]$. Then,

$$\begin{aligned} UJ^*(f^t,t) &= UE\left[\sum_{t'=t}^{T} R(f^{t'},\pi^*(f^{t'},t'))\right] \\ &= \sum_{t'=t}^{T}\left[U\max_u E_f[s^{t'}_u]\right] \\ &\geq \sum_{t'=t}^{T} E_f[\max_u s^{t'}_u] \\ &= E\left[\sum_{t'=t}^{T} R^o(c^{t'},\pi^o(c^{t'},t'))\right] = J^o(t) \quad ■ \end{aligned}$$

The preceding bound is a worse case bound, but it is tight, as the following example shows.

EXAMPLE 1. *Let the servers be identically distributed (besides being independent). Assume the scheduling horizon is* 1*. The distribution for each server is such that with probability $\epsilon > 0$ it is in a good state, so that $s_u = 1$ and with probability $1-\epsilon$ it is in the bad state, so that $s_u = 0$. Because the servers are i.i.d. the policy over* unknown *server state picks one server at random, achieving*

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

reward $J^*(f^1,1) = E_f[s_u^1] = \epsilon$. *The policy over* known *server state picks any of the servers in the good state and picks a server randomly if they are all in the bad state. We can determine the expected reward by finding the expected value of the maximum* s_y. *The distribution of* s_y *is given by:*

$$P(s_y = x) = \begin{cases} (1-\epsilon)^U, & \text{if } x = 0; \\ 1-(1-\epsilon)^U, & \text{if } x = 1. \end{cases}$$

Therefore, $E_f[s_y] = J^o(1) = 1-(1-\epsilon)^U$ *and* $\frac{J^o(1)}{J^*(f^1,1)} = \frac{1-(1-\epsilon)^U}{\epsilon}$. *Letting* $\epsilon \to 0$ *and using L'Hopital's rule gives:*

$$\begin{aligned} \lim_{\epsilon\to 0} \frac{J^o(1)}{J^*(f^1,1)} &= \lim_{\epsilon\to 0} \frac{1-(1-\epsilon)^U}{\epsilon} \\ &= \lim_{\epsilon\to 0} U(1-\epsilon)^{U-1} \\ &= U \end{aligned}$$

Therefore, the bound in Theorem 3 is tight.

In this example, server are identically distributed. The loss of throughput grows linearly with the number of servers in the system. This is because the server-state-knowing schedule π^o can pick the single server with the *best* quality (i.e. fastest speed); however the policy with unknown state picks a server at random. As the number of servers grows, the probability that the π^* policy is able to pick the best server decreases. Hence the loss of throughput grows.

Note that this example is highly degenerate with the servers being i.i.d. and the probability of being in the good state going to 0. While the bound in Theorem 3 is tight, in practice, the loss is likely to be smaller than U.

3.2 The Case When Some Server States are Known

We note that the preceding results hold in the case where all U server states are unknown. However, it may be the case that some server states are known, while others are not. Suppose now that there are $U = U_{kwn} + U_{ukn}$ server, where U_{kwn} denotes the number of server with perfect, *known* state information and U_{ukn} denotes the number of servers with *uncertain* state information.

THEOREM 4. *(Throughput Loss) For* $U = U_{kwn}+U_{ukn}$ *servers, we have*

$$\frac{J^o(t)}{J^*(f^t,t)} \le (U_{ukn}+1) \tag{13}$$

The proof is similar to that of Theorem 3, hence, we omit it.

3.3 Evolution of Throughput Loss over Time

The preceding bounds are guaranteed for any distribution for server states. Let α be the throughput loss in a single time slot. Then under some server state distributions, the throughput loss may be bounded by α. For instance the throughput loss over T time slots is also α if all server states are independent and identically distributed across time. Note the servers do not have to be identically distributed to each other. However, if their distribution is statistically static over time, a throughput loss of α is incurred across multiple time slots if it is incurred in a single time slot.

If the server state distributions are not identical across time, the throughput loss may depend on time, α_t, such that $E_f[\max_u s_u^t(c_u^t)] \le \alpha_t \max_u E_f[s_u^t(c_u^t)]$. Then the throughput loss over T time slots is $\alpha_{t^*} = \max_t \alpha_t$:

$$\begin{aligned} J^o(t) &= E\left[\sum_{t'=t}^{T} R^o(c^{t'}, \pi^o(c^{t'}, t'))\right] \\ &= \sum_{t'=t}^{T} E_f[\max_u s_u^{t'}] \\ &\le \sum_{t'=t}^{T} \left[\alpha_{t'} \max_u E_f[s_u^{t'}]\right] \\ &\le \alpha_{t^*} E\left[\sum_{t'=t}^{T} R(f^{t'}, \pi^*(f^{t'}, t'))\right] \\ &= \alpha_{t^*} J^*(f^t, t) \end{aligned} \tag{14}$$

We have shown the worst case bound on the loss of throughput due to unknown server state. An asymptotically degenerate distribution is given as an example which achieves this bound. However, in many applications of interest it is undesirable and unlikely for a scheduling session to occur when the probability of successful service is given by some small $\epsilon > 0$. As will be shown in subsequent sections, for other server state distributions, this bound can be improved.

