Timely-Throughput Optimal Scheduling with Prediction

Kun Chen and Longbo Huang
IIIS, Tsinghua University
chenkun14@mails.tsinghua.edu.cn, longbohuang@tsinghua.edu.cn

***Abstract*—Motivated by the increasing importance of providing delay-guaranteed services in general computing and communication systems, and the recent wide adoption of learning and prediction in network control, in this work, we consider a general stochastic single-server multi-user system and investigate the fundamental benefit of predictive scheduling in improving timely-throughput, being the rate of packets that are delivered to destinations before their deadlines. By adopting an error rate-based prediction model, we first derive a Markov decision process (MDP) solution to optimize the timely-throughput objective subject to an average resource consumption constraint. Based on a packet-level decomposition of the MDP, we explicitly characterize the optimal scheduling policy and rigorously quantify the timely-throughput improvement due to predictive-service, which scales as** $\Theta(p\left[C_1\frac{(a-a_{\max}q)}{p-q}\rho^{\tau}+C_2(1-\frac{1}{p})\right](1-\rho^{D}))$**, where** $a, a_{\max}, \rho \in (0,1), C_1 > 0, C_2 \geq 0$ **are constants,** p **is the true-positive rate in prediction,** q **is the false-negative rate,** τ **is the packet deadline and** D **is the prediction window size. We also conduct extensive simulations to validate our theoretical findings. Our results provide novel insights into how prediction and system parameters impact performance and provide useful guidelines for designing predictive low-latency control algorithms.**

I. Introduction

How to provide low-latency packet delivery has long been an important problem in network optimization research, particularly due to the increasingly more stringent user delay requirements in a wide range of applications. For instance, low delay is critical for video traffic in mobile networks, which has already accounted for 60% of total mobile data in 2016 and will account for more than 78% by 2021 according to a recent Cisco report [1]. Other areas such as online gaming, online health care and supply chain also have rigid delay requirements. Indeed, user requirements are so strong, that it has been reported that for companies like Amazon and Google, if their service latency increases by 500ms, they will lose 1.2% of their customers and millions of dollars revenue [2]. As a result, the problem of guaranteeing low-latency has received much attention in the last decade, and many scheduling algorithms have been designed based on various mathematical techniques, e.g., [3], [4], [5], [6], [7], [8], [9].

On the other hand, driven by the availability of large amount user behavior data and the rapid development of data mining and machine learning tools, it has become common in practice to *predict* user demand and to *proactively* serve customer requests. For example, Amazon predicts what customers may purchase and pre-ships products to distribution centers close to them, in order to reduce shipping time [10]. Netflix also tries to predict what customers may want and preload videos onto user devices to improve quality-of-experience [11]. Another example is brunch prediction in computer architecture, where prediction is used to decide how to pre-execute certain parts of the workload, so as to reduce computing time [12]. Despite the continuing success of this prediction-based approach in practice, it has not received much attention in theoretical study. Therefore, it remains largely unknown how prediction can fundamentally improve delay-guaranteed services.

In this paper, we aim to fill this gap and investigate *the impact of prediction on timely-throughput*. Specifically, we consider a single-server multi-user system where the server delivers packets to users. Each packet has a user-dependent deadline before which it needs to reach the user. The service channel for each user is time-varying and the transmission success probability depends on the resource spent sending a packet. The server gets access to an *imperfect* prediction window, in which forecasts about future arrivals are available, and can *pre-serve* packets before they actually enter the system. The overall objective of the system is to maximize a weighted sum of timely-throughputs of users, being the rates of packets delivered before their deadlines. This formulation is general and models various important practical applications, e.g., video streaming, sending time-critical control information, and grocery delivery.

There has been an increasing set of recent results investigating the impact of prediction in networked system control. [13] and [14] consider utility optimal scheduling in downlink systems based on perfect user prediction. [15] shows that proactive scheduling can effectively reduce queueing delay in stochastic single-queue systems. [16] considers how network state prediction can be incorporated into algorithm design. [17] and [18] focus on understanding the cost saving aspect of proactive scheduling based on demand prediction. [19], [20] and [21] also investigate the benefit of prediction from an online algorithm design perspective. However, we note that the aforementioned works all focus on understanding the utility improvement aspect of prediction and proactive service, and delay saving often comes as a by-product of the resulting predictive control algorithms. Thus, the results are not applicable to delay-constrained problems, where meeting the latency guarantee is an explicit requirement.

Our formulation is closest to recent works [8], [22], [13], and [9], which focus on delay-constrained traffic scheduling. Our work is different as follows. [8] focuses on the setting where traffic is generated and delivered within synchronized frames for all users and [22] focuses on periodic traffic, while our work allows heterogeneous deadlines for user packets and random arrivals. [13] focuses on optimizing system utility subject to stability constraint, while we work explicitly with delay constraints. Lastly, while our work builds upon the novel results in [9], we focus on quantifying the impact of prediction and proactive service in a Markov system, whereas [9] considers a causal system with an i.i.d. setting. The extension to incorporate prediction significantly complicates the solution and analysis. Our results offers novel insights into the benefits of prediction in delay-constrained network control.

The main contributions are summarized as follows.

(i) We propose a novel framework for studying timely-throughput optimization with imperfect prediction and proactive scheduling. Our model captures key features of practical delay-constrained problems and facilitates analysis.

(ii) We derive the exact optimal solution to the prediction-based timely-throughput optimization problem using Markov decision process (MDP). We rigorously quantify that prediction improves timely-throughput by $\Theta(p\left[C_1\frac{(a-a_{\max}q)}{p-q}\rho^{\tau}+C_2(1-\frac{1}{p})\right](1 - \rho^D))$, where $a, a_{\max}, C_1 > 0, C_2 \geq 0, \rho \in (0,1)$ are constants, p is the true-positive rate in prediction, q is the false-negative rate, τ is the packet deadline and D is the prediction window size. This concise and explicit characterization provides insights into how different parameters impact system performance.

(iii) We conduct extensive simulations to validate our theoretical findings. Our results show that prediction-based system control can significantly boost timely-throughput.

The rest of the paper is organized as follows. In Section II, we present the system model. The MDP-based solution is presented in Section III. Structural properties of the optimal solution and exact timely-throughput improvement for a static setting are derived in Section IV. The general scenario is considered in Section V. Simulation results are presented in Section VI and conclusion comes in Section VII.

Due to space limitation, we have omitted all proofs. Readers please refer to our technical repport [23] for details.

II. System Model

Consider a general single-server system with N users as shown in Fig. 1. The server can simultaneously transmit multiple packets to different users with cost due to resource expenditure, e.g., energy consumption. The channels are unreliable and transmissions may fail. Each packet has a hard deadline within which it must be delivered successfully. Otherwise, it becomes outdated and will be useless for the receiver. We assume that time is discrete, i.e., $t \in \{0, 1, \dots\}$, and a packet transmission to any user takes one time-slot.

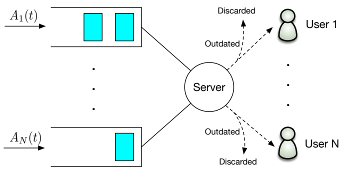

Fig. 1. A single-server multi-user system where packets have hard deadlines.

A. The Delay-Constrained Traffic Model

The number of packet arrivals destined for user n at time t is denoted by $A_n(t)$. We assume that $A_n(t)$ is i.i.d across time and independent for different users, with an average rate $\mathbb{E}\{A_n(t)\} = a_n$. We also assume the number of packet arrivals is bounded for all time and for all users, i.e., $0 \leq A_n(t) \leq A_{\max}, \forall\, n, t$. For each user n, there is a hard deadline or sustainable delay for his packets, denoted by τ_n. This means that for any packet in $A_n(t)$, it should be successfully delivered by time $t + \tau_n$. Otherwise, it becomes useless and will be discarded from the system at time $t + \tau_n$. We further assume $\tau_n \leq \Gamma, \forall n$, for some finite constant Γ.

