Learning-aided Stochastic Network Optimization with State Prediction

Longbo Huang*, Minghua Chen^{+}, Yunxin Liu†

*longbohuang@tsinghua.edu.cn, IIIS@Tsinghua University

+minghua@ie.cuhk.edu.hk, IE@CUHK

†yunxin.liu@microsoft.com, Microsoft Research Asia

Abstract—**We investigate the problem of stochastic network optimization in the presence of state prediction and non-stationarity. Based on a novel state prediction model featured with a distribution-accuracy curve, we develop the *predictive learning-aided control* (`PLC`) algorithm, which jointly utilizes historic and predicted network state information for decision making. `PLC` is an online algorithm that consists of three key components, namely sequential distribution estimation and change detection, dual learning, and online queue-based control. We show that for stationary networks, `PLC` achieves a near-optimal utility-delay tradeoff. For non-stationary networks, `PLC` obtains an utility-backlog tradeoff for distributions that last longer than a time proportional to the square of the prediction error, which is smaller than that needed by Backpressue [1] for achieving the same utility performance. Moreover, `PLC` detects distribution change $O(w)$ slots faster with high probability (w is the prediction size) and achieves a convergence time faster than that under Backpressure. Our results demonstrate that state prediction helps (i) achieve faster detection and convergence, and (ii) obtain better utility-delay tradeoffs. They also quantify the benefits of prediction in four important performance metrics, i.e., utility (efficiency), delay (quality-of-service), detection (robustness), and convergence (adaptability), and provide new insight for joint prediction, learning and optimization in stochastic networks.**

I. Introduction

Enabled by recent developments in sensing, monitoring, and machine learning methods, utilizing prediction for performance improvement in networked systems has received a growing attention in both industry and research. For instance, recent research works [2], [3], and [4] investigate the benefits of utilizing prediction in energy saving, job migration in cloud computing, and video streaming in cellular networks. On the industry side, various companies have implemented different ways to take advantage of prediction, e.g., Amazon utilizes prediction for better package delivery [5] and Facebook enables prefetching for faster webpage loading [6]. However, despite the continuing success in these attempts, most existing results in network control and analysis do not investigate the impact of prediction. Therefore, we still lack a thorough theoretical understanding about the *value-of-prediction* in stochastic network control. Fundamental questions regarding how prediction should be integrated in network algorithms, the ultimate prediction gains, and how prediction error impacts performance, remain largely unanswered.

To contribute to developing a theoretical foundation for utilizing prediction in networks, in this paper, we consider a general constrained stochastic network optimization formulation, and aim to rigorously quantify the benefits of system state prediction and the impact of prediction error. Specifically, we are given a discrete-time stochastic network with a dynamic state that evolves according to some potentially non-stationary probability law. Under each system state, a control action is chosen and implemented. The action generates traffic into network queues but also serves workload from them. The action also results in a system utility (cost) due to service completion (resource expenditure). The traffic, service, and cost are jointly determined by the action and the system state. The objective is to maximize the expected utility (or equivalently, to minimize the cost) subject to traffic/service constraints, given imperfect system state prediction information.

This is a general framework that models various practical scenarios, for instance, mobile networks, computer networks, supply chains, and smart grids. However, understanding the impact of prediction in this framework is challenging. First, statistical information of network dynamics is often unknown a-priori. Hence, in order to achieve good performance, algorithms must be able to quickly learn certain sufficient statistics of the dynamics, and make efficient use of prediction while carefully handling prediction error. Second, system states appear randomly in every time slot. Thus, algorithms must perform well under such incremental realizations of the randomness. Third, quantifying system service quality often involves handling queueing in the system. As a result, explicit connections between control actions and queues must be established.

There has been a recent effort in developing algorithms that can achieve good utility and delay performance for this general problem without prediction in various settings, for instance, wireless networks, [7], [8], [9], [10], processing networks, [11], [12], cognitive radio, [13], and the smart grid, [14], [15]. However, existing results mostly focus on networks with stationary distributions. They either assume full system statistical information beforehand, or rely on stochastic approximation techniques to avoid the need of such information. Works [16] and [17] propose schemes to incorporate historic system information into control, but they do not consider prediction. Recent results in [18], [19], [20], [21] and [22] consider problems with traffic demand prediction, and [23] jointly considers demand and channel prediction. However, they focus

either on $M/M/1$-type models, or do not consider queueing, or do not consider the impact of prediction error. Along a different line of work, [24], [25], [26] and [27] investigate the benefit of prediction from the online algorithm design perspective. Although the results provide new understanding about the effect of prediction, they do not apply to the general constrained network optimization problem in consideration, where action outcomes are general functions of time-varying network states, queues evolve in a controlled manner, i.e., arrival and departure rates depend on the control policy, and prediction can contain error.

