Timely-Throughput Optimal Scheduling With Prediction
Kun Chen and Longbo Huang

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 paper, 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 \( \Theta[(p/C_1(a-a_{\max}q))^{\rho^*/(p-q)} + C_2(1/(1/p))[1-(p^*)^q]] \), where \( a, a_{\max}, \rho \in (0, 1), C_1 > 0, C_2 > 0 \) are constants, \( p \) is the true-positive rate in prediction, \( q \) is the false-negative rate, \( \tau \) 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.

Index Terms—Network control, delay-guaranteed service, predictive scheduling, timely-throughput.

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]–[11].

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 practical systems to predict user demand and to proactively serve customer requests. For example, Amazon tries to predict what customers may purchase and pre-ships products to distribution centers close to them, in order to reduce shipping time [12]. Netflix, on the other hand, tries to predict what customers may want and preload videos onto user devices to improve quality-of-experience [13]. 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 [14]. 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. [15] and [16] consider utility optimal scheduling in downlink systems based on perfect user prediction. Reference [17] shows that proactive scheduling can effectively reduce queueing delay in stochastic single-queue systems. Reference [18] considers how network state prediction can be incorporated into algorithm design. Reference [19] and [20] focus on understanding the cost saving aspect of proactive scheduling based on demand prediction. Reference [21], [22], and [23] also investigate the benefit of prediction from an online algorithm design perspective. However, we note that the
Our formulation is closest to recent works [8]–[11], [15], [24], 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 [24] focuses on periodic traffic, while our work allows heterogeneous deadlines for user packets and random arrivals. Reference [15] focuses on optimizing system utility subject to stability constraint, while we work explicitly with delay constraints. Lastly, our work builds upon the novel results in the series of works [9]–[11], where [9] introduce the packet-level decomposition approach, and [10], [11] use this approach to study unreliable transmissions or wireless networks with interference in causal systems. Our work extend their method to deal with proactive service in a Markov system. The extension to incorporate prediction significantly complicates both the solution and analysis. Moreover, previous works focus on the algorithm aspect, i.e., to find the optimal scheduling algorithm, while we focus on characterizing properties of the optimal scheduling policy and quantifying the impact of prediction and proactive service, which requires detailed investigation of the system. Our results offer novel insights into the fundamental benefits of prediction in delay-constrained network control.

The main contributions of our paper 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 $\Theta(p \left[ C_1 \frac{a-a_{\max}}{p-q} \rho^\tau + C_2 (1 - \frac{1}{p}) \right] (1 - \rho^D))$, 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, $\tau$ is the packet deadline and $D$ is the prediction window size. This concise and explicit characterization is novel and 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.
We assume without loss of generality that there is a total order on set $S$ 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 (the setting with hard capacity constraint will be considered in Section V). This assumption is made to facilitate analysis and was also adopted in [10].

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 $D_{n}(t) = \{A_{n}(t + 1), \ldots, A_{n}(t + D_{n})\}$ for each user. Moreover, the system implements predictive service, i.e., it tries to pre-serve future arrivals in $D_{n}(t)$ 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 [12].

In this work, we focus on two prediction models.

1) Perfect Prediction: In this case, the predicted arrivals in $D_{n}(t)$ 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., [15], [16], and [25].

2) Imperfect Prediction: In the second model, 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 $p_{n} > q_{n}$.

Note that the perfect prediction case corresponds to $p_{n} = 1$ and $q_{n} = 0$. This model was previously adopted in [15], [17], and [26] for studying the delay reduction due to prediction.

D. System Objective

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

$$x_{n} = \lim_{T \to \infty} \frac{1}{T} E \left\{ \sum_{t=1}^{T} X_{n}(t) \right\},$$

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:

$$E_{av} = \lim_{T \to \infty} \frac{1}{T} E \left\{ \sum_{t=1}^{T} \sum_{n=1}^{N} E_{n}(t) \right\},$$

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

Denote $x = (x_{1}, \ldots, x_{N})$. Given a weight vector $\beta = (\beta_{1}, \ldots, \beta_{N})$ with $\beta_{n} \geq 0, \forall 1 \leq n \leq N$, the weighted timely-throughput $\phi$ is defined as $\phi \triangleq \beta^{\top} x$. In this paper, we focus on the problem of maximizing the weighted timely-throughput, subject to an average resource constraint $B$, i.e.,

$$\phi^{*} \triangleq \max \beta^{\top} x,$$

s.t. $E_{av} \leq B$.

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

E. Model Discussion

Our work builds upon the novel results in recent work [10]. However, our model and results are different as follows. (i) We consider a Markov model for the system while [10] 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., [10] and [8], mostly consider causal systems. Analysis for predictive systems is significantly complicated by potential errors in prediction and requires different arguments compared to those for causal systems.

Understanding how prediction impacts delay-constrained services is critical for future intelligent communication and computing systems, as providing delay-guaranteed services has long been an important problem in various applications and predictive scheduling has been successfully utilized in different delay-sensitive scenarios, such as video streaming [27] and supply chain optimization [28].

III. SCHEDULING BY PACKET-LEVEL DECOMPOSITION

Problem (3) can be formulated as a constrained Markov Decision Process (MDP). However, it is known that 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 [10] 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 $(r, \tau, i)$. 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. $\tau$ is the time duration before reaching 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. Note that the number of arrivals at any time-slot from any user is bounded by $A_{\text{max}}$, each packet can stay in the system for no more than $\Gamma$ slots, and the number of channel states is finite, so the number of system states is finite (though can be exponentially large).

A (possibly randomized) scheduling policy $\pi$ 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 [29].

A. Packet-Level Decomposition for the Constrained MDP

Let $\lambda$ be the Lagrange multiplier for constraint (4). The Lagrangian of (3) can be written as

$$L(\pi, \lambda) = \lim_{T \to \infty} \frac{1}{T} \mathbb{E} \left\{ \sum_{n=1}^{N} \sum_{t=1}^{T} \beta_n X_n(t) \right\} - \lambda \mathbb{E} \left\{ \sum_{t=1}^{T} \sum_{n=1}^{N} E_n(t) \right\} + \lambda B. \quad (5)$$

Here $\sum_{n=1}^{N} \sum_{t=1}^{T} \beta_n X_n(t)$ 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 $A_n(T)$ the set of packet arrivals for user $n$ up to time $T$, the Lagrangian (5) can be decomposed into the following packet-level form:

$$L(\pi, \lambda) = \lim_{T \to \infty} \frac{1}{T} \mathbb{E} \left\{ \sum_{n=1}^{N} \sum_{\xi \in A_n(T)} \left[ \beta_n \delta(\xi) - \lambda E(\xi) \right] \right\} + \lambda B, \quad (6)$$

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

From (6), the term related to packet $\xi$ of user $n$ is

$$\mathbb{E} \left\{ \beta_n \delta(\xi) - \lambda E(\xi) \right\}. \quad (7)$$

As a result, maximizing the Lagrangian (5) can be accomplished by maximizing (7) for each packet. In the following, we refer to problem (7) as the Single Packet Scheduling Problem (SPS) and describe how this problem can be solved in the presence of prediction and predictive-service.