B. The Service Model

The system serves user packets by transmitting them over service channels, at the expense of resource consumption, e.g., energy. To model system dynamics, we assume the success of a packet transmission for user n is a random event and its probability is determined by the instantaneous condition of the service channel, denoted by the channel state $S_n(t)$, which is modeled by an ergodic finite-state Markov chain with state space $\mathcal{S} \triangleq \{s_1, \dots, s_K\}, \forall n$. The transition matrix and the stationary distribution are denoted as $(P_n^{i,j})_{K\times K}$ and $\boldsymbol{\eta}_n = (\eta_n^1, \dots, \eta_n^K)$, respectively.

At every time t, the server needs to decide the resource consumption level for transmitting each present packet, which is chosen from a bounded set of consumption levels $\mathcal{E}$. If at time t, the channel state is s_i and the resource level is e, then the probability of a successful packet transmission for user n is $\zeta_n(i, e)$. $e = 0$ means that a packet will not be transmitted and $\zeta_n(i, 0) = 0$. Also, $\zeta_n(i, e) > 0$ for all $e > 0$. We further assume that $\zeta_n(i, e)$ is a concave and strictly increasing function of e. We assume that there is a total order on set $\mathcal{S}$ based on $\zeta_n(i, e)$, i.e., for each pair of i, j, either $\zeta_n(i, e) \geq \zeta_n(j, e), \forall e$ or $\zeta_n(i, e) \leq \zeta_n(j, e), \forall e$. We also assume that there is no hard capacity constraint for the server, i.e., it can transmit an arbitrary number of packets every time, although it has to maintain an average resource consumption guarantee (we consider the setting with hard capacity constraint in [23]). This assumption is made to facilitate analysis and was also adopted in [9].

C. The Predictive Service Model

Different from prior results in the literature that often only consider *causal* systems, we are interested in understanding how prediction and predictive-service fundamentally impact system performance. Thus, we assume that the system gets access to a *prediction* window $\mathcal{D}_n(t) = \{A_n(t+1), \dots, A_n(t+D_n)\}$ for each user. Moreover, the system implements *predictive service*, i.e., it tries to pre-serve future arrivals in $\mathcal{D}_n(t)$ in the current time slot. Such scenario is common in practice. For instance, Amazon predicts user behavior and pre-ships goods to distribution centers closest to users [10].

In this work, we focus on two prediction models.

1) Perfect prediction: In this case, the predicted arrivals in $\mathcal{D}_n(t)$ are exact. This is an idealized case and results in this case will serve as an upper bound for the benefit of predictive scheduling. Such a perfect prediction model has been used in the literature, eg., [13] and [14].

2) Imperfect prediction: Here prediction made by the system can contain error. Specifically, we adopt the imperfect prediction model parameterized by the true-positive and false-negative rates as follows. Each predicted arrival in the prediction window is correct with probability p_n, and every actual packet arrival will be missed with probability q_n, i.e., a packet will arrive unexpectedly with probability q_n, as illustrated in Fig. 2. Thus, the true-positive rate is p_n and the false-negative rate is q_n. These two rates are decided by the learning methods used to forecast future arrivals and our analysis holds for general p_n and q_n. Without loss of generality, we assume $p_n > q_n$[1] (perfect prediction corresponds to $p_n = 1$ and $q_n = 0$). This model was previous adopted in [13] and [15].

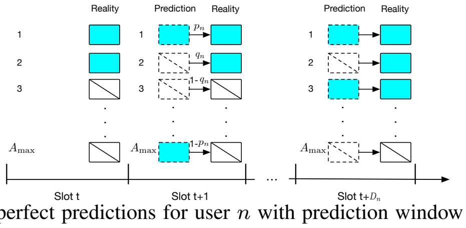

Fig. 2. Imperfect predictions for user n with prediction window $\mathcal{D}_n(t)$. The dashed packets are predictions and the solid packets are the actual outcome. There are $A_{\max}$ possible arrivals in each time-slot and the server makes a prediction for each packet. The (dashed) blue box means that the server predicts an arrival (positive prediction) and the correct probability is p_n. The (dashed) crossed white box means that the server predicts no arrival (negative prediction), but a packet may come with probability q_n. At slot t, predicted arrivals will enter the system and the server sees the actual realizations.

D. System Objective

We define the *timely-throughput* as the average number of packets delivered successfully before their deadlines, i.e.,[2]

$$x_n = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\big\{\sum_{t=1}^{T} X_n(t)\big\}, \tag{1}$$

where $X_n(t)$ denotes the number of packets that timely reach their destinations for user n at time t. We also define the average resource expenditure as:

$$E_{av} = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\big\{\sum_{t=1}^{T}\sum_{n=1}^{N} E_n(t)\big\}, \tag{2}$$

where $E_n(t)$ is the resource consumed by transmissions of packets for user n at time t.

Denote $\boldsymbol{x} = (x_1, \dots, x_N)$. Given a weight vector $\boldsymbol{\beta} = (\beta_1, \dots, \beta_N)$ with $\beta_n \geq 0, \forall 1 \leq n \leq N$, the *weighted timely-throughput* ϕ is defined as $\phi \triangleq \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}$. In this paper, we focus on the problem of maximizing the weighted timely-throughput, subject to an average resource constraint B, i.e.,

$$\phi^* \triangleq \max \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}, \quad \text{s.t. } E_{av} \leq B. \tag{3}$$

This formulation is general and models many delay-constrained applications, e.g., video streaming [24] and supply chain optimization [25].

E. Model Discussion

Our work builds upon the recent work [9]. However, our model and results are different as follows. (i) We consider a Markov model for the system while [9] focuses on an i.i.d. setting. (ii) We focus on prediction and predictive scheduling and quantify their fundamental impact on timely-throughput, while previous delay-constrained results, e.g., [9] and [8], consider causal systems. Analysis for predictive systems is significantly complicated due to prediction errors and requires different arguments compared to those for causal systems.

III. Scheduling by Packet-Level Decomposition

Problem (3) can be formulated as a constrained Markov Decision Process (MDP). However, the number of system states can grow exponentially large, making it complicated to obtain efficient algorithms. To tackle this issue, we adopt the packet-level decomposition approach in [9] and extend it to handle prediction in our setting. Specifically, for every individual packet still in the system at time t, its state is described by the user it belongs to and a triple (r, τ, i). Here r denotes the reception status of the packet, i.e., $r = 0$ means that the packet is at the source and $r = 1$ means that the packet has reached the destination. τ is the time duration before its deadline, and i is the channel state index. Then, the state of the system at time t can be described by the state of all packets. Since the number of arrivals at any time-slot is bounded by $NA_{\max}$, each packet can stay in the system for no more than Γ slots, and the number of channel states is finite, the system state size is finite (though can be exponentially large).

A (possibly randomized) scheduling policy π decides at each system state, which packets to transmit and at what resource levels. Since the distribution of system state at time $t + 1$ is decided by the state and the scheduling decision at time t, problem (3) is a constrained MDP with finite states, which can be solved with algorithms such as value iteration or policy iteration [26].