In this paper, we develop a novel control algorithm for the general framework called *predictive learning-aided control* (PLC). PLC is an online scheme that consists of three components, sequential distribution estimation and change detection, dual learning, and online control (see Fig. 1).
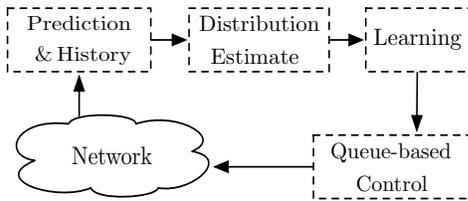

Fig. 1. The PLC algorithm contains (i) a distribution estimator that utilizes both historic and predicted information to simultaneously form a distribution estimation and detect distribution change, (ii) a learning component that computes an empirical Lagrange multiplier based on the estimation, and (iii) a queue-based controller whose decision-making information is augmented by the multiplier.

The distribution estimator conducts sequential statistical comparisons based on prediction and historic network state records. Doing so efficiently detects changes of the underlying probability distribution and guides us in selecting the right state samples to form distribution estimates. The estimated distribution is then fed into a dual learning component to compute an empirical multiplier of an underlying optimization formulation. This multiplier is further incorporated into the Backpressure (BP) controller [1] to perform realtime network operation. Compared to the commonly adopted receding-horizon-control approach (RHC), e.g., [28], PLC provides another way to utilize future state information, which focuses on using the predicted distribution for guiding action selection in the present slot and can be viewed as performing steady-state control under the predicted future distribution.

We summarize our main contributions as follows.

i. We propose a general state prediction model featured with a distribution-accuracy curve. Our model captures key factors of several existing prediction models, including window-based [22], distribution-based [29], and filter-based [26] models.

ii. We propose a general constrained network control algorithm called *predictive learning-aided control* (PLC). PLC is an online algorithm that is applicable to both stationary and non-stationary systems. It jointly performs distribution estimation and change detection, dual learning, and queue-based online control.

iii. We show that for stationary networks, PLC achieves an $[O(\epsilon), O(\log^2(1/\epsilon))]$ utility-delay tradeoff. For non-stationary networks, PLC obtains an $[O(\epsilon), O(\log^2(1/\epsilon)$

$+ \min(\epsilon^{c/2-1}, e_w/\epsilon))]$ utility-backlog tradeoff for distributions that last $\Theta(\frac{\max(\epsilon^{-c}, e_w^{-2})}{\epsilon^{1+a}})$ time, where e_w is the prediction accuracy, $c \in (0, 1)$ and $a > 0$ is an $\Theta(1)$ constant (the Backpressue algorithm [1] requires an $O(\epsilon^{-2})$ length for the same utility performance with a larger backlog).[1]

iv. We show that for both stationary and non-stationary system dynamics, PLC detects distribution change $O(w)$ slots (w is prediction window size) faster with high probability and achieves a fast $O(\min(\epsilon^{-1+c/2}, e_w/\epsilon) + \log^2(1/\epsilon))$ convergence time, which is faster than the $O(\epsilon^{-1+c/2} + \epsilon^{-c})$ time of the OLAC scheme [16], and the $O(1/\epsilon)$ time of Backpressure.

The rest of the paper is organized as follows. In Section II, we discuss a few motivating examples. We set up the notations in Section III, and present the problem formulation in Section IV. Background information is provided in Section V. Then, we present PLC and its analysis in Sections VI and VII. Simulation results are presented in Section VIII, followed by conclusions in Section IX. To facilitate reading, all the proofs are placed in the appendices.

II. Motivating Examples

In this section, we present a few interesting practical scenarios that fall into our general framework.