We will carry out our analysis of the SPS problem with a fixed $\lambda$ value in Section III-B, based on which in Section III-C we can determine the optimal $\lambda$ 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 multiplier $\lambda$. Recall that the state of a packet is described by a triple $(r, \tau, i)$. At each time-slot, the time-to-deadline $\tau$ is decremented by one. If a packet is still at the source when $\tau$ 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 $\beta_n$. The cost charged for resource expenditure in each transmission is $\lambda$ per unit.

1) Perfect Prediction: We start with the perfect prediction case (note that zero prediction corresponds to having $D_n = 0$). 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 $V_n(\tau, i)$ as the optimal value function for a packet of user $n$ at state $(\tau, i)$. The value function and the optimal scheduling decision at each state can be obtained with the following Bellman equations:

$$V_n(0, \tau, i) = \max_{e} \left\{ -\lambda e + \zeta_n(i, e) \sum_{j} P_{n}^{i,j} V_n(1, \tau - 1, j) \right\},$$

$$0 < \tau \leq \tau_n + D_n, \quad (8)$$

$$V_n(0, 0, i) = 0, \quad \forall n, i, \quad (9)$$

$$V_n(1, \tau, i) = \beta_n, \quad \forall n, i, \quad 0 \leq \tau < \tau_n + D_n. \quad (10)$$

2) Imperfect Prediction: The imperfect prediction case is more complicated. We tackle this case by dividing the arrivals into two categories, namely, the true-positive part and the false-negative part. The latter part contains the unpredicted true arrivals. Since these arrivals are not expected, they can only be served after they enter the system. Thus, scheduling decisions for these packets are 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 in this case. If a predicted arrival is actually a false-alarm, we cannot collect any reward. As a result, 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.

$$V_n(0, \tau, i) = \max_{e} \left\{ -\lambda e + \zeta_n(i, e) \sum_{j} P_{n}^{i,j} V_n(1, \tau - 1, j) \right\}$$

$$+ (1 - \zeta_n(i, e)) \sum_{j} P_{n}^{i,j} V_n(0, \tau - 1, j),$$

$$0 < \tau \leq \tau_n + D_n, \quad \tau \neq \tau_n + 1. \quad (11)$$
Using Lemma 3 in [10], the optimal weighted timely-throughput is given by
\[ V_n(0, \tau_n + 1, i) = \max_{\pi} \{-\lambda e^{-\lambda(\tau_n + 1)} + \eta_n(i, e) \sum_j P_n^{i,j} V_n(1, \tau_n, j) + (1 - \eta_n(i, e))p_n \sum_j P_n^{i,j} V_n(0, \tau_n, j)\}, \] (12)
where \[ V_n(0, 0, i) = 0, \quad \forall n, i, \] (13)
\[ V_n(1, \tau, i) = \beta_n, \quad \forall n, i, \quad 0 \leq \tau < \tau_n, \] (14)
\[ V_n(1, \tau, i) = p_n\beta_n, \quad \forall n, i, \quad \tau_n \leq \tau < \tau_n + D_n. \] (15)
Here (11) and (14) are for unpredicted arrivals, and (12) and (15) are for predicted packets. Compared to (8), the main difference is that for predicted packets, one needs to take into account the fact that the system will collect a reward \( \beta_n \) from pre-serving a packet only with probability \( p_n \). Also note that the time \( \tau_n + 1 \) is special, as in the next time slot we will be able to verify the correctness of a predicted arrival. Hence, it is separately treated in (12).

C. The Optimal Weighted Timely-Throughput

After solving the SPS problem for a fixed \( \lambda \), the policy \( \pi^*(\lambda) \) that maximizes the Lagrangian (5) 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.,
\[ g(\lambda) = \max_{\pi} L(\pi, \lambda). \] (16)
Using Lemma 3 in [10], the optimal weighted timely-throughput \( \phi^* \) equals the optimal value of the dual problem, i.e.,
\[ \phi^* = \min_{\lambda \geq 0} g(\lambda). \] (17)
This can be established by showing that the constrained MDP, with or without prediction, is equivalent to a linear program [30], [31]. Hence, the duality gap is zero.

In the following, we look at the Lagrange dual function in the perfect and imperfect prediction cases.

1) Dual Under Zero or Perfect Prediction: Let
\[ V_n \triangleq (V_n(0, \tau_n + D_n, 1), V_n(0, \tau_n + D_n, 2), \ldots, V_n(0, \tau_n + D_n, K)) \]
in the case with perfect prediction (zero prediction corresponds to \( D_n = 0 \)). Then, using (6), the dual function can be expressed as (recall that \( \eta_n \) is the steady-state distribution of the channel state for user \( n \)):
\[ g(\lambda) = \sum_{n=1}^{N} a_n \eta_n^\top V_n + \lambda B. \] (18)

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 \( \tilde{a}_n \). We have:
\[ a_n = \tilde{a}_n p_n + (a_{\text{max}} - \tilde{a}_n)q_n. \]
Here the second term is due to the fact that each packet is missed, i.e., not predicted, with probability \( q_n \). Thus,
\[ \tilde{a}_n = \frac{a_n - a_{\text{max}}q_n}{p_n - q_n}. \] (19)
Since \( p_n > q_n \) and \( 0 \leq \tilde{a}_n \leq a_{\text{max}} \), we get
\[ q_n \leq \tilde{a}_n \leq a_{\text{max}}. \] (20)

Let \( V_n \triangleq (V_n(0, \tau_n, 1), V_n(0, \tau_n, 2), \ldots, V_n(0, \tau_n, K)) \) and \( D_n \triangleq (D_n(0, \tau_n + D_n, 1), D_n(0, \tau_n + D_n, 2), \ldots, D_n(0, \tau_n + D_n, K)) \). Similar to the perfect prediction case, the dual function can be expressed as:
\[ g(\lambda) = \sum_{n=1}^{N} \tilde{a}_n \eta_n^\top V_n + (a_{\text{max}} - \tilde{a}_n)q_n \eta_n^\top V_n + \lambda B, \] (21)
where \( \tilde{a}_n \) is determined in (19).

After obtaining the dual function for a fixed \( \lambda \), we still need to find the optimal Lagrange multiplier \( \lambda^* = \arg \min_{\lambda \geq 0} g(\lambda) \). One possible approach is to use the subgradient descent method, where we take an iterative procedure to converge to \( \lambda^* \) 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 \( \lambda_k \). Then, the multiplier for the next iteration is given by \( (\epsilon_k \lambda) \) 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 \( \epsilon_k \) sequence, \( \lambda_k \rightarrow \lambda^* \) [32].