4. I.I.D. GOOD-BAD SERVER STATES

In Example 1, the server qualities were independent and identically distributed. We examine this scenario in more depth. We can assume that $E_f[s_u] > 0$, otherwise, no services will ever be successful and the throughput is always 0.

Because all servers have the same distribution, the optimal policy for unknown server state is to schedule any server at random. Let's define random variable $Z_u = \frac{s_u}{E_f[s_u]} \in [0,\infty)$, so that $E[Z_u] = 1$. Now, the loss of throughput over one time slot due to unknown server state information is given by

$$\frac{E_f[\max_{u\in\mathcal{U}} s_u]}{\max_{u\in\mathcal{U}} E_f[s_u]} = E_f[\max_{u\in\mathcal{U}} Z_u]$$

In wireless applications, a common channel (server) model is the ON-OFF channel where the channel can either be in the "ON" state and packet transmissions (item services) are successful with probability 1, or it can be in the "OFF" state and packet transmissions fail with probability 1. We consider a similar server model where a server can either be in a GOOD state or a BAD state.

EXAMPLE 2. *(i.i.d. GOOD-BAD Servers) Consider a GOOD-BAD server such that* $P(s_u = p_g) = \gamma$ *and* $P(s_u = p_b) = 1-\gamma$. *Renormalizing to define* Z_u, *we have:*

$$Z_u = \begin{cases} z_g = \frac{p_g}{(1-\gamma)p_b+\gamma p_g}, & \text{w.p. } \gamma; \\ z_b = \frac{p_b}{(1-\gamma)p_b+\gamma p_g}, & \text{w.p. } 1-\gamma. \end{cases}$$

Recall that $E[Z_u] = 1 = z_g\gamma + z_b(1-\gamma)$ *and so* $z_b < 1$ *and* $z_g > 1$. *Assume that* $T = 1$, *we have then that the loss of throughput due to unknown server state for the GOOD-BAD server is:*

$$\begin{aligned} E[\max_{u\in\mathcal{U}} Z_u] &= z_g(1-(1-\gamma)^U) + z_b(1-\gamma)^U \\ &= z_g - \frac{(1-\gamma)^U(1-z_b)}{\gamma} \\ &< z_g \end{aligned}$$

Because $z_b < 1$, *as* $U \to \infty$, *the loss in throughput for the GOOD-*

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

BAD server approaches

$$z_g = \frac{p_g}{(1-\gamma)p_b + \gamma p_g} < \frac{p_g}{p_b}.$$

This can be arbitrarily close to $\frac{p_g}{p_b}$ *as* $\gamma \to 0$.

We have just shown the loss of throughput due to unknown server state in the case of i.i.d. GOOD-BAD servers. It is interesting to note that for a fixed support for the server quality state, the GOOD-BAD server results in the worse loss in throughput. Before we show this result, we begin with a preliminary one.

LEMMA 2. *(GOOD-BAD Servers with Fixed Mean) Suppose the support for each server state is given by a closed interval defined by* $[a,b] \subseteq [0,1]$*, so that* $P(s_u \notin [a,b]) = 0$*. Let* $E_f[s_u] = \mu \in (a,b)$*. Then the maximum loss in throughput is given by a GOOD-BAD server with* $P(s_u = a) = \frac{b-\mu}{b-a} = \gamma$ *and* $P(s_u = b) = 1-\gamma$*, i.e.* $f(x) = \gamma\delta_a(x) + (1-\gamma)\delta_b(x)$ *(*$\delta_y(x)$ *is the impulse function that equals* 1 *when* $x = y$ *and* 0 *otherwise). The loss is:*

$$E_f[\max_{u\in\mathcal{U}} s_u] \le (\frac{b}{(1-\gamma)a+\gamma b} \wedge U)E_f[s_u] = (\frac{b}{\mu} \wedge U)E_f[s_u]$$

where $x \wedge y = \min\{x,y\}$.

PROOF. Let us consider the case where $\frac{b}{\mu} < U$, since by Theorem 3 $E_f[\max_{u\in\mathcal{U}} s_u] \le UE_f[s_u]$ for any distribution of s_u. Now from Example 2, we have seen the GOOD-BAD server achieves this loss as $U \to \infty$. What remains to be shown is that the GOOD-BAD server is the worst possible throughput loss for this support and mean.