Matching in sharing platforms: Consider a Uber-like company that provides ride service to customers. At every time, customer requests enter the system and available cars join to provide service. Depending on the environment condition (state), e.g., traffic condition or customer status, matching customers to drivers can result in different user satisfaction, and affect the revenue of the company (utility). The company gets access to future customer demand and car availability, and system condition information (prediction), e.g., through reservation or machine learning tools. The objective is to optimally match customers to cars so that the utility is maximized, e.g., [30] and [31].

Energy optimization in mobile networks: Consider a base-station (BS) sending traffic to a set of mobile users. The channel conditions (state) between users and the BS are time-varying. Thus, the BS needs different amounts of power for packet transmission (cost) at different times. Due to higher layer application requirements, the BS is required to deliver packets to users at pre-specified rates. On the other hand, the BS can predict future user locations in some short period of time, from which it can estimate future channel conditions (prediction). The objective of the BS is to jointly optimize power allocation and scheduling among users, so as to minimize energy consumption, while meeting the rate requirements, e.g., [8], [13]. Other factors such as energy harvesting, e.g., [32], can also be incorporated in the formulation.

Resource allocation in cloud computing: Consider an operator, e.g., a dispatcher, assigning computing jobs to servers for processing. The job arrival process is time-varying (state), and available processing capacities at servers are also dynamic (state), e.g., due to background processing. Completing users' job requests brings the operator reward (utility). The operator

---

[1]Note that when there is no prediction, i.e., $w = 0$ and $e_w = \infty$, we recover previous results of OLAC [16].

may also have information regarding future job arrivals and service capacities (prediction). The goal is to allocate resources and to balance the loads properly, so as to maximize system utility. This example can be extended to capture other factors such as rate scaling [33] and data locality constraints [34].

In these examples and related works, not only can the state statistics be potentially non-stationary, but the systems also often get access to certain (possibly imperfect) future state information through various prediction techniques. These features make the problems different from existing settings considered, e.g., [8] and [15], and require different approaches for both algorithm design and analysis.

III. Notations

$\mathbb{R}^n$ denotes the n-dimensional Euclidean space. $\mathbb{R}_+^n$ ($\mathbb{R}_-^n$) denotes the non-negative (non-positive) orthant. Bold symbols $\boldsymbol{x} = (x_1, ..., x_n)$ denote vectors in $\mathbb{R}^n$. $\boldsymbol{w}$ 1 denotes "with probability 1." $\|\cdot\|$ denotes the Euclidean norm. For a sequence $\{y(t)\}_{t=0}^\infty$, $\overline{y} = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{y(\tau)\}$ denotes its average (when exists). $\boldsymbol{x} \succeq \boldsymbol{y}$ means $x_j \geq y_j$ for all j. For distributions $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$, $\|\boldsymbol{\pi}_1 - \boldsymbol{\pi}_2\|_{TV} = \sum_i |\pi_{1i} - \pi_{2i}|$ denotes the total variation distance.

IV. System Model

Consider a controller that operates a network with the goal of minimizing the time average cost, subject to the queue stability constraint. The network operates in slotted time, i.e., $t \in \{0, 1, 2, ...\}$, and there are $r \geq 1$ queues in the network.

A. Network state

In every slot t, $S(t)$ denotes the current network state, which summarizes current network parameters, such as a vector of conditions for each network link, or a collection of other relevant information about the current network channels and arrivals. $S(t)$ is independently distributed across time, and each realization is drawn from a state space of M distinct states denoted as $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$.[2] We denote $\pi_i(t) = \Pr\{S(t) = s_i\}$ the probability of being in state s_i at time t and denote $\boldsymbol{\pi}(t) = (\pi_1(t), ..., \pi_M(t))$ the state distribution. The network controller can observe $S(t)$ at the beginning of every slot t, but the $\pi_i(t)$ probabilities are unknown. We assume that each $\boldsymbol{\pi}(t)$ stays unchanged for multiple timeslots, and denote $\{t_k, k = 0, 1, ...\}$ the starting point of the k-th constant distribution interval $\mathcal{I}_k$, i.e., $\boldsymbol{\pi}(t) = \boldsymbol{\pi}_k$ for all $t \in \mathcal{I}_k \triangleq \{t_k, t_{k+1} - 1\}$. The length of $\mathcal{I}_k$ is denoted by $d_k \triangleq t_{k+1} - t_k$.