Despite the generality of (18) and (21), directly solving them is complicated. Thus, in the following section, we first consider a slightly less general setting where user channels are static (can be different across users) and the resource expenditure option is binary.\(^3\) Results for the general case will be presented 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 \( \zeta_n \). 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 \( (r, \tau) \).

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 \( (D_n = 0 \text{ for all } n) \).

**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:
\[ V_n(0, \tau) = \frac{1 - (1 - \zeta_n)^{\tau}}{\zeta_n}(\lambda + \zeta_n \beta_n), \quad 0 < \tau \leq \tau_n + D_n. \] (22)

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

**Proof:** See Appendix A.

\(^3\)Due to the complexity of the general stochastic scenario, similar static settings were also considered in the literature, e.g., [33] and [34].
Remark 1: When \( \lambda = \lambda^* \), based on the KKT conditions [32], it can be shown that for packets with \( \zeta_n \beta_n = \lambda^* \), the optimal policy is to transmit the packet at every timeslot with probability \( \bar{p}_P = \frac{\sum_{\zeta_n \beta_n = \lambda^*} \bar{a}_n E_n}{\bar{a}_n E_n} \), where \( E_n = \zeta_n + 2(1 - \zeta_n) \zeta_n + \cdots + (\tau_n + D_n)(1 - \zeta_n)^{\tau_n + D_n - 1} \), 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 \( \tau_n \). Otherwise, packets from user \( \tau_n \) should never be served if the expected reward is less than the cost (from the point of optimizing the weighted throughput under an average resource constraint). From Theorem 1 and (17), we have the following corollary.

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

\[
g_P(\lambda) = \max_{\pi} L(\pi, \lambda) = \sum_{n: \zeta_n \beta_n > \lambda} \alpha_n \frac{1 - (1 - \zeta_n)^{\tau_n + D_n}}{\zeta_n} (-\lambda + \zeta_n \beta_n) + \lambda B,
\]

and

\[
\phi^*_P = \min_{\lambda} \sum_{n: \zeta_n \beta_n > \lambda} \alpha_n \frac{1 - (1 - \zeta_n)^{\tau_n + D_n}}{\zeta_n} (-\lambda + \zeta_n \beta_n) + \lambda B.
\]

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 \( \phi^*_0 \) 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:

\[
\phi^*_P - \phi^*_0 \leq \sum_{n: \zeta_n \beta_n > \lambda^*_0} \frac{\alpha_n}{\zeta_n} \left[ (1 - \zeta_n)^{\tau_n} - (1 - \zeta_n)^{\tau_n + D_n} \right] \times (-\lambda^*_0 + \zeta_n \beta_n)
\]

and

\[
\phi^*_P - \phi^*_0 \geq \sum_{n: \zeta_n \beta_n > \lambda^*_P} \frac{\alpha_n}{\zeta_n} \left[ (1 - \zeta_n)^{\tau_n} - (1 - \zeta_n)^{\tau_n + D_n} \right] \times (-\lambda^*_P + \zeta_n \beta_n).
\]

Proof: From (24), we have that:

\[
\phi^*_P = \min_{\lambda} g_P(\lambda) \leq \phi_P(\lambda^*_0),
\]

\[
\phi^*_0 = \min_{\lambda} g_0(\lambda) \leq g_0(\lambda^*_P).
\]

Thus, \( g_P(\lambda^*_P) - g_0(\lambda^*_0) \leq \phi^*_P - \phi^*_0 \leq g_P(\lambda^*_0) - g_0(\lambda^*_0) \), which gives us Theorem 2.

Theorem 2 suggests two things. (i) for each user \( \tau_n \), prediction improves the throughput by an amount \( \Theta(\rho^* (1 - \rho^* \tau_n)) \) with \( \rho = 1 - \zeta_n \). This implies that the impact of prediction is decreasing with the deadline \( \tau_n \), which is expected. (ii) the gap to the optimal improvement \( \rho^* \) decreases exponentially as the prediction power \( D_n \) increases. This demonstrates the power of prediction and highlights the potential of investing in improving system prediction.

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 these predicted arrivals, on the other hand, the system will be able to start their services the moment they appear in the prediction window, following (11) to (15).

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 \( \tau_n \).

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

\[
c_n \triangleq \frac{\lambda}{(-\lambda + \zeta_n \beta_n)(1 - \zeta_n)^{\tau_n} + \lambda}.
\]

(i) If \( \rho_n > c_n \), then 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. The value function is given by:

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

where \( 0 < w \leq D_n \). (ii) If \( \rho_n \leq c_n \), the value function is given by:

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

where \( 0 < w \leq D_n \). Specially, if \( \rho_n < c_n \), 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 \( \zeta_n \beta_n < \lambda \).

Proof: See Appendix B.

Remark 2: Similar to Theorem 1, when \( \lambda = \lambda^* \), if there exist \( n_1, n_2 \) such that \( p_{n_1} = c_{n_1}, \zeta_n \beta_{n_2} = \lambda^* \), then the optimal policy is to preserve \( n_1 \) packets with certain probability \( \rho_{n_1} \) and to transmit \( n_2 \) packets with certain probability \( \rho_{n_2} \), so that \( E_{av} = B \).

Theorem 3 shows that for a predicted user \( \tau_n \) 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 \( \rho_n \frac{1 - (1 - \zeta_n)^w}{\zeta_n}(-\lambda + \zeta_n \beta_n) \). This is consistent with our intuition compared with (22), as that the probability that a predicted arrival is real is \( p_n \). Also note that although \( q_n \) does not appear in \( V_n(0, \tau_n) \), we will see in the following corollary that it indirectly impacts the final timely-throughput by affecting the effective arrival rate as in (19).

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

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

Theorem 4: Let $g_1(\lambda)$ and $\phi^*_1$ denote the the dual function of (3) and the optimal weighted timely-throughput in the imperfect prediction case. We have:

$$g_1(\lambda) = \lambda B + \sum_{n: \zeta_n \beta_n > \lambda} \left( \hat{a}_n v_n(\lambda) \right)$$

$$+ \left( \max_{n: \zeta_n \beta_n > \lambda} \hat{a}_n \right) q_n \frac{1 - (1 - \zeta_n)^{\tau_n}}{\zeta_n} \left( -\lambda + \zeta_n \beta_n \right),$$

and

$$\phi^*_1 = \min_{\lambda} g_1(\lambda),$$

where $\hat{a}_n$ is the rate of predicted arrivals given in (19).

Combining Corollaries 1 and 2, we conclude the following theorem regarding the improvement in timely-throughput with imperfect prediction.