Suppose there exists some distribution $\hat{f}(x) \neq f(x)$. We will show that the throughput loss under distribution $\hat{f}(x)$ can be made worse. Let $g(x) = f(\frac{x}{\mu})$ be the p.d.f. for Z_u under the GOOD-BAD distribution given by $f(x)$. Similarly, let $\hat{g}(x) = g(\frac{x}{\mu}) \neq g(x)$ be the p.d.f. for Z_u under the $\hat{f}$ distribution.

Now, because $\hat{g} \neq g$, there exists some $k \in (\frac{a}{\mu}, \frac{b}{\mu})$ and $\epsilon > 0$ such that

$$\int_{k-\epsilon}^{k+\epsilon} \hat{g}(x)dx = \alpha > 0 \text{ and } \int_{k-\epsilon}^{k+\epsilon} x\hat{g}(x)dx = \alpha k.$$

Define the following p.d.fs:

$$g_1(x) = \begin{cases} \frac{1}{1-\alpha}\hat{g}(x), & x \notin [k-\epsilon, k+\epsilon]; \\ 0, & x \in [k-\epsilon, k+\epsilon]. \end{cases}$$

$$g_2(x) = \begin{cases} \frac{1}{\alpha}\hat{g}(x), & x \in [k-\epsilon, k+\epsilon]; \\ 0, & x \notin [k-\epsilon, k+\epsilon]. \end{cases}$$

$$g_3(x) = \begin{cases} \zeta = \frac{b-k\mu}{b-a}, & x = \frac{a}{\mu}; \\ 1-\zeta, & x = \frac{b}{\mu}; \\ 0, & \text{otherwise.} \end{cases}$$

Hence $\hat{g}(x) = g_1(x)$ with probability $1-\alpha$ and $\hat{g}(x) = g_2(x)$ with probability α. Define $\tilde{g}(x)$ as a modification of $\hat{g}(x)$ which has a GOOD-BAD server mode defined by g_3, i.e.

$$\tilde{g}(x) = \begin{cases} g_1(x), & \text{w.p. } 1-\alpha; \\ g_3(x), & \text{w.p. } \alpha. \end{cases}$$

The loss in throughput under the $\tilde{g}$ distribution is, in fact *larger* than the loss in throughput under the $\hat{g}$ distribution. Let $\hat{G}$ and $\tilde{G}$ denote the cdfs for $\hat{g}$ and $\tilde{g}$, respectively.

$$\begin{aligned} E_{\tilde{g}}[\max_u Z_u] &- E_{\hat{g}}[\max_u Z_u] \\ &= \int_{\frac{a}{\mu}}^{\frac{b}{\mu}} \Big[1-\tilde{G}^U(x)\Big]dx - \int_{\frac{a}{\mu}}^{\frac{b}{\mu}} \Big[1-\hat{G}^U(x)\Big]dx \\ &= \alpha\int_{\frac{a}{\mu}}^{\frac{b}{\mu}} G_3^U(x)dx - \alpha\int_{\frac{a}{\mu}}^{\frac{b}{\mu}} G_2^U(x)dx \\ &= \alpha\zeta^U\Big(\frac{b-a}{\mu}\Big) - \alpha\int_{k-\epsilon}^{\frac{b}{\mu}} G_2^U(x)dx \\ &\le \alpha\zeta^U\Big(\frac{b-a}{\mu}\Big) - \alpha(\frac{b}{\mu} - k + \epsilon) \\ &= \alpha\Big[\Big(\frac{b-k\mu}{b-a}\Big)^U\Big(\frac{b-a}{\mu}\Big) - \frac{b}{\mu} + k - \epsilon\Big] \\ &< \alpha\Big[\Big\{\zeta^{U-1} - 1\Big\}(\frac{b}{\mu} - k)\Big] \\ &\le 0 \end{aligned}$$

The first equality comes from the fact the server states are i.i.d. The second inequality come from the definition of $\hat{g}$ and $\tilde{g}$. The third equality comes from the definition of g_3 and g_2 (which is 0 for $x < k-\epsilon$). The first inequality comes from the fact that $G_2(x) \le 1$ for all x. The last equality comes from the fact that $\zeta \in [0,1]$.