A. Packet-Level Decomposition for the Constrained MDP

The Lagrangian of (3) can be written as

$$\begin{aligned} L(\pi, \lambda) = \lim_{T\to\infty} \frac{1}{T}\Big[&\mathbb{E}\big\{\sum_{n=1}^{N}\sum_{t=1}^{T} \beta_n X_n(t)\big\} \\ &-\lambda\mathbb{E}\big\{\sum_{t=1}^{T}\sum_{n=1}^{N} E_n(t)\big\}\Big] + \lambda B, \end{aligned} \tag{4}$$

with a Lagrange multiplier λ. $\sum_{n=1}^{N}\sum_{t=1}^{T}\beta_n X_n(t)$ counts the timely deliveries of packets, and $\sum_{t=1}^{T}\sum_{n=1}^{N} E_n(t)$ comprises the resource consumed. Further notice that there is no capacity constraint for the server. Thus, by denoting $\mathcal{A}_n(T)$ the set of packet arrivals for user n up to time T, the Lagrangian can be decomposed into the following packet-level form:

$$L(\pi, \lambda) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\big\{\sum_{n=1}^{N}\sum_{\xi\in\mathcal{A}_n(T)} [\beta_n\delta(\xi) - \lambda E(\xi)]\big\} + \lambda B, \tag{5}$$

where $\delta(\xi)$ is the indicator that packet ξ reaches the destination before its deadline and $E(\xi)$ is its total resource consumption.

From (5), the term related to packet ξ of user n is

$$\mathbb{E}\big\{\beta_n\delta(\xi) - \lambda E(\xi)\big\}. \tag{6}$$

As a result, maximizing the Lagrangian (4) can be accomplished by maximizing (6) for each packet. In the following, we refer to problem (6) as the *Single Packet Scheduling*

[1] Otherwise, one can inverse the predictions to ensure that a positive prediction is more likely to become a true arrival than a negative prediction.

[2] In this paper, we assume all limits in consideration exist with probability 1. The more general case can be tackled with $\lim\sup$ or $\lim\inf$ arguments.

Problem (SPS) and describe how this problem can be solved in the presence of prediction and predictive-service. We will solve the SPS problem with a fixed λ value in Section III-B, based on which in Section III-C we can determine the optimal λ and achieve the optimal weighted timely-throughput.

B. The Single Packet Scheduling Problem

In this subsection, we consider the optimal solution to the SPS problem under a fixed λ. Recall that the state of a packet is described by a triple (r, τ, i). At each time-slot, the time-to-deadline τ is decremented by one. If a packet is still at the source when τ becomes 0, it will be discarded from the system. On the other hand, if a packet is delivered successfully before the deadline, we collect a reward β_n. The cost charged for resource expenditure in each transmission is λ per unit.

1) Perfect prediction: We start with perfect prediction (zero prediction corresponds to $D_n = 0$). In this case, the arrival of each packet from user n is known D_n timeslots in advance. Thus, we need to solve the SPS problem with an extended deadline of $\tau_n + D_n$. We can define $V_n(r, \tau, i)$ as the optimal value function for a packet of user n at state (r, τ, i). The value function and the optimal scheduling decision at each state can be obtained with the following Bellman equations.

$$V_n(0,\tau,i) = \max_e \Big\{ -\lambda e + \zeta_n(i,e) \sum_j P_n^{i,j} V_n(1, \tau-1, j) + (1-\zeta_n(i,e)) \sum_j P_n^{i,j} V_n(0, \tau-1, j) \Big\}, \quad 0 < \tau \le \tau_n + D_n, \tag{7}$$

$$V_n(0,0,i) = 0, \forall n, i, \tag{8}$$

$$V_n(1,\tau,i) = \beta_n, \forall n, i, 0 \le \tau < \tau_n + D_n. \tag{9}$$

2) Imperfect prediction: The imperfect prediction case is more complicated. We tackle this case by dividing the arrivals into two categories, i.e., the true-positive part and the false-negative part. The latter part contains unpredicted real arrivals. Since these arrivals are not expected, they can only be served after they enter the system. Thus, scheduling decisions for them remain the same as those in the zero prediction case. For predicted arrivals, the server can pre-serve them while they are still in the prediction window. However, there is a complication. If a predicted arrival is actually a false-alarm, we cannot collect any reward. Thus, the resource consumed to pre-serve the packet is wasted. Moreover, the correctness of a prediction can only be verified at the time when the predicted packet is supposed to arrive. Before that, the server will have to take chances and treat all predictions equally.

Based on the above reasoning, we will treat predicted packets and the mis-detections differently in the DP formulation. The optimal predictive-service can be done by the following augmented Bellman equations.

$$V_n(0,\tau,i) = \max_e \{-\lambda e + \zeta_n(i,e) \sum_j P_n^{i,j} V_n(1, \tau-1, j) + (1-\zeta_n(i,e)) \sum_j P_n^{i,j} V_n(0, \tau-1, j)\}, \quad 0 < \tau \le \tau_n + D_n, \tau \ne \tau_n + 1, \tag{10}$$

$$V_n(0,\tau_n+1,i) = \max_e \{-\lambda e + \zeta_n(i,e) \sum_j P_n^{i,j} V_n(1, \tau_n, j) + (1-\zeta_n(i,e)) p_n \sum_j P_n^{i,j} V_n(0, \tau_n, j)\}, \tag{11}$$

$$V_n(0,0,i) = 0, \forall n, i, \tag{12}$$

$$V_n(1,\tau,i) = \beta_n, \forall n, i, 0 \le \tau < \tau_n, \tag{13}$$

$$V_n(1,\tau,i) = p_n \beta_n, \forall n,, i, \tau_n \le \tau < \tau_n + D_n. \tag{14}$$

Here (10) and (13) are for unpredicted arrivals, and (11) and (14) are for predicted packets. Compared to (7), the main difference is that for predicted packets, one needs to take into account the fact that the system will collect a reward β_n from pre-serving a packet only with probability p_n.

C. The Optimal Weighted Timely-Throughput

After solving the SPS problem for a fixed λ, the policy $\pi^*(\lambda)$ that maximizes the Lagrangian (4) can be derived by letting each packet take its own optimal scheduling decision. Next we describe how to optimize the overall problem.

Denote $g(\lambda)$ the Lagrange dual function, i.e.,

$$g(\lambda) = \max_\pi L(\pi, \lambda). \tag{15}$$

Using Lemma 3 in [9], the optimal weighted timely-throughput ϕ^* equals the optimal value of the dual problem, i.e.,

$$\phi^* = \min_{\lambda \ge 0} g(\lambda). \tag{16}$$

This can be established by showing that the constrained MDP, with or without prediction, is equivalent to a linear program [27], [28]. Hence, the duality gap is zero.

We now look at the dual function in both cases.

1) Dual under zero or perfect prediction: Let $\boldsymbol{V}_n \triangleq (V_n(0, \tau_n + D_n, 1), V_n(0, \tau_n + D_n, 2), \dots, V_n(0, \tau_n + D_n, K))$ in the case with perfect prediction (zero prediction corresponds to $D_n = 0$). Then, using (5), we have:

$$g(\lambda) = \sum_{n=1}^{N} a_n \boldsymbol{\eta}_n^\intercal \boldsymbol{V}_n + \lambda B. \tag{17}$$

2) Dual under imperfect prediction: In this case, we note that the rate of predicted arrivals may not equal the actual arrival rate. Denote the predicted arrival rate from user n as $\tilde{a}_n$. Setting $a_{\max} = A_{\max}$, we have:

$$a_n = \tilde{a}_n p_n + (a_{\max} - \tilde{a}_n) q_n.$$

Here the second term is due to the fact that each packet is missed, i.e., not predicted, with probability q_n. Thus,

$$\tilde{a}_n = \frac{a_n - a_{\max} q_n}{p_n - q_n}. \tag{18}$$

Since $p_n > q_n$ and $0 \le \tilde{a}_n \le a_{\max}$, we get

$$q_n \le \frac{a_n}{a_{\max}} \le p_n. \tag{19}$$

Let $\boldsymbol{V}_n \triangleq (V_n(0, \tau_n, 1), V_n(0, \tau_n, 2), \dots, V_n(0, \tau_n, K))$ and $\tilde{\boldsymbol{V}}_n \triangleq (V_n(0, \tau_n + D_n, 1), V_n(0, \tau_n + D_n, 2), \dots, V_n(0, \tau_n + D_n, K))$. Similar to the perfect prediction case, the dual function can be expressed as:

$$g(\lambda) = \sum_{n=1}^{N} \Big[\tilde{a}_n \boldsymbol{\eta}_n^\intercal \tilde{\boldsymbol{V}}_n + (a_{\max} - \tilde{a}_n) q_n \boldsymbol{\eta}_n^\intercal \boldsymbol{V}_n \Big] + \lambda B, \tag{20}$$

where $\tilde{a}_n$ is determined in (18).