**Theorem 4:** Suppose $\lambda^*_1 = \arg\min g_1(\lambda)$. The weighted timely-throughput improvement satisfies

$$\phi^*_1 - \phi^*_0 \leq \sum_{n: \zeta_n \beta_n > \lambda^*_0} \left\{ \hat{a}_n v_n(\lambda^*_0) \right\}$$

$$- \frac{(a_n p_n - \max_{n: \zeta_n \beta_n > \lambda^*_0} a_n q_n)}{(p_n - q_n) \zeta_n} \left( 1 - (1 - \zeta_n)^{\tau_n} \right) \left( -\lambda^*_0 + \zeta_n \beta_n \right),$$

and

$$\phi^*_1 - \phi^*_0 \geq \sum_{n: \zeta_n \beta_n > \lambda^*_1} \left\{ \hat{a}_n v_n(\lambda^*_1) \right\}$$

$$- \frac{(a_n p_n - \max_{n: \zeta_n \beta_n > \lambda^*_1} a_n q_n)}{(p_n - q_n) \zeta_n} \left( 1 - (1 - \zeta_n)^{\tau_n} \right) \left( -\lambda^*_1 + \zeta_n \beta_n \right),$$

where $\hat{a}_n$ is given in (19).

**Proof:** Similar to the proof of Theorem 2. □

**Remark 3:** Similar to Theorem 2, we still have in the imperfect prediction case that, the improvement scales in the order of $O(p_n \left[ (a_n - \max_{n: \zeta_n \beta_n > \lambda^*_0} a_n) p_n^{-1} \right] (1 - q_n) \left( 1 - (1 - \zeta_n)^{\tau_n} \right) \left( -\lambda^*_0 + \zeta_n \beta_n \right) \left( 1 - (1 - \zeta_n)^{\tau_n + D_n} \right))$, which recovers the perfect prediction result when $p_n = 1$ and $q_n = 0$. This general result shows how different parameters affect the optimal timely-throughput, and will be useful for guiding prediction and control algorithm design for general computing systems.

### B. Influence of Prediction Accuracy

We now investigate how prediction accuracy impacts performance improvement. The following theorem summarizes our results.

**Theorem 5:** Let $q = \{q_1, q_2, \ldots, q_N\}$ denote the false-negative rate vector, and let $p = \{p_1, p_2, \ldots, p_N\}$ be the true-positive rate vector. Then,

(i) The optimal weighted timely-throughput $\phi^*_1$ is a non-decreasing function of $q$.

(ii) If $q = 0$, then the optimal weighted timely-throughput $\phi^*_1$ is a non-decreasing function of $p$.

**Proof:** See Appendix C. □

The results, though intuitive, are non-trivial to establish and require detailed investigation of the structure of the value functions. The impact of general $(p, q)$ pairs, on the other hand, is much more complicated to characterize, as we will see in Fig. 6 in the simulation section.

### 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 $\zeta_n(i, c)$ is a concave strictly increasing function of $c$. 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:

$$e^*_n(0, \tau + 1, i) \leq e^*_n(0, \tau, i), \quad \forall n, i,$$

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

**Proof:** See Appendix D. □

Theorem 6 shows that the optimal scheduling policy is a “lazy” policy, i.e., the server will not try hard to transmit the packet unless it is getting close to the deadline. From the proof of Theorem 6, we see that this is because the value function under the same channel state is monotonically non-decreasing with the time-to-deadline. Thus, 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. This result nicely complements existing efficient scheduling results in the literature [35]–[37], and can be viewed as an extension to the predictive online scheduling setting.

It is tempting to conclude that similar property also holds in the general imperfect prediction case. However, this monotonicity actually does not hold under imperfect prediction. This is because the value function is no longer monotone due to prediction error. This fact will be illustrated by simulation in Section VI, where we see that the resource level can actual 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 \( \varsigma_{n}(i, e) \) and \( \varsigma_{n}^{\text{max}}(i, e) \) respectively. We have:

\[
V_{n}(\tau) \triangleq \max_{e > 0} \left\{ 1 - \left[ 1 - \varsigma_{n}(\varsigma_{n}^{\text{min}}(i, e), e) \right]^{\tau} - \lambda e + \varsigma_{n}(i, e) \beta_{n} \right\},
\]

(36)

where \( V_{n}(0) = 0 \), and

\[
V_{n}^{u}(\tau) \triangleq \sum_{e = 0}^{\varsigma_{n}^{\text{max}}(i, e)} \left\{ -\lambda e + \varsigma_{n}(\varsigma_{n}^{\text{min}}(i, e), e) \beta_{n} \right\},
\]

(37)

where \( V_{n}^{u}(0) = 0 \). For each user \( n \) packet, given a fixed \( \lambda \), the value function satisfies:

\[
\max \{ 0, V_{n}^{l}(\tau) \} \leq V_{n}(0, \tau, i) \leq \min \{ \beta_{n}, V_{n}^{u}(\tau) \},
\]

(38)

where \( 0 < \tau \leq \tau_{n} + D_{n} \).

**Proof:** Similar with Theorem 1, Theorem 7 can be proved via induction, but it requires subtle design of upper and lower bounds and a careful induction step. For details, please refer to the supplementary material.

We then immediately have the following corollary, which shows that the functions \( V_{n}^{l} \) and \( V_{n}^{u} \) enable us to bound the optimal timely-throughput with prediction.

**Corollary 3:** Define

\[
g_{P}^{l}(\lambda) \triangleq \lambda B + \sum_{n=1}^{N} a_{n} \min \{ \beta_{n}, V_{n}^{u}(\tau_{n} + D_{n}) \}
\]

(39)

and

\[
g_{P}^{u}(\lambda) \triangleq \lambda B + \sum_{n=1}^{N} a_{n} \max \{ 0, V_{n}^{l}(\tau_{n} + D_{n}) \},
\]

(40)

where \( V_{n}^{l}(\tau) \) and \( V_{n}^{u}(\tau) \) are defined in (36) and (37).

**Theorem 8:** Let \( \lambda_{0}^{*} = \arg \min_{\lambda} g_{P}^{l}(\lambda) \) and \( \lambda_{P}^{*} = \arg \min_{\lambda} g_{P}^{l}(\lambda) \). The weighted timely-throughput improvement satisfies:

\[
g_{P}^{l}(\lambda_{0}^{*}) - g_{0}^{l}(\lambda_{0}^{*}) \leq \phi_{P}^{l} - \phi_{0}^{l} \leq g_{P}^{u}(\lambda_{P}^{*}) - g_{0}^{u}(\lambda_{0}^{*}).
\]

(43)

**Proof:** Similar to the proof of Theorem 2.

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. Nonetheless, they are tight when applied to the static setting.