Therefore, the loss in throughput due to unknown server state is larger for $\tilde{g}$ than for $\hat{g}$, which has some GOOD-BAD properties. This contradicts the maximal loss of throughput of $\hat{g}$. Hence, the GOOD-BAD server has the worst loss of throughput for a given support and mean server quality/speed. The loss of throughput in this case is ($\gamma = \frac{b-\mu}{b-a}$):

$$\begin{aligned} E_f[\max_{u\in\mathcal{U}} s_u] &= \Big(\Big[1-(1-\gamma)^U\Big]\frac{b}{\mu} + (1-\gamma)^U\frac{a}{\mu}\Big)E_f[s_u] \\ &= \Big(\frac{b}{\mu} + (1-\gamma)^U\frac{a-b}{\mu}\Big)E_f[s_u] \\ &\to \frac{b}{\mu}E_f[s_u] \text{ (as } U \to \infty) \end{aligned}$$ ■

We have just shown for a fixed support and mean for the i.i.d. server state distribution, the GOOD-BAD is the worst it can get. This is true without the constraint of a fixed mean.

THEOREM 5. *(I.I.D GOOD-BAD Servers) Suppose the support for each server state is given by a closed interval* $[a,b] \subseteq [0,1]$*, so that* $P(s_u \notin [a,b]) = 0$*. Then the maximum loss in throughput is given by:*

$$\frac{J^o(t)}{J^*(f^t,t)} \le (\frac{b}{a} \wedge U) \tag{15}$$

and is achieved with the following distribution with $\gamma \to 0$*:*

$$s_u = \begin{cases} b, & \text{w.p. } \gamma; \\ a, & \text{w.p. } 1-\gamma. \end{cases}$$

PROOF. Again, consider the case where $\frac{b}{a} < U$, since by Theorem 3 $E_f[\max_{u\in\mathcal{U}} s_u] \le UE_f[s_u]$ for any distribution of s_u. By Lemma 2, for fixed mean μ, the GOOD-BAD server achieves the worst loss in throughput given by $\frac{b}{\mu}$. Maximizing this over the support $[a,b]$ means minimizing μ, which can approach a as $\gamma \to 0$ for the given GOOD-BAD distribution.

Therefore the throughput loss in a single time slot for i.i.d. server with support in $[a,b]$ is given by $\frac{b}{a} \wedge U$. By (14), for statistically static server states over time, the loss over T timeslots is also $\frac{b}{a} \wedge U$.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

This yields the desired result. ■

It can be easily seen that when restricted to GOOD-BAD server over a given support, the i.i.d case results in the worst throughput loss.

We have closely examined the case of i.i.d server states and have shown that the GOOD-BAD distribution corresponds to the largest throughput loss due to unknown server state. Intuitively, GOOD-BAD servers are as disparate as possible. Hence, when the π^* policy misses a server which is in the GOOD state while the clairvoyant policy is able to use it, the π^* policy may end up scheduling a server in the BAD state. The quality of this BAD state compared to the quality of the GOOD state is the worst it can get. Hence, the GOOD-BAD server leads to the worst throughput. For servers with "smoother" distributions, the loss of throughput should improve.

5. I.I.D. UNIFORM SERVERS STATES

In some instances, the server state may be known to fall within a fixed range; however, its exact state is unknown. A uniform distribution over this interval is one way to model such a scenario. If the probabilities of successful service by each server were independent and uniformly distributed, then the bound in Theorem 3 can be significantly improved to a constant factor, independent of the number of servers in the system.

Let a_u and b_u define the uniform distribution of the quality/speed of server u, so that

$$f_u(x) = \begin{cases} \frac{1}{b_u - a_u}, & x \in [a_u, b_u]; \\ 0, & \text{otherwise.} \end{cases}$$

Then, the throughput loss due to unknown server state is bounded by 2.

THEOREM 6. *(Throughput Loss on I.I.D. Uniform Server States) If in each time slot the quality/speed of server u is uniformly distributed from $[a_u, b_u]$, then the throughput loss due to unknown server state is bounded by a factor of 2, that is,*

$$\frac{J^o(t)}{J^*(f^t, t)} \leq 2. \tag{16}$$

PROOF. Without loss of generality assume that the servers are ordered such that $b_1 \leq b_2 \leq \cdots \leq b_{U-1} \leq b_U$. Furthermore, the exists some sequence $\{n_1, n_2, \ldots, n_U\}$ such that $a_{n_1} \leq a_{n_2} \leq \cdots \leq a_{n_U}$. Without loss of generality we can assume that $a_{n_U} \leq b_1$. If this were not true, then the probability of successful service by server 1 will *always* be less than that to server n_U: $P(s_1 < s_{n_U}) = 1$ and server 1 will never be scheduled. Hence, this is the same scheduling scenario as having $U' = U - 1$ servers. Therefore, we will assume that $a_{n_U} \leq b_1$.