B. State prediction

At every time slot, the operator gets access to a prediction module, e.g., a machine learning algorithm, which provides prediction of future network states. Different from recent works, e.g., [25], [26] and [35], which assume prediction models on individual states, we assume that the prediction

module outputs a sequence of predicted distributions $\mathcal{D}_w(t) \triangleq \{\hat{\boldsymbol{\pi}}(t), \hat{\boldsymbol{\pi}}(t+1), ..., \hat{\boldsymbol{\pi}}(t+w)\}$, where $w + 1$ is the prediction window size. Moreover, the prediction quality is characterized by a distribution-accuracy curve $\{e(0), ..., e(w)\}$ as follows. For every $0 \leq k \leq w$, $\hat{\boldsymbol{\pi}}(t+k)$ satisfies:

$$\|\hat{\boldsymbol{\pi}}(t+k) - \boldsymbol{\pi}(t+k)\|_{TV} \leq e(k), \ \forall \ k \tag{1}$$

That is, the predicted distribution at time k has a total-variation error bounded by some $e(k) \geq 0$.[3] Note that $e(k) = 0$ for all $0 \leq k \leq w$ corresponds to a perfect predictor, in that it predicts the exact distribution in every slot. We assume the $\{e(0), ..., e(w)\}$ curve is *known* to the operator and denote $e_w \triangleq \frac{1}{w+1}\sum_{k=0}^w e(k)$ the average prediction error.

It is often possible to achieve prediction guarantees as in (1), e.g., by adopting a maximum likelihood estimator (Section 5.1 in [36]) based on historic state data. Also note that the prediction model in (1) is general and captures key features of several existing prediction models: (i) the exact distribution prediction model in [29], where the future demand distribution is known ($e(k) = 0$ for all k), (ii) the window-based prediction model, e.g., [22], where each $\hat{\boldsymbol{\pi}}(t+k)$ corresponds to the indicator for the true state, and (iii) the error-convolution prediction model in [35], [25] and [26], which captures key features of the Wiener filter and Kalman filter.

C. The cost, traffic, and service

At each time t, after observing $S(t) = s_i$, the controller chooses an action $x(t) \in \mathcal{X}_i$. The set $\mathcal{X}_i$ is called the feasible action set for network state s_i and is assumed to be time-invariant and compact for all $s_i \in \mathcal{S}$. The cost, traffic, and service generated by the action $x(t) = x_i$ are as follows:

(a) The chosen action has an associated cost given by the cost function $f(t) = f(S(t), x(t)) = f(s_i, x_i) : \mathcal{X}_i \mapsto \mathbb{R}_+$ (or $\mathcal{X}_i \mapsto \mathbb{R}_-$ in reward maximization problems).[4]

(b) The amount of traffic generated by the action to queue j is determined by the traffic function $A_j(t) = A_j(S(t), x(t)) = A_j(s_i, x_i) : \mathcal{X}_i \mapsto \mathbb{R}_+$, in units of packets.

(c) The amount of service allocated to queue j is given by the rate function $\mu_j(t) = \mu_j(S(t), x(t)) = \mu_j(s_i, x_i) : \mathcal{X}_i \mapsto \mathbb{R}_+$, in units of packets.

Here $A_j(t)$ can include both exogenous arrivals from outside the network to queue j, and endogenous arrivals from other queues, i.e., transmitted packets from other queues to queue j. We assume the functions $-f(s_i, \cdot)$, $\mu_j(s_i, \cdot)$ and $A_j(s_i, \cdot)$ are time-invariant, their magnitudes are uniformly upper bounded by some constant $\delta_{\max} \in (0, \infty)$ for all s_i, j, and they are known to the operator. Note that this formulation is general and models many network problems, e.g., [8], [15], and [37].

D. Problem formulation

Let $\boldsymbol{q}(t) = (q_1(t), ..., q_r(t))^T \in \mathbb{R}_+^r$, $t = 0, 1, 2, ...$ be the queue backlog vector process of the network, in units of

---

[2] The independent assumption is made to facilitate presentation and understanding. The results in this paper can likely be generalized to systems where $S(t)$ evolves according to general time inhomogeneous Markovian dynamics.

[3] We focus on state distribution prediction instead of predicting individual states. In this case, it makes sense to assume a deterministic upper bound of the difference because we are dealing with distributions.