After obtaining $g(\lambda)$ for a fixed λ, we still need to find the optimal Lagrange multiplier $\lambda^* = \arg\min_{\lambda \ge 0} g(\lambda)$. One approach is to use the subgradient descent method, where we take an iterative procedure to converge to λ^* as follows. In the k-th iteration, we solve the SPS problem to get the optimal policy $\pi^*(\lambda_k)$ and the average resource expenditure $E_{av}(\pi^*(\lambda_k))$ based on the current multiplier λ_k. Then, the multiplier for the next iteration is given by (ϵ_k is a step size):

$$\lambda_{k+1} = \lambda_k + \epsilon_k[E_{av}(\pi^*(\lambda_k)) - B],$$

It is known that with an appropriately chosen $\{\epsilon_k\}_{k=1}^{\infty}$ sequence, $\lambda_k \to \lambda^*$ [29].

Despite the generality of (17) and (20), directly solving them is complicated. Thus, in next section, we first consider a slightly less general setting where user channels are static (different across users) and the resource expenditure option is binary.[3] Results for the general case are in Section V.

IV. The Static Scenario

In this scenario, we assume that the channel states are static, i.e., the success probability for transmitting a user n packet is a constant ζ_n. Moreover, we assume the resource level set is $\mathcal{E} = \{0, 1\}$, i.e., at each time-slot, the scheduling decision for each packet is to transmit it or not. In this case, the state of each packet can be described by (r, τ).

A. The Optimal Scheduling Policy

1) Perfect prediction: First we consider the perfect prediction case. We have the following theorem. Recall that the zero prediction case is a special case ($D_n = 0$ for all n).

Theorem 1. *For each user n packet, if $\zeta_n\beta_n > \lambda$, then the optimal policy is to transmit the packet at every time-slot, until it is either successfully delivered to the destination or becomes outdated. Moreover, the value function is given by:*

$$V_n(0, \tau) = \frac{1 - (1 - \zeta_n)^{\tau}}{\zeta_n}(-\lambda + \zeta_n\beta_n), 0 < \tau \leq \tau_n + D_n. \tag{21}$$

Otherwise, if $\zeta_n\beta_n \leq \lambda$, the value function is $V_n(0, \tau) = 0, 0 < \tau \leq \tau_n + D_n$. Specially, if $\zeta_n\beta_n < \lambda$, the optimal policy is to not transmit the packet at all.

Remark 1. *When $\lambda = \lambda^*$, based on the KKT conditions [29], it can be shown that for packets with $\zeta_n\beta_n = \lambda^*$, the optimal policy is to transmit the packet at every time-slot with probability $\bar{p}_P = \frac{B - \sum_{j:\zeta_j\beta_j > \lambda^*} a_j\bar{E}_j}{a_n\bar{E}_n}$, where $\bar{E}_n = \zeta_n + 2(1 - \zeta_n)\zeta_n + \cdots + (\tau_n + D_n)(1 - \zeta_n)^{\tau_n + D_n - 1}$, and not to transmit the packet otherwise. Notice that this has no influence on the value of the dual function $g(\lambda)$, as well as the optimal weighted timely-throughput obtained by $\phi^* = g(\lambda^*)$.*

Theorem 1 shows that if the expected reward is larger than the cost in one transmission, then the server should try its best to deliver packets for user n. We have the following corollary.

Corollary 1. *Let $g_P(\lambda)$ denote the the dual function of* (3) *and ϕ_P^* denote the optimal weighted timely-throughput in the perfect prediction case. We have:*

$$\begin{aligned} g_P(\lambda) &= \max_{\pi} L(\pi, \lambda) \\ &= \sum_{n:\zeta_n\beta_n > \lambda} a_n \frac{1 - (1 - \zeta_n)^{\tau_n + D_n}}{\zeta_n}(-\lambda + \zeta_n\beta_n) + \lambda B, \end{aligned}$$

$$\phi_P^* = \min_{\lambda} \sum_{n:\zeta_n\beta_n > \lambda} a_n \frac{1 - (1 - \zeta_n)^{\tau_n + D_n}}{\zeta_n}(-\lambda + \zeta_n\beta_n) + \lambda B. \tag{22}$$

Corollary 1 enables us to characterize the fundamental improvement in weighted timely-throughput due to prediction, which is shown in the next theorem. In the theorem, $g_0(\lambda)$ and ϕ_0^* denote the dual function and optimal weighted timely-throughput without prediction, respectively.

Theorem 2. *Suppose $\lambda_0^* = \arg\min_{\lambda} g_0(\lambda)$ and $\lambda_P^* = \arg\min_{\lambda} g_P(\lambda)$, then the weighted timely-throughput improvement satisfies:*

$$\begin{aligned} \phi_P^* - \phi_0^* \leq &\sum_{n:\zeta_n\beta_n > \lambda_0^*} \frac{a_n}{\zeta_n}\left[(1 - \zeta_n)^{\tau_n} - (1 - \zeta_n)^{\tau_n + D_n}\right] \\ &\times(-\lambda_0^* + \zeta_n\beta_n), \end{aligned} \tag{23}$$

$$\begin{aligned} \phi_P^* - \phi_0^* \geq &\sum_{n:\zeta_n\beta_n > \lambda_P^*} \frac{a_n}{\zeta_n}\left[(1 - \zeta_n)^{\tau_n} - (1 - \zeta_n)^{\tau_n + D_n}\right] \\ &\times(-\lambda_P^* + \zeta_n\beta_n). \end{aligned} \tag{24}$$

Theorem 2 shows that for user n, prediction improves the throughput by an amount of $\Theta(\rho^{\tau_n}(1 - \rho^{D_n}))$ with $\rho = 1 - \zeta_n$. This implies that the impact of prediction is decreasing with the deadline τ_n, which is expected, and the gap to the optimal improvement ρ^{τ_n} decreases *exponentially* as the prediction power D_n increases.

2) Imperfect prediction: In this case, we first note that there is not much the system can do with the unpredicted arrivals. Thus, the optimal scheduling policy and value function for these packets are the same as those in Theorem 1. For the predicted arrivals, on the other hand, the system will be able to start their services the moment they appear in the prediction window, following (10) to (14).

The following theorem characterizes the optimal predictive-service policy and the corresponding value functions. Recall that p_n is the true-positive probability and q_n is the false-negative probability.

Theorem 3. *Consider a predicted arrival for user n.*

(A) Suppose $\zeta_n\beta_n > \lambda$. Define

$$c_n \triangleq \frac{\lambda}{(-\lambda + \zeta_n\beta_n)(1 - \zeta_n)^{\tau_n} + \lambda}. \tag{25}$$

(i) If $p_n > c_n$, the optimal pre-service policy is to transmit the packet at every time-slot once it enters the prediction window, until it is either successfully delivered to the destination or revealed to be a false-alarm. Also,

$$\begin{aligned} V_n(0, \tau_n + w) = &-(1 - p_n)\frac{1 - (1 - \zeta_n)^{w}}{\zeta_n}\lambda \\ &+ p_n\frac{1 - (1 - \zeta_n)^{\tau_n + w}}{\zeta_n}(-\lambda + \zeta_n\beta_n), \end{aligned} \tag{26}$$

where $0 < w \leq D_n$. (ii) If $p_n \leq c_n$, the value function is:

$$V_n(0, \tau_n + w) = p_n\frac{1 - (1 - \zeta_n)^{\tau_n}}{\zeta_n}(-\lambda + \zeta_n\beta_n), \tag{27}$$

where $0 < w \leq D_n$. Specially, if $p_n < c_n$, the optimal policy is to not pre-serve the packet and to wait until it enters the system (if it is a true-positive).