Based on the ordering of servers, it is easy to see that $E_f[\max_u s_u] \leq b_U$. Clearly, this is only achieved with equality if $a_U = b_U$, which would then imply that $\max_u E_f[s_u] = b_U$. Then the clairvoyant and unknown server state policies will coincide, $\pi^o = \pi^*$ and there will be no loss of throughput due to unknown server state. However, if $a_U \neq b_U$, then there may be some loss. Now, because $a_U \geq 0$, $\max_u E_f[s_u] \geq E_f[s_U] \geq \frac{b_U}{2}$. Hence:

$$E_f[\max_u s_u] \leq b_U \leq 2 \max_u E_f[s_u].$$

This is the throughput loss in one time slot. By (14), this implies the same loss over t time slots (certainly for $t = T$) so that:

$$J^o(t) \leq 2J^*(f^t, t) \qquad \blacksquare$$

This bound is tight as shown by the following example.

EXAMPLE 3. *Suppose that $T = 1$ and the server states are i.i.d. and uniformly distributed on $[0, 1]$. Then $E_f[s_u] = \frac{1}{2}$ and*

$$\begin{aligned} E_f[\max_u s_u] &= \int_0^1 \left[1 - F^U(x)\right] dx \\ &= 1 - \int_0^1 x^U dx \\ &= 1 - \frac{1}{U+1} \\ &\to 1 \text{ (as } U \to \infty) \end{aligned}$$

Hence,

$$\frac{J^o(1)}{J^*(f^1, 1)} = \frac{E_f[\max_u s_u]}{E_f[s_u]} = 2 - \frac{2}{U+1} \to 2 \text{ (as } U \to \infty).$$

This example shows that we can get arbitrarily close to the bound in Theorem 6 by increasing the number of servers in the system.

The contrast between the loss of throughput due to unknown server state for GOOD-BAD servers versus uniformly distributed server states is striking. In Example 3, the servers were i.i.d. In this case, even if the policy with unknown server state, π^*, schedules a different server than the clairvoyant policy π^o the success probabilities are likely to be similar. The uniform distribution is the least disparate while the GOOD-BAD distribution is the most. We expect the loss of throughput for other types of server state distributions to fall in between the bounds given for these distributions in Theorems 5 and 6.

6. EXPERIMENTAL INVESTIGATION AND VERIFICATION: WIRELESS PACKET SCHEDULING

Thus far, our results have been about general distributions and classes of distributions for server states. We have discussed the results in terms of a general model which a number of scheduling problems of interest can be cast. In this section, we consider the application of wireless packet scheduling and study the loss of throughput due to unknown channel state via a numerical study of specific channel types.

In wireless packet scheduling a single transmitter has a series of packets to transmit to U users over channels dedicated to each user. Due to varying path-loss, fading, and interference, the channel quality is random and varies over time. The goal is to determine a scheduling policy which selects a user to transmit a packet to while considering the communication channel quality and maximizing throughput.

Mapping wireless packet scheduling to the general model means that 'items' correspond to packets, servers corresponds to users's channels, and successful service corresponds to successful transmission of packets.

6.1 I.I.D. ON-OFF Channels

In Theorem 3, we proved that the worse case throughput loss is given by the number of users with unknown server state. We showed that this bound was tight via Example 1. In order to achieve this worse case bound, the server distributions are i.i.d. ON-OFF channels with probability of being ON, $P(s_u = 1) = \epsilon$, approaching zero, $\epsilon \to 0$. In Fig. 1, we examine the loss of throughput for unknown channel state over a single time slot as a function of the number of users, U, with unknown channel state. The channels are i.i.d. ON-OFF with $P(s_u = 1) = 1 - P(s_u = 0) = \epsilon$. Certainly, as U increases, the loss of throughput also increases. However, $\frac{J^o}{J^*}$ does not approach U until $\epsilon < 10^{-3}$. This shows that in or-

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

der to approach this worse case bound, the channels must be *highly* degenerate. It is reasonable to assume that most communications will not occur in this regime and that the loss of throughput will be much lower. In fact, for i.i.d. channels where $P(s_u = 1) = .1$, the loss of throughput is at most a factor of 10. This is still a very degenerate scenario with packet error rates of .9. Communication is undesirable and unlikely to occur in environments with such high packet error rates.

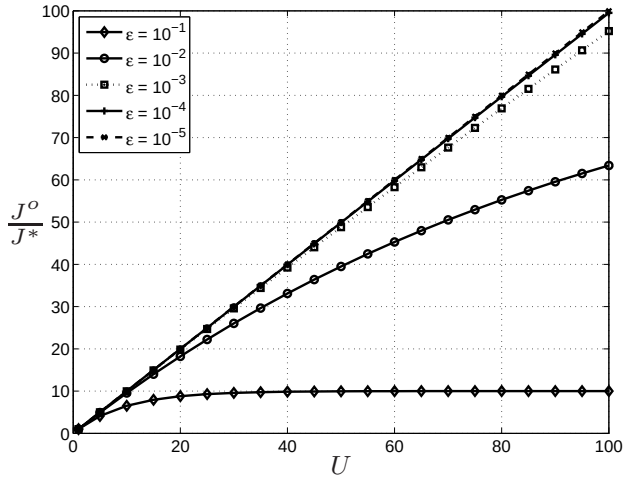

Figure 1: Throughput loss for i.i.d. ON-OFF channels as a function of the number of users U. $P(s_u = 1) = 1 - P(s_u = 0) = \epsilon$.