[4] We use cost and utility interchangeably in this paper.

packets. We assume the following queueing dynamics:

$$q_j(t+1) = \max\big[q_j(t) - \mu_j(t) + A_j(t), 0\big], \quad \forall j \qquad (2)$$

and $\boldsymbol{q}(0) = \boldsymbol{0}$. By using (2), we assume that when a queue does not have enough packets to send, null packets are transmitted, so that the number of packets entering $q_j(t)$ is equal to $A_j(t)$. We adopt the following notion of queue stability [1]:

$$\bar{q}_{\text{av}} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{j=1}^{r} \mathbb{E}\big\{q_j(\tau)\big\} < \infty. \qquad (3)$$

We use Π to denote an action-choosing policy, and use f_{av}^{Π} to denote its time average cost, i.e.,

$$f_{\text{av}}^{\Pi} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{f^{\Pi}(\tau)\big\}, \qquad (4)$$

where $f^{\Pi}(\tau)$ is the cost incurred at time τ under policy Π. We call an action-choosing policy *feasible* if at every time slot t it only chooses actions from the feasible action set $\mathcal{X}_i$ when $S(t) = s_i$. We then call a feasible action-choosing policy under which (3) holds a *stable* policy.

In every slot, the network controller observes the current network state and prediction, and chooses a control action, with the goal of minimizing the time average cost subject to network stability. This goal can be mathematically stated as:[5]

(P1) $\min_{\Pi} \; f_{\text{av}}^{\Pi}$, s.t. (3).

In the following, we call **(P1)** *the stochastic problem*, and we use $f_{\text{av}}^{\boldsymbol{\pi}}$ to denote its optimal solution given a fixed distribution $\boldsymbol{\pi}$. It can be seen that the examples in Section II can all be modeled by our stochastic problem framework.

Throughout our paper, we make the following assumption.

Assumption 1. *For every system distribution $\boldsymbol{\pi}_k$, there exists a constant $\epsilon_k = \Theta(1) > 0$ such that for any valid state distribution $\boldsymbol{\pi}' = (\pi'_1, ..., \pi'_M)$ with $\|\boldsymbol{\pi}' - \boldsymbol{\pi}_k\|_{TV} \le \epsilon_k$, there exist a set of actions $\{x_z^{(s_i)}\}_{i=1,...,M}^{z=1,2,...,\infty}$ with $x_z^{(s_i)} \in \mathcal{X}_i$ and variables $\vartheta_z^{(s_i)} \ge 0$ for all s_i and z with $\sum_z \vartheta_z^{(s_i)} = 1$ for all s_i (possibly depending on $\boldsymbol{\pi}'$), such that:*

$$\sum_{s_i} \pi'_i \big\{ \sum_z \vartheta_z^{(s_i)} [A_j(s_i, x_z^{(s_i)}) - \mu_j(s_i, x_z^{(s_i)})] \big\}$$
$$\le -\eta_0, \; \forall j \qquad (5)$$

where $\eta_0 = \Theta(1) > 0$ is independent of $\boldsymbol{\pi}'$. $\Diamond$

Assumption 1 corresponds to the "slack" condition commonly assumed in the literature with $\epsilon_k = 0$, e.g., [37] and [38].[6] With $\epsilon_k > 0$, we assume that when two systems are relatively close to each other (in terms of $\boldsymbol{\pi}$), they can both be stabilized by some (possibly different) randomized control policy that results in the same slack.

E. Discussion of the model

Two key differences between our model and previous ones include (i) $\boldsymbol{\pi}(t)$ itself can be time-varying and (ii) the operator gets access to a prediction window $\mathcal{W}_w(t)$ that contains

---

[5]When $\boldsymbol{\pi}(t)$ is time-varying, the optimal system utility needs to be defined carefully. We will specify it when discussing the corresponding results.

[6]Note that $\eta_0 \ge 0$ is a necessary condition for network stability [1].

---

imperfect prediction. These two extensions are important to the current network control literature. First, practical systems are often non-stationary. Thus, system dynamics can have time-varying distributions. Thus, it is important to have efficient algorithms to automatically adapt to the changing environment. Second, prediction has recently been made increasingly accurate in various contexts, e.g., user mobility in cellular network and harvestable energy availability in wireless systems, by data collection and machine learning tools. Thus, it is critical to understand the fundamental benefits and limits of prediction, and its optimal usage. Finally, note that the convexity of the problem (**P1**) depends largely on the structure of the feasible action sets. In the case, when all feasible action sets are convex, it can be shown that the resulting problem is convex using a similar argument as that in [39].