(B) If $\zeta_n\beta_n \leq \lambda$, $V_n(0, \tau_n + w) = 0, 0 < w \leq D_n$. The optimal policy is to not transmit the packet at all if $\zeta_n\beta_n < \lambda$.

[3] Due to the complexity of the general stochastic scenario, similar static settings were also considered in the literature, e.g., [30] and [31].

Remark 2. *Similar with Theorem 1, when $\lambda = \lambda^*$, if there exist n_1, n_2 such that $p_{n_1} = c_{n_1}, \zeta_{n_2}\beta_{n_2} = \lambda^*$, then the optimal policy is to preserve n_1 packets with certain probability $\bar{p}_{I,1}$ and to transmit n_2 packets with certain probability $\bar{p}_{I,2}$, such that $E_{av} = B$.*

Theorem 3 shows that for a predicted user n arrival with $\zeta_n\beta_n > \lambda$, if the server waits until it enters the system, then does its best to deliver it, the expected reward is $p_n\frac{1-(1-\zeta_n)^{\tau_n}}{\zeta_n}(-\lambda + \zeta_n\beta_n)$. This is intuitive, as that the probability that a predicted arrival is real is p_n. Also note that although q_n does not appear in $V_n(0,\tau)$, we will see in Corollary 2 that it indirectly affects the final timely-throughput by affecting the effective arrival rate as in (18).

Note that c_n can intuitively be viewed as the weight put on resource consumption compared to reward. Hence, when $p_n > c_n$, the true-positive rate is large enough such that pre-transmitting the packet in one time slot, i.e., at time-slot $\tau_n + 1$, will increase the value function. Under this circumstance, Theorem 3 shows that the optimal scheduling is to transmit the packet as early as possible.

For a user n that satisfies $\zeta_n\beta_n > \lambda$, define

$$v_n(\lambda) = \begin{cases} p_n\frac{1-(1-\zeta_n)^{\tau_n}}{\zeta_n}(-\lambda+\zeta_n\beta_n),\ p_n \le c_n, \\ -(1-p_n)\frac{1-(1-\zeta_n)^{D_n}}{\zeta_n}\lambda \\ +p_n\frac{1-(1-\zeta_n)^{\tau_n+D_n}}{\zeta_n}(-\lambda+\zeta_n\beta_n),\ p_n > c_n. \end{cases}$$

We have the following immediate corollary from Theorem 3.

Corollary 2. *Let $g_I(\lambda)$ and ϕ_I^* denote the the dual function of* (3) *and the optimal weighted timely-throughput in the imperfect prediction case. We have:*

$$g_I(\lambda) = \lambda B + \sum_{n:\zeta_n\beta_n>\lambda}\Big[\tilde{a}_n v_n(\lambda)$$
$$+(a_{\max} - \tilde{a}_n)q_n\frac{1-(1-\zeta_n)^{\tau_n}}{\zeta_n}(-\lambda+\zeta_n\beta_n)\Big],$$
$$\phi_I^* = \min_\lambda g_I(\lambda), \tag{28}$$

where $\tilde{a}_n$ is the rate of predicted arrivals given in (18).

From Corollaries 1 and 2, we have the following theorem.

Theorem 4. *Suppose $\lambda_I^* = \arg\min g_I(\lambda)$. The weighted timely-throughput improvement satisfies*

$$\phi_I^* - \phi_0^* \le \sum_{n:\zeta_n\beta_n>\lambda_0^*}\Bigg\{\tilde{a}_n v_n(\lambda_0^*) \tag{29}$$
$$-\frac{(a_np_n - a_{\max}p_nq_n)[1-(1-\zeta_n)^{\tau_n}]}{(p_n-q_n)\zeta_n}(-\lambda_0^*+\zeta_n\beta_n)\Bigg\},$$
$$\phi_I^* - \phi_0^* \ge \sum_{n:\zeta_n\beta_n>\lambda_I^*}\Bigg\{\tilde{a}_n v_n(\lambda_I^*) \tag{30}$$
$$-\frac{(a_np_n - a_{\max}p_nq_n)[1-(1-\zeta_n)^{\tau_n}]}{(p_n-q_n)\zeta_n}(-\lambda_I^*+\zeta_n\beta_n)\Bigg\},$$

where $\tilde{a}_n$ is given in (18).

Remark 3. *Similar to Theorem 2, we still have in the imperfect prediction case that , the improvement scales in the order of $\Theta(p_n\left[C_1\frac{(a_n-a_{\max}q_n)}{p_n-q_n}\rho^{\tau_n} + C_2(1-\frac{1}{p_n})\right](1-\rho^{D_n}))$, which recovers the perfect prediction result when $p_n = 1$ and $q_n = 0$. This shows how different parameters affect the optimal timely-throughput, and provides guideline for prediction and control algorithm design.*

B. Influence of Prediction Accuracy

We now investigate how prediction accuracy impacts performance improvement. We have the following theorem.

Theorem 5. *Let $\boldsymbol{q} = \{q_1, q_2, \ldots, q_N\}$ denote the false-negative rate vector, and let $\boldsymbol{p} = \{p_1, p_2, \ldots, p_N\}$ be the true-positive rate vector. Then, (i) The optimal weighted timely-throughput ϕ_I^* is a non-increasing function of $\boldsymbol{q}$. (ii) If $\boldsymbol{q} = \boldsymbol{0}$, then the optimal weighted timely-throughput ϕ_I^* is a non-decreasing function of $\boldsymbol{p}$.*

The results, though intuitive, are non-trivial to established and require detailed investigation of the structure of the value functions. The impact of general $(\boldsymbol{p}, \boldsymbol{q})$ pairs is much more complicated to characterize (see [23] for simulation results).

V. The General Scenario

We now return to the general case. We first show that the optimal policy in the perfect prediction case (including zero prediction) is monotone with respect to the time-to-deadline.

Theorem 6. *Suppose $\zeta_n(i,e)$ is a concave strictly increasing function of e. In the perfect prediction case, define $e_n^*(0,\tau,i)$ as the optimal scheduling decision for a user n packet at state $(0,\tau,i)$. We have:*

$$e_n^*(0,\tau+1,i) \le e_n^*(0,\tau,i), \forall n, i, \tag{31}$$

where $0 < \tau < \tau_n + D_n$ ($D_n = 0$ denotes zero prediction).

Theorem 6 shows that the optimal scheduling policy is a "lazy" policy, i.e., the server will try to spend less resource at the beginning to see if the packet can luckily get through, and spend more resource when the deadline is getting close, so as to pursue the reward of successful delivery. From the proof of Theorem 6 (see [23]), we see that this is because the value function under the same channel state is monotonically non-decreasing with the time-to-deadline. This result nicely complements existing efficient scheduling results in the literature [32], [33], [34], and can be viewed as an extension to the predictive online scheduling setting.

It is tempting to conclude that similar property holds in the general imperfect prediction case. However, *this monotonicity actually does not hold under imperfect prediction.* This is because the value function is non-monotone due to prediction error. This fact will be illustrated by simulation in Section VI, where we see that the resource level can actually decrease with a smaller time-to-deadline (See User 2's strategy in Table III).

A. Throughput Improvement

In this subsection, we investigate the throughput improvement due to prediction in the general scenario.

1) Perfect prediction: First we consider the perfect prediction case. Define $i_n^{\max} \triangleq \arg\max_i \zeta_n(i,e), \forall e$ and $i_n^{\min} \triangleq \arg\min_i \zeta_n(i,e), \forall e$. Both $i_n^{\max}$ and $i_n^{\min}$ are well defined since there is a complete order on $\mathcal{S}$ based on $\zeta_n(i,e)$ (see Section II-B). We then have the following theorem. Note that the zero prediction case is a special case ($D_n = 0$ for all n).