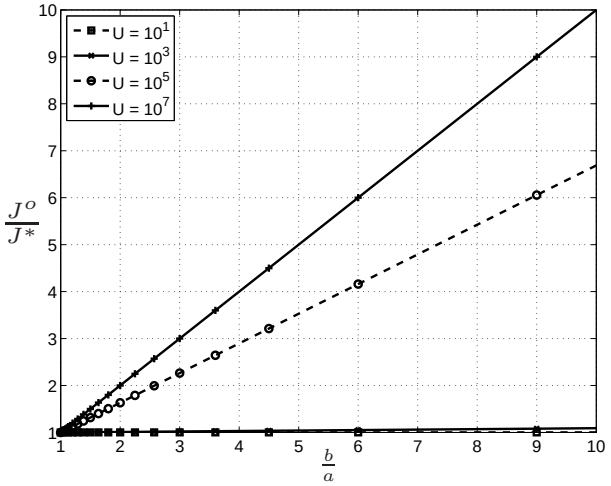

Figure 2: Throughput loss for i.i.d. GOOD-BAD channels as a function of the ratio of GOOD versus BAD quality, $\frac{b}{a}$. $P(s_u = b) = 1 - P(s_u = a) = \gamma = 10^{-5}$.

6.2 I.I.D. Good-Bad Channels

Here we examine the case of GOOD-BAD channels where the probability of successful transmission in the GOOD state, b, is higher than in the BAD state, a, i.e. $b > a$. From Theorem 5, the loss of throughput in the case of i.i.d. GOOD-BAD channels is given by $\frac{b}{a}$ and this is achieved in the limit as $P(s_u = b) = \gamma \to 0$. We consider the case of a single transmission time slot where $\gamma = 10^{-5}$, $b = .9$, and a varies from $[.1, .9]$. The probability of being in the BAD state is $1 - \gamma = 1 - 10^{-5} \sim 1$. When $\frac{b}{a}$ is large (a is small), this is a highly degenerate scenario. We see in Fig. 2 for reasonable numbers of users $U \leq 10^3$, the throughput when channel state is unknown is nearly equal to that when channel state is known. However, for very large numbers of users, the loss of throughput approaches the bound given by Theorem 5. For smaller γ, these losses are likely to be smaller, as suggested by Fig. 1.

6.3 I.I.D. Uniform Channels

We have seen that these bi-modal distributions for channels, where each channel can take on one of two states, can lead to large losses in throughput. In Theorem 6, we saw that the loss of throughput for uniform channels is bounded by a constant factor, 2. In Fig. 3, we see the loss of throughput due to unknown channel state as a function of the number of users, U. Each of the U channels are i.i.d. uniformly over $[0, 1]$. As predicted, we can see that as $U \to \infty$, $\frac{J^0}{J^*} \to 2$.

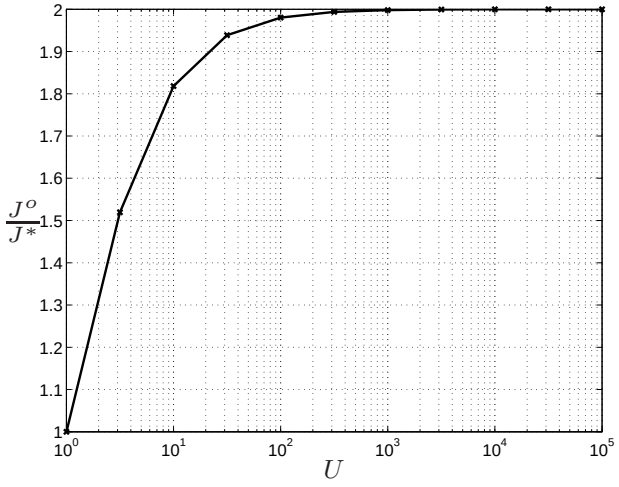

Figure 3: Throughput loss for i.i.d. uniform $[0, 1]$ channels as a function of the number of users U.