V. THE DETERMINISTIC PROBLEM

For our later algorithm design and analysis, we define the *deterministic problem* and its dual problem [40]. Specifically, the deterministic problem for a given distribution $\boldsymbol{\pi}$ is defined as follows [40]:

$$\min : V \sum_{s_i} \pi_i f(s_i, x^{(s_i)}) \qquad (6)$$

$$\text{s.t. } \sum_{s_i} \pi_i [A_j(s_i, x^{(s_i)}) - \mu_j(s_i, x^{(s_i)})] \le 0, \; \forall j$$

$$x^{(s_i)} \in \mathcal{X}_i \quad \forall i = 1, 2, ..., M.$$

Here the minimization is taken over $\boldsymbol{x} \in \prod_i \mathcal{X}_i$, where $\boldsymbol{x} = (x^{(s_1)}, ..., x^{(s_M)})^T$, and $V \ge 1$ is a positive constant introduced for later analysis. The dual problem of (6) can be obtained as follows:

$$\max : \; g(\boldsymbol{\gamma}, \boldsymbol{\pi}), \quad \text{s.t. } \boldsymbol{\gamma} \succeq \boldsymbol{0}, \qquad (7)$$

where $g(\boldsymbol{\gamma}, \boldsymbol{\pi})$ is the dual function for problem (6) and is defined as:

$$g(\boldsymbol{\gamma}, \boldsymbol{\pi}) = \inf_{x^{(s_i)} \in \mathcal{X}_i} \sum_{s_i} \pi_i \bigg\{ V f(s_i, x^{(s_i)}) \qquad (8)$$
$$+ \sum_j \gamma_j \big[A_j(s_i, x^{(s_i)}) - \mu_j(s_i, x^{(s_i)}) \big] \bigg\}.$$

$\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_r)^T$ is the Lagrange multiplier of (6). It is well known that $g(\boldsymbol{\gamma}, \boldsymbol{\pi})$ in (8) is concave in the vector $\boldsymbol{\gamma}$ for all $\boldsymbol{\gamma} \in \mathbb{R}^r$. Hence, the problem (7) can usually be solved efficiently, e.g., using dual subgradient methods [41]. If the cost functions and rate functions are separable over different network components, the problem also admits distributed solutions [41]. We use $\boldsymbol{\gamma}_{\boldsymbol{\pi}}^*$ to denote the optimal multiplier corresponding to a given $\boldsymbol{\pi}$ and sometimes omit the subscript when it is clear. Denote $g_{\boldsymbol{\pi}}^*$ the optimal value of (7) under a fixed distribution $\boldsymbol{\pi}$. It was shown in [42] that:

$$f_{\text{av}}^{\boldsymbol{\pi}} = g_{\boldsymbol{\pi}}^*. \qquad (9)$$

That is, $g_{\boldsymbol{\pi}}^*$ characterizes the optimal time average cost of the stochastic problem. For our analysis, we make the following assumption on the $g(\boldsymbol{\gamma}, \boldsymbol{\pi}_k)$ function.

Assumption 2. *For every system distribution $\boldsymbol{\pi}_k$, $g(\boldsymbol{\gamma}, \boldsymbol{\pi}_k)$ has a unique optimal solution $\boldsymbol{\gamma}_{\boldsymbol{\pi}_k}^* \ne \boldsymbol{0}$ in $\mathbb{R}^r$.* $\Diamond$

Assumption 2 holds for many network utility optimization problems, e.g., [8], [40] and [43].

VI. PREDICTIVE LEARNING-AIDED CONTROL

In this section, we present the *predictive learning-aided control* algorithm (PLC). PLC contains three main components: a distribution estimator, a learning component, and an online queue-based controller. Below, we first present the estimation part. Then, we present the PLC algorithm.

A. Distribution estimation and change detection

Here we specify the distribution estimator. The idea is to first combine the prediction in $\mathcal{W}_w(t)$ with historic state information to form an *average* distribution, and then perform statistical comparisons for change detection. We call this module the *average distribution estimate* (ADE).