Theorem 7. *For $1 \le n \le N, 0 < \tau \le \tau_n + D_n$, define*

$$V_n^l(\tau) \triangleq \max_{e>0}\frac{1-[1-\zeta_n(i_n^{\min},e)]^\tau}{\zeta_n(i_n^{\min},e)}[-\lambda e + \zeta_n(i_n^{\min},e)\beta_n], \tag{32}$$

where $V_n^l(0) = 0$, and

$$V_n^u(\tau) \triangleq \sum_{z=1}^{\tau} \max_e\{-\lambda e + \zeta_n(i_n^{\max}, e)(\beta_n - \max\{0, V_n^l(z-1)\})\}, \tag{33}$$

where $V_n^u(0) = 0$. For each user n packet, given a fixed λ, the value function for $0 < \tau \leq \tau_n + D_n$ satisfies:

$$\max\{0, V_n^l(\tau)\} \leq V_n(0, \tau, i) \leq \min\{\beta_n, V_n^u(\tau)\}. \tag{34}$$

We then immediately have the following corollary.

Corollary 3. *Define*

$$g_P^u(\lambda) \triangleq \lambda B + \sum_{n=1}^{N} a_n \min\{\beta_n, V_n^u(\tau_n + D_n)\}, \tag{35}$$

$$g_P^l(\lambda) \triangleq \lambda B + \sum_{n=1}^{N} a_n \max\{0, V_n^l(\tau_n + D_n)\}, \tag{36}$$

where $V_n^l(\tau)$ and $V_n^u(\tau)$ are defined in (32) *and* (33). *Let $g_P(\lambda)$ and ϕ_P^* be the dual function and the optimal timely-throughput with perfect prediction in the general case, respectively. We have:*

$$g_P^l(\lambda) \leq g_P(\lambda) \leq g_P^u(\lambda), \forall \lambda \geq 0, \tag{37}$$

$$\min_\lambda g_P^l(\lambda) \leq \phi_P^* \leq \min_\lambda g_P^u(\lambda). \tag{38}$$

We similarly define $g_0^u(\lambda)$ and $g_0^l(\lambda)$ for the zero prediction case. Then, the following theorem characterizes the improvement in weighted timely-throughput from prediction.

Theorem 8. *Let $\lambda_0^* = \arg\min_\lambda g_0^l(\lambda)$ and $\lambda_P^* = \arg\min_\lambda g_P^l(\lambda)$. The weighted timely-throughput improvement satisfies:*

$$g_P^l(\lambda_P^*) - g_0^u(\lambda_P^*) \leq \phi_P^* - \phi_0^* \leq g_P^u(\lambda_0^*) - g_0^l(\lambda_0^*). \tag{39}$$

Theorem 8 is the counterpart of Theorem 2 in the general case, with the main difference that due to the general Markov dynamics, the bounds can be loose compared to those in Theorem 2.

2) Imperfect prediction: We now turn to the imperfect prediction case. Recall that p_n is the true-positive probability and q_n is the false-negative probability.

Theorem 9. *For $1 \leq n \leq N$, $0 < \tau < \tau_n$, let $\tilde{V}_n^l(\tau) = V_n^l(\tau)$ in* (32), *and $\tilde{V}_n^u(\tau) = V_n^u(\tau)$ in* (33). *Also define*

$$\tilde{V}_n^l(\tau) = \max_{e>0}\left\{ - (1-p_n)\frac{1-[1-\zeta_n(i_n^{\min}, e)]^{\tau-\tau_n}}{\zeta_n(i_n^{\min}, e)}\lambda e + p_n\frac{1-[1-\zeta_n(i_n^{\min}, e)]^{\tau}}{\zeta_n(i_n^{\min}, e)}[-\lambda e + \zeta_n(i_n^{\min}, e)\beta_n]\right\}, \tag{40}$$

$$\begin{aligned}\tilde{V}_n^u(\tau) = &\sum_{z=\tau_n+1}^{\tau} \max_e\{-\lambda e + \zeta_n(i_n^{\max}, e) \\ &\times (p_n\beta_n - \max\{0, \tilde{V}_n^l(\tau_n), \tilde{V}_n^l(z-1)\})\} \\ &+ p_n \sum_{z=1}^{\tau_n} \max_e\{-\lambda e + \zeta_n(i_n^{\max}, e) \\ &\times (\beta_n - \max\{0, \tilde{V}_n^l(z-1)\})\},\end{aligned} \tag{41}$$

for $\tau_n \leq \tau \leq \tau_n + D_n$. Notice that $\tilde{V}_n^l(\tau_n) = p_n V_n^l(\tau_n)$ and $\tilde{V}_n^u(\tau_n) = p_n V_n^u(\tau_n)$. Consider a predicted user n arrival, given a fixed λ, the value function satisfies ($0 < w \leq D_n$):

$$V_n(0, \tau + w, i) \geq \max\{0, \tilde{V}_n^l(\tau_n), \tilde{V}_n^l(\tau_n + w)\}, \tag{42}$$

$$V_n(0, \tau + w, i) \leq \min\{p_n\beta_n, \tilde{V}_n^u(\tau_n + w)\}. \tag{43}$$

Interestingly, we see in the proof (see [23]) that the three terms in $\max\{0, \tilde{V}_n^l(\tau_n), \tilde{V}_n^l(\tau_n + w)\}$ in (42) correspond to not transmitting, transmitting after packet arrival, and proactive transmission. From Theorem 9, we have the following corollary.

Corollary 4. *Define*

$$g_I^u(\lambda) \triangleq \lambda B + \textstyle\sum_{n=1}^{N} \left\{\tilde{a}_n \min[p_n\beta_n, \tilde{V}_n^u(\tau_n + D_n)] + (a_{\max} - \tilde{a}_n)q_n \min[\beta_n, V_n^u(\tau_n)]\right\}, \tag{44}$$

$$g_I^l(\lambda) \triangleq \lambda B + \textstyle\sum_{n=1}^{N} \left\{\tilde{a}_n \max[0, \tilde{V}_n^l(\tau_n), \tilde{V}_n^l(\tau_n + D_n)] + (a_{\max} - \tilde{a}_n)q_n \max[0, V_n^l(\tau_n)]\right\}. \tag{45}$$

Let $g_I(\lambda)$ and ϕ_I^* be the dual function and the optimal timely-throughput in the imperfect prediction case, respectively. Then,

$$g_I^l(\lambda) \leq g_I(\lambda) \leq g_I^u(\lambda), \forall \lambda \geq 0, \tag{46}$$

$$\min_\lambda g_I^l(\lambda) \leq \phi_I^* \leq \min_\lambda g_I^u(\lambda). \tag{47}$$

From the above, we can now bound the improvement in weighted timely-throughput with imperfect prediction.

Theorem 10. *Suppose $\lambda_I^* = \arg\min_\lambda g_I^l(\lambda)$, then the weighted timely-throughput improvement satisfies:*

$$g_I^l(\lambda_I^*) - g_0^u(\lambda_I^*) \leq \phi_I^* - \phi_0^* \leq g_I^u(\lambda_0^*) - g_0^l(\lambda_0^*). \tag{48}$$

Although it is hard to exactly characterize the timely-throughput improvement and how it depends on p_n and q_n in this general case (due to the general system dynamics), Theorems 8 and 10 provide useful upper and lower bounds. Simulation results show that performance in the general case is similar to that in the static case.