6.4 Markov Channels

Thus far we have examined the throughput loss over channels which are statistically static over time. However, wireless channels are often time-varying. As such a dynamic channel model is necessary–Markov models for dynamic channels are popular. We consider the commonly used, 2-state Gilbert-Elliot channel [17] depicted in Fig. 4. Due to the symmetry of the channel, the steady state distribution is an ON-OFF channel with $\gamma = .5$.

We assume that the scheduler knows the Markov model for the $U = 10$ i.i.d. channels. However, it does not know the *current* channel states prior to making the transmission decision. Following the transmission, the channel state information are updated so that the channel state estimates, $\{g_u^t\}$ (or equivalently, $\{f_u^t\}$), are given by the perfectly known channel states in the previous time slot. This might be the case if channel state information is transmitted from each user to the base station in each time slot, but due to transmission delays, it does not arrive in time for the transmission decision in the current time slot to be based on this information.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

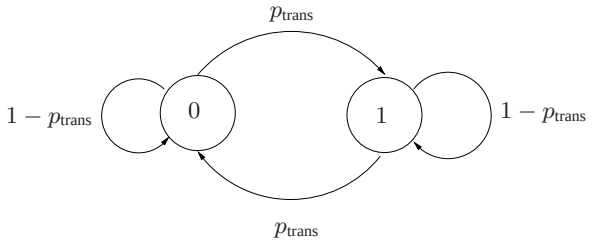

Figure 4: Gilbert-Elliot channel.

However, the transmission decision in the subsequent time slot can use this information.

We see the loss of throughput due to unknown channel state for Markov channels in Fig. 5 as a function of the state transition probability, p_{trans}. Our experiments are over a time horizon of $T = 100$ time slots and averaged over 1000 realizations. When $p_{\text{trans}} = .5$, the channels are i.i.d. over time which, by Lemma 2 gives that the T period loss in throughput is 2. However, for $p_{\text{trans}} \neq .5$, the history of the channel, i.e. the channel state in the prior time slot, provides information about the current channel state and hence performance of the π^* policy is improved. The symmetry in the figure is due to the symmetry in "information" gained with information about the channel state in the previous time slot. Due to the Markovian property, this information defines a GOOD-BAD channel in the current time slot. The GOOD-BAD distribution is given by the previous channel state as well as the transition probability. Being in the GOOD state with transition probability, p_{trans} will result in the same GOOD-BAD distribution as being in the BAD state with transition probability, $1 - p_{\text{trans}}$.

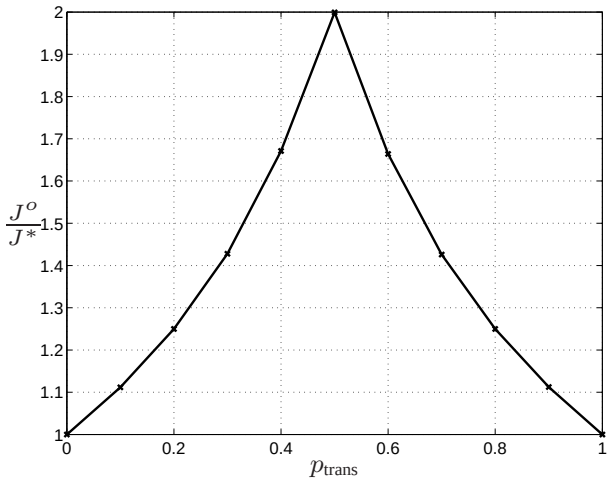

Figure 5: throughput loss for ON-OFF Markovian channels as a function of the probability of channel state transition, p_{trans}.

6.5 Diverse Channel Classes

Most of our analysis has focused on the case where channels are i.i.d. We now examine the case of mixed channel types. Our experiments are over a time horizon of $T = 100$ time slots and averaged over 1000 realizations. We assume there are $U = 20$ users. $U_{\text{ON-OFF}}$ users are ON-OFF channels with $P(s_u = 1) = p_{ON}$ and $U_{\text{unif}} = U - U_{\text{ON-OFF}}$ users are uniform $[0, 1]$ channels. We know that if all channels are uniform, $U_{\text{unif}} = U$, then the throughput loss is bounded by 2. Conversely, if all channels are ON-OFF, $U_{\text{ON-OFF}} = U$, then the throughput loss is bounded by $\min\{\frac{1}{p_{\text{ON}}}, U\}$. For $\mu_{\text{ON-OFF}} = p_{\text{ON}} < .5$, the expected value of the ON-OFF channels is less than that of the the uniform channels $\mu_{\text{unif}} = .5$; hence, the policy with unknown channel state will always choose to transmit to a user with the uniform distribution. Fig. 6 shows the loss in throughput when there is a mix of ON-OFF and uniform channels. We can see that in this case the loss of throughput is quite small ($\sim 1 - 2$). However, when there are no channels with uniform distribution ($U_{\text{ON-OFF}} = 20$), the policy with unknown channel state must pick one of the ON-OFF channels at random, and the throughput loss approaches the bound given by Lemma 2.