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 (36), and \( \tilde{V}_{n}^{u}(\tau) = V_{n}^{u}(\tau) \) in (37). Then, for \( \tau_{n} \leq \tau \leq \tau_{n} + D_{n} \), define

\[
\tilde{V}_{n}^{l}(\tau) = \max_{e > 0} \left\{ -\lambda e + \varsigma_{n}(\varsigma_{n}^{\text{max}}(i, e), e) \right\},
\]

(44)

and

\[
\tilde{V}_{n}^{u}(\tau) = \sum_{e = 0}^{\varsigma_{n}^{\text{max}}(i, e)} \left\{ -\lambda e + \varsigma_{n}(\varsigma_{n}^{\text{min}}(i, e), e) \right\},
\]

(45)

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 \( \lambda \), the value function satisfies:

\[
V_{n}(0, \tau_{n} + w, i) \geq \max \{ 0, \tilde{V}_{n}(\tau_{n}), \tilde{V}_{n}^{l}(\tau_{n} + w) \}
\]

(46)

\[
V_{n}(0, \tau_{n} + w, i) \leq \min \{ p_{n} \beta_{n}, \tilde{V}_{n}^{u}(\tau_{n} + w) \},
\]

(47)

where \( 0 < w \leq D_{n} \).

**Proof:** Similar with Theorem 3, Theorem 9 can be proved via induction, but it requires subtle design of upper and lower bounds and a careful induction step. For details, please refer to the supplementary material.

Interestingly, we will see in the proof that the three terms in max\{0, \( V_{n}^{l}(\tau_{n}), \tilde{V}_{n}^{l}(\tau_{n} + w) \} \} in (46) correspond to not transmitting, transmitting after packet arrival, and proactive transmission.

From Theorem 9, we have the following corollary that bounds the optimal timely-throughput.

**Corollary 4:** Define

\[
g_{I}^{l}(\lambda) \triangleq \lambda B + \sum_{n=1}^{N} \left\{ \tilde{a}_{n} \min \{ p_{n} \beta_{n}, \tilde{V}_{n}^{u}(\tau_{n} + D_{n}) \} \right\}
\]

(48)

and

\[
g_{I}^{u}(\lambda) \triangleq \lambda B + \sum_{n=1}^{N} \left\{ \tilde{a}_{n} \max \{ 0, \tilde{V}_{n}^{l}(\tau_{n}), \tilde{V}_{n}^{u}(\tau_{n} + D_{n}) \} \right\}
\]

(49)

**Let** \( g_{I}^{l}(\lambda) \) and \( g_{I}^{u}(\lambda) \) be the dual function and the optimal timely-throughput in the imperfect prediction case, respectively. We have:

\[
g_{I}^{l}(\lambda) \leq g_{I}^{u}(\lambda) \leq g_{I}^{l}(\lambda), \quad \forall \lambda \geq 0
\]

(50)

and

\[
\min_{\lambda} g_{I}^{l}(\lambda) \leq \phi_{I} \leq \min_{\lambda} g_{I}^{u}(\lambda).
\]

(51)
From the above, we can now bound the improvement in weighted timely-throughput with imperfect prediction.

**Theorem 10:** Suppose \( \lambda^*_t = \arg \min_{\lambda} g^t_1(\lambda) \), then the weighted timely-throughput improvement satisfies:

\[
g^t_1(\lambda^*_t) - g^t_0(\lambda^*_t) \leq \phi^*_1 - \phi^*_0 \leq g^t_1(\lambda^*_0) - g^t_0(\lambda^*_0).
\]

**Proof:** Similar to the proof of Theorem 2. □

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. We will also see in the simulation section that the performance in the general case is similar to that in the static case as characterized in Theorems 2 and 4.

### B. Capacity Constraint

In our model, we have assumed that there is no transmission capacity constraint to facilitate analysis. Here we discuss what happens when we face a hard transmission capacity constraint rather than the average resource constraint, i.e.,

\[
\max \quad \beta^T x \\
\text{s.t.} \quad \sum_{n=1}^{N} c_n(t) \leq C, \quad \forall t,
\]

where \( c_n(t) \) is the number of user \( n \) packets transmitted at time \( t \).

In this case, the problem is significantly different as the scheduling decisions of packets are dependent. To solve this problem, we first consider a relaxation of problem (53) with an average transmission constraint,

\[
\max \quad \beta^T x \\
\text{s.t.} \quad c_{av} \leq C.
\]

Here the scheduling decision at each time slot is whether to serve/pre-serve a packet or not, similar to the scenario in Section IV. Thus, we can use the same method as before to derive the optimal policy \( \pi^* \) for (54). Then, we can construct a truncated policy \( \tilde{\pi}^* \) based on \( \pi^* \) as follows.

(a) Solve problem (54) to get the optimal policy \( \pi^* \).

(b) At slot \( t \), define \( \Omega_{\pi^*}(t) \) as the set of packets that are scheduled to be transmitted under \( \pi^* \). If \( |\Omega_{\pi^*}(t)| \leq C \), the scheduling decisions of \( \tilde{\pi}^* \) are the same as \( \pi^* \). Otherwise if \( |\Omega_{\pi^*}(t)| > C \), randomly select a subset of packets \( \Omega_{\pi^*}(t) \subseteq \Omega_{\pi^*}(t) \) with \( |\Omega_{\pi^*}(t)| = C \) to transmit, and discard packets in \( \Omega_{\pi^*}(t) - \Omega_{\pi^*}(t) \).

That is, \( \tilde{\pi}^* \) is constructed by first obtaining the optimal policy which only satisfies the transmission constraint on average, and then using truncation to make it feasible. The truncated policy \( \tilde{\pi}^* \) can be shown to be asymptotically optimal by using a similar argument as that in [10]. The proof is similar and is omitted here.

### VI. Simulation Results

We present simulation results of the optimal scheduling policy in this section. Our simulation is conducted for the system in Fig. 1 with \( N = 4 \) users. The arrival rates of packets are given by \( (a_1, \ldots, a_4) = (0.7, 0.6, 0.4, 0.3) \) with \( a_{\text{max}} = 1 \) and the deadlines are given by \( (\tau_1, \ldots, \tau_4) = (2, 3, 4, 5) \).

We set the reward weight vector to be \( \beta = (3, 1, 2, 4) \). Each channel has \( K = 4 \) states \( (s_1, \ldots, s_4) = (1, 2, 3, 4) \), each representing a noise level. We assume the channel state transition matrix is the same for all users and is given by:

\[
P = \begin{bmatrix}
0.4 & 0.3 & 0.2 & 0.1 \\
0.25 & 0.3 & 0.25 & 0.2 \\
0.2 & 0.25 & 0.3 & 0.25 \\
0.1 & 0.2 & 0.3 & 0.4
\end{bmatrix}.
\]

The set of resource levels \( \mathcal{E} \) is a discrete set of the form \{0.0001\(z\), \(z = 0, 1, \ldots, 6 \times 10^4\)\}. 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:

\[
\zeta_n(i,e) = \frac{2}{1 + e^{-2d_n/s_i}} - 1,
\]

where \( d = (d_1, \ldots, d_4) = (1.1, 1.2, 1.3, 1.4) \) 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 \( p = (0.8, 0.8, 0.8, 0.8) \) and \( q = (0.2, 0.1, 0.1, 0.2) \).