VI. Simulation Results

We present simulation results in this section. We consider for the system in Fig. 1 with $N = 4$ users. The arrival rates are given by $(a_1, \ldots, a_4) = (0.7, 0.6, 0.4, 0.3)$ with $a_{\max} = 1$ and the deadlines are given by $(\tau_1, \ldots, \tau_4) = (2, 3, 4, 5)$. The reward weight vector is $\boldsymbol{\beta} = (3, 1, 2, 4)$. Each channel has $K = 4$ states $(s_1, \ldots, s_4) = (1, 2, 3, 4)$, each representing a noise level. The channel state transition matrix is the same for all users and is given by: $P = (0.4, 0.3, 0.2, 0.1; 0.25, 0.3, 0.25, 0.2; 0.2, 0.25, 0.3, 0.25; 0.1, 0.2, 0.3, 0.4)$. The resource level set is $\mathcal{E} = \{0.0001z, z = 0, 1, ..., 6 \times 10^4\}$. The average resource expenditure budget is $B = 6$. The probability that a transmission for user n is successful under state s_i and resource level e is given by: $\zeta_n(i, e) = \frac{2}{1+e^{-2\frac{e}{d_n^3 s_i}}} - 1$, where $\boldsymbol{d} = (d_1, \ldots, d_4) = (1.1, 1.2, 1.3, 1.4)$ and d_n denotes the distance between user n and the server. This setting models a wireless downlink system. The prediction window sizes are the same for all users, i.e., $D_n = 2, n = 1, \ldots, 4$. For the imperfect prediction case, the true-positive rates and false-negative rates are $\boldsymbol{p} = (0.8, 0.8, 0.8, 0.8)$ and $\boldsymbol{q} = (0.2, 0.1, 0.1, 0.2)$.

1) Optimal Policy: We start with the optimal actions for different users under different states. Table I and Table II show the optimal scheduling decisions for User 2 ($\tau_2 = 3$) and User 4 ($\tau_4 = 5$), i.e., $e_n^*(0, \tau, i)$, in the zero prediction and perfect prediction cases. We can verify the monotonicity of the optimal policy, as shown in Theorem 6. We also see that the resource offered for User 4 is larger than that under the same state for User 2. This is intuitive since $\beta_4 > \beta_2$.

(a) User 2

τ S(t)	3	2	1
s_1	1.3915	1.4674	1.6328
s_2	0.0	0.2762	1.0554
s_3	0.0	0.0	0.0
s_4	0.0	0.0	0.0

(b) User 4

τ S(t)	5	4	3	2	1
s_1	2.6024	2.7488	2.9284	3.2269	4.1129
s_2	2.3917	2.9635	3.5677	4.3658	6.0
s_3	0.0	0.0	1.6809	3.9651	6.0
s_4	0.0	0.0	0.0	1.9522	6.0

TABLE I
OPTIMAL SCHEDULING DECISIONS $e_n^*(0, \tau, i)$ (ZERO PREDICTION)

(a) User 2

τ S(t)	3+2	3+1	3	2	1
s_1	1.3651	1.4248	1.4906	1.5797	1.7865
s_2	0.0	0.0	0.6726	1.1571	1.6759
s_3	0.0	0.0	0.0	0.0	0.0
s_4	0.0	0.0	0.0	0.0	0.0

(b) User 4

τ S(t)	5+2	5+1	5	4	3	2	1
s_1	2.3912	2.5051	2.6355	2.7873	2.9807	3.3168	4.3156
s_2	1.2882	1.948	2.5292	3.1024	3.7309	4.6111	6.0
s_3	0.0	0.0	0.0	0.0	2.315	4.5133	6.0
s_4	0.0	0.0	0.0	0.0	0.0	3.4852	6.0

TABLE II
OPTIMAL SCHEDULING DECISIONS $e_n^*(0, \tau, i)$ (PERFECT PREDICTION)

Table III shows the optimal policy with imperfect prediction. From User 2's scheduling decisions under channel state s_1, we see that *the monotonicity property actually does not hold when there is prediction error*, i.e., resource expenditure for state s_1 and $\tau = 3{+}1$ is smaller than that for state s_1 and $\tau = 3{+}2$. We also note that resource allocated under imperfect prediction is less than that with perfect prediction. This is because resource may be wasted when prediction is imperfect. Thus, the server will be more conservative in resource allocation.

(a) User 2

τ S(t)	3+2	3+1	3	2	1
s_1	**0.8382**	**0.8269**	1.2468	1.313	1.4435
s_2	0.0	0.0	0.0	0.0	0.0
s_3	0.0	0.0	0.0	0.0	0.0
s_4	0.0	0.0	0.0	0.0	0.0

(b) User 4

τ S(t)	5+2	5+1	5	4	3	2	1
s_1	1.9575	1.9758	2.5562	2.6954	2.8624	3.1235	3.8744
s_2	0.0	0.0	2.1895	2.7631	3.3489	4.0761	5.4601
s_3	0.0	0.0	0.0	0.0	0.0	3.2527	5.7612
s_4	0.0	0.0	0.0	0.0	0.0	0.0	4.5436

TABLE III
OPTIMAL SCHEDULING DECISIONS $e_n^*(0, \tau, i)$ (IMPERFECT PREDICTION)

2) Influence of Parameters: Next we investigate the influence of system parameters. Fig. 3 shows how the weighted timely-throughput changes when all users have the same deadline going from 2 to 7 ($D_n = 2$). We see that the timely-throughputs in both cases, with or without prediction, increase with the deadline, and the throughput improvement decreases as τ_n becomes larger. This is expected, as a large deadline gives more flexibility to scheduling. Thus, the marginal benefit of prediction decreases. Moreover, we see that the throughput with prediction is always higher, demonstrating the effectiveness of prediction.

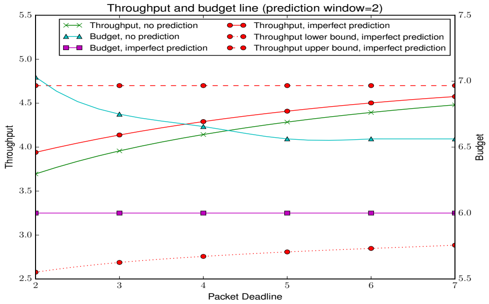

Fig. 3. Timely-throughput increases in packet deadline. The right y-axis (blue and purple curves) also shows the resource budget needed in the zero prediction case to achieve the same throughput as in the imperfect prediction case. With prediction, the overall resource consumption is set as 6. However, without prediction, we need significantly higher resource consumption rates to achieve the same throughputs.

Fig. 4 shows how the optimal weighted throughput changes with prediction power. From the results, we see that although prediction is imperfect, if used properly, it can still significantly improve timely-throughput. The throughput bounds in Fig. 3 and Fig. 4 appear loose due to the complicated Markov dynamics. Fig. 5, on the other hand, shows that the bounds are tight when there is a single channel state.

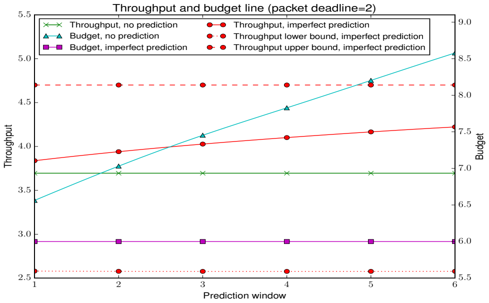

Fig. 4. Timely-throughput changes with the prediction window. The right y-axis (blue and purple curves) also shows the budget needed for achieving the same throughputs as with prediction. Without prediction, one needs significantly higher resource consumption rates to achieve the same throughputs.

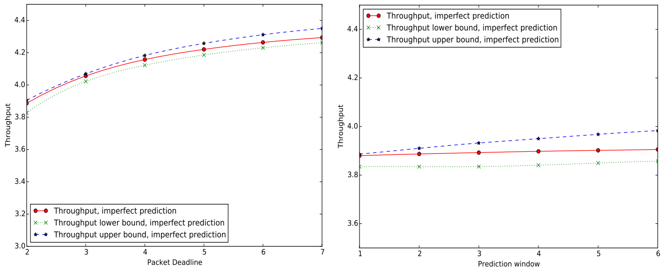

Fig. 5. Timely-throughput and the corresponding upper and lower bounds. The set of channel states is $\mathcal{S} = \{s_2\}$. We see that the bounds are tight in this case.