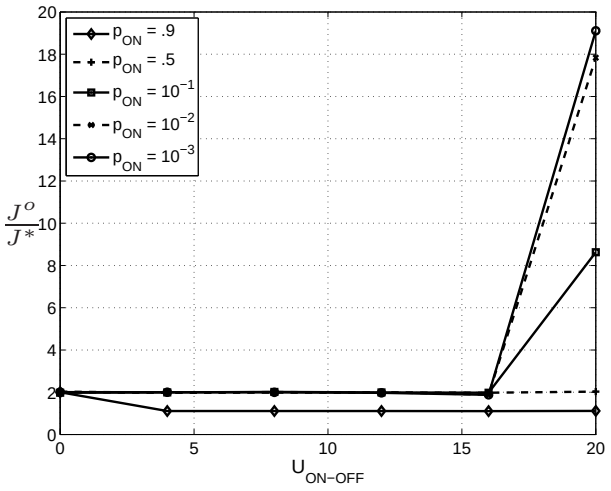

Figure 6: Throughput loss for a mix of ON-OFF channels and uniform $[0, 1]$ channels. $U_{\text{ON-OFF}} + U_{\text{unif}} = U = 20$.

7. CONCLUSION

In many scheduling applications, obtaining perfect server state information is a difficult estimation problem. It is often only possible to obtain an estimate of the quality of server. Certainly imperfect knowledge of server state will lead to a loss in performance. We have examined the loss of throughput due to unknown server state in the case of throughput maximization in a general scheduling model. Applications of wireless and Internet packet scheduling as well as product line design can be cast within this general model.

The loss in throughput is bounded by the number of servers with unknown state; this bound is tight. Under certain distributions of server quality, this bound can be improved. To achieve these bounds, highly degenerate server distributions are necessary, which suggests that the loss of throughput due to unknown server state are likely to be much smaller in practice.

Gathering estimates for server states can be an expensive process. A natural question is, given limited resources, which servers should be probed in order to minimize the loss due to unknown server state. In light of our study, it seems that servers with highly disparate quality often lead to the largest losses. Avoiding these types of distributions by investing more resources to gather more accurate estimations may significantly improve performance. This question of resource allocation to improve performance and reduce the effect of server state uncertainty is an interesting and continuing

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547

area of research.

8. REFERENCES

[1] S. Borst and P. Whiting, "Dynamic rate control algorithms for hdr throughput optimization," in *Proc. IEEE INFOCOM*, vol. 2, Apr. 2001, pp. 976 – 985.

[2] X. Liu, E. K. P. Chong, and N. B. Shroff, "Transmission scheduling for efficient wireless utilization," in *Proc. IEEE INFOCOM*, vol. 2, Apr. 2001, pp. 776 – 785.

[3] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1986–1992, Nov. 1997.

[4] V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel state information for throughput maximization," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2566–2582, Nov. 2004.

[5] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. ACM Sigcomm*, vol. 32, Oct. 2002, pp. 89–102.

[6] S. Athuraliya, S. Low, V. Li, and Q. Yin, "REM: active queue management," *IEEE Netw.*, vol. 15, May 2001.

[7] G. van Ryzin and S. Mahajan, "On the relationship between inventory costs and variety benefits in retail assortments," *Management Science*, vol. 45, no. 11, pp. 1496–1509, 1999.

[8] W. Hopp and X. Xu, "Product line selection and pricing with modularity," *Manufacturing and Service Operations Management*, vol. 7, no. 3, pp. 172–187, 2005.

[9] M. Medard, "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 933–946, May 2000.

[10] T. Yoo and A. Goldsmith, "Capacity and power allocation for fading mimo channels with channel estimation error," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2203–2214, May 2006.

[11] P. Chaporkar, A. Proutiere, H. Asnani, and A. Karandikar, "Scheduling with limited information in wireless systems," in *Proc. ACM MobiHoc*, 2009, pp. 75–84.

[12] K. Pruhs, "Competitive online scheduling for server systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 4, pp. 52–58, 2007.

[13] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[14] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[15] A. Lapidoth and S. Shamai, "Fading channels: How perfect need 'perfect side information' be?" *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1118–1134, May 2002.

[16] A. Vakili, M. Sharif, and B. Hassibi, "The effect of channel estimation error on the throughput of broadcast channels," in *Proc. IEEE ICASSP*, vol. 4, May 2006, pp. 29–32.

[17] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Systems Technical Journal*, vol. 42, pp. 1977–1997, Sep. 1963.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7547
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7547