### A. 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 (\( r_2 = 3 \)) and User 4 (\( r_4 = 5 \)), i.e., \( e^*_n(0, \tau, i) \), in the zero prediction and perfect prediction cases. From the results, we can verify the monotonicity of the optimal policy, as shown in Theorem 6. We also see that the resource offered by the server for User 4 is larger than that under the same state for User 2. This is intuitive since \( \beta_4 > \beta_2 \).

Table III shows the optimal policy in the imperfect prediction case. From the scheduling decisions for User 2 under channel state \( s_1 \), we see that the monotonicity property actually does not hold when there is prediction error, i.e., resource

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>3.94</td>
<td>3.97</td>
<td>3.99</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.01</td>
<td>4.04</td>
<td>4.07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>1.39</td>
<td>1.47</td>
<td>1.63</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.27</td>
<td>1.05</td>
<td>5.54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2.60</td>
<td>2.74</td>
<td>2.92</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2.39</td>
<td>2.96</td>
<td>3.56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.16</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>1.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>0.40</td>
<td>0.30</td>
<td>0.20</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>0.20</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.10</td>
<td>0.20</td>
<td>0.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>τ</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>0.60</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
</tbody>
</table>
TABLE II
OPTIMAL SCHEDULING DECISIONS $e^*_n(0, \tau, i)$ (PERFECT PREDICTION). (a) USER 2. (b) USER 4

<table>
<thead>
<tr>
<th>$s(t)$</th>
<th>$\tau$</th>
<th>3+2</th>
<th>3+1</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1.3651</td>
<td>1.4248</td>
<td>1.4906</td>
<td>1.5797</td>
<td>1.7565</td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.6726</td>
<td>1.1571</td>
<td>1.6759</td>
<td></td>
</tr>
<tr>
<td>$s_3$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>$s_4$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

(a)

(b)

TABLE III
OPTIMAL SCHEDULING DECISIONS $e^*_n(0, \tau, i)$ (IMPERFECT PREDICTION). (a) USER 2. (b) USER 4

<table>
<thead>
<tr>
<th>$s(t)$</th>
<th>$\tau$</th>
<th>3+2</th>
<th>3+1</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>2.3912</td>
<td>2.5051</td>
<td>2.8355</td>
<td>2.7873</td>
<td>2.9807</td>
<td>3.3168</td>
</tr>
<tr>
<td>$s_2$</td>
<td>1.2882</td>
<td>1.948</td>
<td>2.5292</td>
<td>3.1024</td>
<td>3.7309</td>
<td>4.6111</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.315</td>
<td>4.5133</td>
</tr>
<tr>
<td>$s_4$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.4852</td>
</tr>
</tbody>
</table>

expenditure for state $s_1$ and $\tau = 3 + 1$ is smaller than that for state $s_1$ and $\tau = 3 + 2$. Another interesting observation is that the resource allocated in the imperfect prediction case is less than that for the same state in the perfect prediction case. This is because resource may be wasted when prediction is imperfect. Thus, the server is more conservative in resource allocation.

B. 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 $\tau_n$ 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 than that without prediction, demonstrating the effectiveness of prediction.

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. 6 shows the relationship between the optimal timely-throughput and the prediction accuracy. We set the true-positive rates and the false-negative rates the same for different users, and vary their values to draw the optimal timely-throughput surface. From this optimal timely-throughput surface, we can see that generally the timely-throughput increases with the true-positive rates and decreases with the false-negative rates, but the detailed correlation is complicated. To make it more clear, in Fig. 7, we fix $q$ and $p$ respectively to show how timely throughput changes with the other. In the left plot, the false-negative rate vector is set to $q = 0$, and all users have the same true-positive rate increasing from...
This stochastic case (this is proven for the static scenario) increases from $p$, we see that the timely-throughput is increasing in the true-positive rate and decreasing in the false-negative rate for $q$. From the results, there is a single channel state.

![Timely-throughput change with $p$ and $q$.](image)

The relationship between timely-throughput and general $(p, q)$ pair is complicated.

![Timely-throughput as a function of $p$ and $q$.](image)

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 when there is a single channel state.

In this paper, we investigate the fundamental benefit of predictive scheduling in improving timely-throughput in a general stochastic single-server multi-user system. Based on an error rate-based prediction model, we first derive a Markov decision process (MDP) solution for optimizing the timely-throughput objective, subject to an average resource consumption constraint. We then explicitly characterize the optimal scheduling 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 useful guidelines for designing future predictive control algorithms.

**VII. CONCLUSION**

In this paper, we investigate the fundamental benefit of predictive scheduling in improving timely-throughput in a general stochastic single-server multi-user system. Based on an error rate-based prediction model, we first derive a Markov decision process (MDP) solution for optimizing the timely-throughput objective, subject to an average resource consumption constraint. We then explicitly characterize the optimal scheduling 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 useful guidelines for designing future predictive control algorithms.

**APPENDIX A**

**Proof of Theorem 1**

Here we prove Theorem 1. Recall that the state of a packet is described by $(r, \tau)$ in this case, where $r \in \{0, 1\}$ indicates whether it is delivered and $0 \leq \tau \leq \tau_n$ denotes the remaining time until deadline.

**Proof:** In the static scenario, the Bellman equations in (8) to (10) reduce to

$$V_n(0, \tau) = \max \{ V_n(0, \tau - 1), -\lambda + \zeta_n V_n(1, \tau - 1) \} + (1 - \zeta_n) V_n(0, \tau - 1), \quad 0 < \tau \leq \tau_n + D_n,$$

$$V_n(0, 0) = 0, \quad \forall n,$$

$$V_n(1, \tau) = \beta_n, \quad \forall n, 0 \leq \tau \leq \tau_n + D_n. \quad (55)$$

We now prove the first part of Theorem 1 via induction, i.e., 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, and the value function is $V_n(0, \tau) = \frac{1 - (1 - \zeta_n)^{\tau - \lambda}}{\zeta_n} (\lambda + \zeta_n \beta_n), 0 < \tau \leq \tau_n + D_n$.

First, we show that this statement holds for $\tau = 1$. To see this, note that solving the one-step equation of (55) gives:

$$V_n(0, 1) = \max \{ 0, -\lambda + \zeta_n \beta_n \} = -\lambda + \zeta_n \beta_n. \quad (56)$$

Thus, if $-\lambda + \zeta_n \beta_n > 0$, the optimal scheduling decision at state $(0, 1)$ is to transmit the packet.