We show in Fig. 6 results with fixed $\boldsymbol{q}$ and $\boldsymbol{p}$, respectively, to show how timely throughput changes with the other. In the left plot, the false-negative rate vector is set to $\boldsymbol{q} = \boldsymbol{0}$, and all users have the same true-positive rate increasing from 0.75 to 0.95. In the right plot, the true-positive rate vector is set as $\boldsymbol{p} = (0.8, 0.8, 0.8, 0.8)$ and all users have the same false-negative rate changing from 0.05 to 0.25. Other settings are kept unchanged. From the results, we see that the timely-throughput is increasing in the true-positive rate and decreasing in the false-negative rate for this stochastic case (this is proven for the static scenario in Theorem 5).

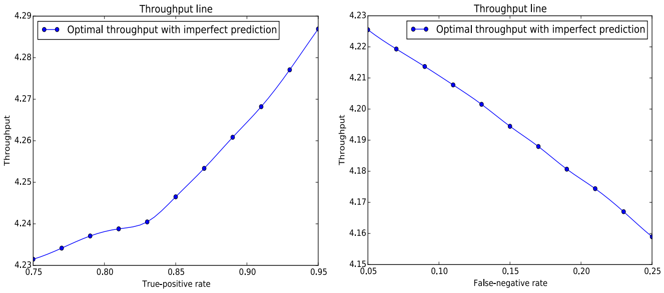

Fig. 6. Timely-throughput change with p and q. Left plot: $\boldsymbol{q} = \mathbf{0}$ and p increases from 0.75 to 0.95. Right plot: $\boldsymbol{p} = (0.8, 0.8, 0.8, 0.8)$ and q increases from 0.05 to 0.25.

VII. Conclusion

We investigate the fundamental benefit of predictive scheduling in improving timely-throughput in a general stochastic single-server multi-user system. We first derive a Markov decision process solution for optimizing the timely-throughput, subject to an average resource consumption constraint. We then explicitly characterize the optimal policy and quantify the timely-throughput improvement due to predictive-service. Extensive simulations are conducted to validate our theoretical results. Our results provide novel insights into how prediction and system parameters impact performance and provide guidelines for designing predictive control algorithms.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China Grants 61672316, 61303195, the Tsinghua Initiative Research Grant, and the China Youth 1000-talent grant.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 20162021," *White Paper*, accessed July 26 2017.

[2] E. Schurman and J. Brutlag, "The user and business impact of server delays, additional bytes and http chunking in web search," *OReilly Velocity Web Performance and Operations Conference*, June 23, 2009.

[3] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory, Vol. 39, No. 2, pp. 466-478*, March1993.

[4] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Nonlinear Optimization of Communication Systems*, vol. 24, no. 8, pp. 1489–1501, Aug. 2006.

[5] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," *Proceedings of IEEE INFOCOM Mini-Conference*, April 2009.

[6] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," *Proceedings of IEEE INFOCOM*, April 2009.

[7] L. Huang and M. J. Neely, "Delay reduction via Lagrange multipliers in stochastic network optimization," *IEEE Trans. on Automatic Control*, vol. 56, no. 4, pp. 842–857, April 2011.

[8] I. Hou and P. Kumar, "Utility-optimal scheduling in time-varying wireless networks with delay constraints," *Proceedings of ACM MobiHoc*, 2010.

[9] R. Singh and P. Kumar, "Throughput optimal decentralized scheduling of multi-hop networks with end-to-end deadline constraints: Unreliable links," *arXiv preprint arXiv:1606.01608*, 2016.

[10] G. Bensinger, "Amazon wants to ship your package before you buy it," *The Wall Street Journal*, Jan 17 2014.

[11] J. Broughton, "Netflix adds download functionality," *IHS Markit*, Dec 2, 2016.

[12] K. M. U. Farooq and L. K. John, "Store-load-branch (slb) predictor: A compiler assisted branch prediction for data dependent branches," *Proceedings of the 19th IEEE International Symposium on High-Performance Computer Architecture*, 2013.

[13] L. Huang, S. Zhang, M. Chen, and X. Liu, "When backpressure meets predictive scheduling," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2237–2250, 2016.

[14] H. Yu, M. Cheung, L.Huang, and J. Huang, "Power-delay tradeoff with predictive scheduling in integrated cellular and wi-fi networks," *IEEE JSAC special issue on Energy-Efficient Techniques for 5G Wireless Communication Systems, vol. 34, issue 4, pp. 735-742*, 2016.

[15] S. Zhang, L. Huang, M. Chen, and X. Liu, "Proactive serving decreases user delay exponentially: The light-tailed service time case," *IEEE/ACM Transactions on Networking (TON), vol. 25, issue 2, 708-723*, April 2017.

[16] L. Huang, M. Chen, and Y. Liu, "Learning-aided stochastic network optimization with imperfect state prediction," *Proceedings of ACM MobiHoc*, 2017.

[17] J. Tadrous and A. Eryilmaz, "On optimal proactive caching for mobile networks with demand uncertainties," *IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2715-2727*, Oct 2016.

[18] J. Tadrous, A. Eryilmaz, and H. E. Gamal, "Proactive data download and user demand shaping for data networks," *IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 1917-1930*, December 2015.

[19] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. H. Andrew, "Online convex optimization using predictions," *Proceedings of ACM Sigmetrics*, 2015.

[20] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on the past," *Proceedings of ACM Sigmetrics*, 2016.

[21] M. Hajiesmaili, C. Chau, M. Chen, and L. Huang, "Online microgrid energy generation scheduling revisited: The benefits of randomization and interval prediction," *Proceedings of ACM e-Energy*, 2016.

[22] I. Hou and P. Kumar, "Scheduling periodic real-time tasks with heterogeneous reward requirements," *Proceedings of RTSS*, 2011.

[23] K. Chen and L. Huang, "Timely-throughput optimal scheduling with prediction," *arXiv preprint arXiv:1712.05677*, 2017.

[24] H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena, "Intelligent workload factoring for a hybrid cloud computing model," in *Proceedings of the 2009 World Conference on Services*. IEEE, 2009, pp. 701–708.

[25] Y. Aviv, "The effect of collaborative forecasting on supply chain performance," *Management science*, vol. 47, no. 10, pp. 1326–1343, 2001.

[26] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vols. I and II*. Boston: Athena Scientific, 2005 and 2007.

[27] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[28] ——, "Constrained markov decision processes with total cost criteria: Occupation measures and primal lp," *Mathematical methods of operations research*, vol. 43, no. 1, pp. 45–72, 1996.

[29] D. P. Bertsekas, A. Nedi, A. E. Ozdaglar *et al.*, *Convex analysis and optimization*. Athena Scientific, 2003.

[30] S. Chen, P. Sinha, N. B. Shroff, and C. Joo, "A simple asymptotically optimal joint energy allocation and routing scheme in rechargeable sensor networks," *IEEE/ACM Trans. on Networking, vol. 22, no. 4, pp. 1325-1336*, Aug. 2014.

[31] H. Yu and M. J. Neely, "A new backpressure algorithm for joint rate control and routing with vanishing utility optimality gaps and finite queue lengths," *Proceedings of IEEE INFOCOM*, 2017.

[32] B. Prabhakar, E. U. Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," 2001.

[33] M. Zafer and E. Modiano, "Minimum energy transmission over a wireless channel with deadline and power constraints," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2841–2852, 2009.

[34] M. A. Zafer and E. Modiano, "A calculus approach to energy-efficient data transmission with quality-of-service constraints," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 898–911, 2009.