Now suppose the statement holds for $\tau = 1, \ldots, t$, we will show that it holds for $\tau = t + 1$. From (55), we have:

$$V_n(0, t + 1) = \max \{ V_n(0, t), -\lambda + \zeta_n \beta_n \} + (1 - \zeta_n) V_n(0, t)$$

$$= \max \left\{ \frac{1 - (1 - \zeta_n)^t}{\zeta_n} (-\lambda + \zeta_n \beta_n), 1 - (1 - \zeta_n)^{t+1} \right\}.$$ 

Thus, if $-\lambda + \zeta_n \beta_n > 0$, $V_n(0, t + 1) = \frac{1 - (1 - \zeta_n)^{t+1}}{\zeta_n} (-\lambda + \zeta_n \beta_n)$, and the optimal scheduling decision at state $(0, t + 1)$ is to transmit the packet. The proof of the second part goes the same. This completes the proof of Theorem 1. □

**APPENDIX B**

**Proof of Theorem 3**

Here we prove Theorem 3.
Proof: First note that the Bellman equations in (11) to (15) reduce to the following.

\[
V_n(0, \tau) = \max \left\{ V_n(0, \tau - 1) - \lambda + \zeta_n V_n(1, \tau - 1) + (1 - \zeta_n) V_n(0, \tau - 1), \right. \\
\left. 0 < \tau \leq \tau_n + D_n, \quad \tau \neq \tau_n + 1 \right\}
\]

\[
V_n(0, \tau_n + 1) = \max \left\{ p_n V_n(0, \tau_n), -\lambda + \zeta_n V_n(1, \tau_n) + (1 - \zeta_n) p_n V_n(0, \tau_n) \right\}
\]

We start from Part (A), i.e., \( \zeta_n \beta_n > \lambda \). Consider \( p_n > c_n = \frac{\lambda}{\lambda - \zeta_n \beta_n + (1 - \zeta_n) \lambda} \). From Theorem 1, we know that \( V_n(0, \tau_n) = \frac{1 - (1 - \zeta_n)^{\tau_n}}{\zeta_n} (-\lambda + \zeta_n \beta_n) \). We want to prove the statement by induction on \( w \).

First, the statement holds for \( w = 1 \). To see this, note that \( V_n(0, \tau_n + 1) = \max \{ p_n V_n(0, \tau_n), -\lambda + \zeta_n V_n(1, \tau_n) + (1 - \zeta_n) p_n V_n(0, \tau_n) \} \).

Since \( p_n > c_n \), one can verify that \(-\lambda + \zeta_n V_n(1, \tau_n) + (1 - \zeta_n) p_n V_n(0, \tau_n) > p_n V_n(0, \tau_n) \). Therefore,

\[
V_n(0, \tau_n + 1) = -\lambda + \zeta_n V_n(1, \tau_n) + (1 - \zeta_n) p_n V_n(0, \tau_n) = -\lambda + \zeta_n V_n(1, \tau_n) + (1 - \zeta_n) p_n V_n(0, \tau_n) + (1 - p_n) \left\{ \frac{1 - (1 - \zeta_n)^{\tau_n}}{\zeta_n} (-\lambda + \zeta_n \beta_n) \right\},
\]

and the optimal scheduling decision at state \( (0, \tau_n + 1) \) is to transmit (pre-serve) the predicted packet.

Then, suppose the statement holds for \( w = 1, \ldots, t \), we show that it also holds for \( w = t + 1 \). From (57), we have:

\[
V_n(0, \tau_n + t + 1) = \max \left\{ V_n(0, \tau_n + t) - \lambda + \zeta_n p_n \beta_n + (1 - \zeta_n) V_n(0, \tau_n + t) \right\},
\]

\[
= \max \left\{ -(1 - p_n) \frac{1 - (1 - \zeta_n)^t}{\zeta_n} \lambda + p_n \frac{1 - (1 - \zeta_n)^{t+1}}{\zeta_n} (-\lambda + \zeta_n \beta_n), \quad (t + 1 - p_n) \frac{1 - (1 - \zeta_n)^{t+1}}{\zeta_n} (-\lambda + \zeta_n \beta_n), \quad (1 - p_n) \frac{1 - (1 - \zeta_n)^{t+1}}{\zeta_n} (-\lambda + \zeta_n \beta_n) \right\},
\]

Since \( p_n > c_n \), we have \( V_n(0, \tau_n + t + 1) = -(1 - p_n) \frac{1 - (1 - \zeta_n)^t}{\zeta_n} \lambda + p_n \frac{1 - (1 - \zeta_n)^{t+1}}{\zeta_n} (-\lambda + \zeta_n \beta_n) \).

We now claim \( v_n(\lambda) \geq p_n \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n) \). To see this, note that if \( p_n \leq \zeta_n \lambda \), then \( v_n(\lambda) = p_n \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n) \). Otherwise, if \( p_n > \zeta_n \lambda \), the optimal pre-service policy is to transmit the packet at every time-slot in the prediction window. But \( p_n \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n) \) is the value function if the packet is not pre-erved. Thus, \( v_n(\lambda) \geq p_n \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n) \). Combining this with the fact that \( \frac{\partial g_1}{\partial p_n} \leq 0 \), which finishes the proof for part (i).

Part (ii) When \( q = 0 \), we have:

\[
g_1(\lambda, p) = \lambda B + \sum_{n : \zeta_n \beta_n > \lambda} \frac{a_n}{p_n} v_n(\lambda)
\]

\[
= \lambda B + \sum_{p_n \leq \zeta_n} \frac{a_n}{\zeta_n} \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n)
\]

\[
+ \sum_{p_n > \zeta_n} \frac{a_n}{p_n} \left[ -(1 - p_n) \frac{1 - (1 - \zeta_n)^{n+D_n}}{\zeta_n} (-\lambda + \zeta_n \beta_n) \right],
\]

where \( n \) satisfies \( \zeta_n \beta_n > \lambda \). Similar to the proof of Part (i), we are going to show \( \frac{\partial g_1}{\partial p_n} \geq 0 \).

Apparently we have \( \frac{\partial g_1}{\partial p_n} = 0 \) if \( \zeta_n \beta_n \leq \lambda \). Now suppose \( \zeta_n \beta_n > \lambda \). If \( p_n > \zeta_n \lambda \), then

\[
\frac{\partial g_1}{\partial p_n} = \frac{a_n}{p_n \zeta_n} \left[ 1 - (1 - \zeta_n)^{D_n} \right] \lambda \geq 0.
\]

Thus, \( g_1(\lambda, p) \) is non-decreasing in \( p_n \) on \( [\zeta_n \lambda, 1] \).

If \( p_n \leq \zeta_n \lambda \), let \( \Delta p_n = (0, \ldots, \Delta p_n, 0, \ldots) \) with \( p_n \geq 0 \).

If \( p_n + \Delta p_n \leq \zeta_n \lambda \) then \( g_1(\lambda, p + \Delta p_n) = g_1(\lambda, p) \). Otherwise, if \( p_n + \Delta p_n > \zeta_n \lambda \), then \( g_1(\lambda, p + \Delta p_n) \geq g_1(\lambda, p) \). Hence, \( g_1(\lambda, p) \) is non-decreasing in \( p_n \), \( \forall n \).

Appendix C
Proof of Theorem 5

We prove Theorem 5 here.

Proof: (Part ii) For clarity, we write \( g_1 \) and \( f_1^* \) as explicit functions of \( q \), i.e., \( g_1(\lambda, q) \) and \( f_1^*(q) \). Since \( f_1^*(q) = \min_{\lambda} g_1(\lambda, q) \), in order to prove the result, we will show that for any fixed \( \lambda \), \( g_1(\lambda, q) \) is a non-increasing function of \( q \).

We know that if \( \zeta_n \beta_n \leq \lambda \), then \( \frac{\partial g_1}{\partial q_n} = 0 \). Otherwise, taking the derivative of (31) with respect to \( q_n \), we get (note that \( \tilde{a}_n \) is also a function of \( q_n \)):

\[
\frac{\partial g_1}{\partial q_n} = \frac{a_n - \max p_n}{(p_n - q_n)^2} v_n(\lambda)
\]

\[
+ \frac{(\max p_n - a_n) p_n}{(p_n - q_n)^2} \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n)
\]

\[
= \frac{a_n - \max p_n}{(p_n - q_n)^2} \left[ v_n(\lambda) - p_n \frac{1 - (1 - \zeta_n)^n}{\zeta_n} (-\lambda + \zeta_n \beta_n) \right]
\]

Appendix D
Proof of Theorem 6

Proof: First we show that the value function has the following properties:

(a) \( V_n(0, \tau, i) \leq \beta_n \forall i, 0 \leq \tau \leq \tau_n + D_n \),

(b) \( V_n(0, \tau, i) \leq V_n(0, \tau + 1, i) \forall i, 0 \leq \tau < \tau_n + D_n \).

Property (a) can be easily verified by induction on \( \tau \). Now, we prove (b) by induction on \( \tau \).

First, if \( \tau = 0 \), we see that (b) holds, since \( V_n(0, 1, i) = \max_{e \in E} [ -\lambda e + \zeta_n (i, e) \beta_n ] \geq 0, \forall i \).
Now suppose (b) holds for \( \tau = 0, \ldots, t - 1 \), we show that it holds for \( \tau = t \). To this end, we have:

\[
V_n(0, t + 1, i) = \max_{e \in \mathcal{E}} \left[ -\lambda e + \zeta_n(i, e) \beta_n + (1 - \zeta_n(i, e)) \sum_j P_{n,j}^i V_n(0, t, j) \right],
\]

\[
V_n(0, t, i) = \max_{e \in \mathcal{E}} \left[ -\lambda e + \zeta_n(i, e) \beta_n + (1 - \zeta_n(i, e)) \sum_j P_{n,j}^i V_n(0, t - 1, j) \right].
\]

Since \( V_n(0, t, j) \geq V_n(0, t - 1, j), \forall j, \) for a fixed \( e \),

\[
-\lambda e + \zeta_n(i, e) \beta_n + (1 - \zeta_n(i, e)) \sum_j P_{n,j}^i V_n(0, t - 1, j) \geq -\lambda e + \zeta_n(i, e) \beta_n + (1 - \zeta_n(i, e)) \sum_j P_{n,j}^i V_n(0, t - 1, j).
\]

Thus, \( V_n(0, t + 1, i) \geq V_n(0, t, i), \forall i. \)

If \( \mathcal{E} \) is continuous, we have from (8) that

\[
e_n^*(0, \tau, i) = \arg \max_{e \in \mathcal{E}} \left[ -\lambda e + \zeta_n(i, e) \beta_n + (1 - \zeta_n(i, e)) \sum_j P_{n,j}^i V_n(0, \tau - 1, j) \right].
\]

By setting the derivative of the right hand side with respect to \( e \) to 0, we obtain:

\[
\frac{d\zeta_n(i, e)}{de} = \frac{\lambda}{\beta_n - \sum_j P_{n,j}^i V_n(0, \tau - 1, j)}.
\]

Since \( \zeta_n(i, e) \) is a concave function of \( e \), from Property (a) and (b) we know (35) holds.

On the other hand, if \( \mathcal{E} \) is discrete, denote \( h_{n,\tau,i}(e) = -\lambda e + \zeta_n(i, e) \beta_n + (1 - \zeta_n(i, e)) \sum_j P_{n,j}^i V_n(0, \tau - 1, j) \). To prove (35), we will show that for any consumption levels \( e_1 \leq e_u, \) if \( h_{n,\tau,i}(e_1) \geq h_{n,\tau,i}(e_u) \), then \( h_{n,\tau+1,i}(e_1) \geq h_{n,\tau+1,i}(e_u) \). From this result, we can then conclude that any \( e \geq e_n^*(0, \tau, i) \) will result in \( h_{n,\tau+1,i}(e) \leq h_{n,\tau+1,i}(e_n^*(0, \tau, i)) \), which proves the result.

Since \( h_{n,\tau,i}(e_1) \geq h_{n,\tau,i}(e_u) \), we have that

\[
\lambda(e_u - e_1) - \beta_n(\zeta_n(i, e_u) - \zeta_n(i, e_1)) + (\zeta_n(i, e_u) - \zeta_n(i, e_1)) \sum_j P_{n,j}^i V_n(0, \tau - 1, j) \geq 0.
\]

Therefore,

\[
h_{n,\tau+1,i}(e_1) - h_{n,\tau+1,i}(e_u) = \lambda(e_u - e_1) - \beta_n(\zeta_n(i, e_u) - \zeta_n(i, e_1)) + (\zeta_n(i, e_u) - \zeta_n(i, e_1)) \sum_j P_{n,j}^i V_n(0, \tau - 1, j) \geq 0.
\]

Here the last step follows since \( V_n(0, \tau - 1, j) \leq V_n(0, \tau, j) \) and \( \zeta_n(i, e) \) is increasing in \( e \). This completes the proof. □

REFERENCES


Kun Chen received the B.Eng. degree in electronic engineering from Tsinghua University in 2014, with an Excellent Graduate Award, where he is currently pursuing the Ph.D. degree with the Institute for Interdisciplinary Information Sciences. His research interests include age-of-information, predictive control in information systems, and reinforcement learning.

Longbo Huang has been a Visiting Scholar at the LIDS lab, MIT, and also at the EECS Department, UC Berkeley. He was a Visiting Professor at The Chinese University of Hong Kong, Bell-labs France, and Microsoft Research Asia. He was a Visiting Scientist at the Simons Institute for the Theory of Computing, UC Berkeley, in 2016. He is currently an Assistant Professor with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China. His research interests are in the areas of stochastic modeling and analysis, machine learning, and optimal control. He was selected into China’s Youth 1000-Talent Program in 2013. He received the Outstanding Teaching Award from Tsinghua University in 2014, and the ACM SIGMETRICS Rising Star Research Award in 2018. He currently serves as an Associate Editor for the ACM Transactions on Modeling and Performance Evaluation of Computing Systems, and an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.