Specifically, ADE maintains two windows $\mathcal{W}_m(t)$ and $\mathcal{W}_d(t)$ to store network state samples, where $\mathcal{W}_d(t)$ roughly contains the most recent d state samples, and $\mathcal{W}_m(t)$ contains at most T_l state samples after $\mathcal{W}_d(t)$. The formal definition of them are given below, i.e.,

$$\mathcal{W}_d(t) = \{b_d^s(t), ..., b_d^e(t)\}, \quad (10)$$

$$\mathcal{W}_m(t) = \{b_m(t), ..., \min[b_d^s(t), b_m(t) + T_l]\}. \quad (11)$$

Here $b_d^s(t)$ and $b_m(t)$ mark the beginning slots of $\mathcal{W}_d(t)$ and $\mathcal{W}_m(t)$, respectively, and $b_d^e(t)$ marks the end of $\mathcal{W}_d(t)$. Ideally, $\mathcal{W}_d(t)$ contains the most recent d samples (including the prediction) and $\mathcal{W}_m(t)$ contains T_l subsequent samples (where T_l is a pre-specified number). We denote $W_m(t) = |\mathcal{W}_m(t)|$ and $W_d(t) = |\mathcal{W}_d(t)|$. Without loss of generality, we assume that $d \geq w + 1$. This assumption is made because, d grows with our control parameter V while prediction power is often limited in practice. We also denote $\mathcal{W}_w(t) \triangleq \{t..., t + w\}$.

We use $\hat{\boldsymbol{\pi}}^d(t)$ and $\hat{\boldsymbol{\pi}}^m(t)$ to denote the empirical distributions of $\mathcal{W}_d(t)$ and $\mathcal{W}_m(t)$, i.e.,[7]

$$\hat{\pi}_i^d(t) = \frac{1}{d}\left(\sum_{\tau=(t+w-d)_+}^{t-1} \mathbf{1}_{[S(\tau)=s_i]} + \sum_{\tau \in \mathcal{W}_w(t)} \hat{\pi}_i(\tau)\right)$$

$$\hat{\pi}_i^m(t) = \frac{1}{W_m(t)} \sum_{\tau \in \mathcal{W}_m(t)} \mathbf{1}_{[S(\tau)=s_i]}.$$

That is, $\hat{\boldsymbol{\pi}}^d(t)$ is the average of the empirical distribution of the "observed" samples in $\mathcal{W}_d(t)$ and the predicted distribution, whereas $\hat{\boldsymbol{\pi}}^m(t)$ is the empirical distribution of $\mathcal{W}_m(t)$.

The formal procedure of ADE is as follows (parameters T_l, d, ϵ_d will be specified later).

Average Distribution Estimate (ADE(T_l, d, ϵ_d)): Initialize $b_d^s(0) = 0$, $b_d^e(0) = t + w$ and $b_m(0) = 0$, i.e., $\mathcal{W}_d(t) = \{0, ..., t + w\}$ and $\mathcal{W}_m(t) = \phi$. At every time t, update $b_d^s(t)$, $b_d^e(t)$ and $b_m(t)$ as follows:

(i) If $W_m(t) \geq d$ and $||\hat{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)||_{TV} > \epsilon_d$, set $b_m(t) = t + w + 1$ and $b_d^s(t) = b_d^e(t) = t + w + 1$.

(ii) If $W_m(t) = T_l$ and there exists k such that $||\hat{\boldsymbol{\pi}}(t+k) - \hat{\boldsymbol{\pi}}^m(t)||_{TV} > e(k) + \frac{2M\log(T_l)}{\sqrt{T_l}}$, set $b_m(t) = b_d^s(t) = b_d^e(t) = t + w + 1$. Mark $t + w + 1$ a reset point.

[7]Note that this is only one way to utilize the samples. Other methods such as EWMA can also be applied when appropriate.

(iii) Else if $t \leq b_d^s(t-1)$, $b_m(t) = b_m(t-1)$, $b_d^s(t) = b_d^s(t-1)$, and $b_d^e(t) = b_d^e(t-1)$.[8]

(iv) Else set $b_m(t) = b_m(t-1)$, $b_d^s(t) = (t+w-d)_+$ and $b_d^e(t) = t + w$.

Output an estimate at time t as follow:

$$\boldsymbol{\pi}_a(t) = \begin{cases} \hat{\boldsymbol{\pi}}^m(t) & \text{if} \quad W_m(t) \geq T_l \\ \frac{1}{w+1}\sum_{k=0}^{w} \hat{\boldsymbol{\pi}}(t+k) & \text{else} \end{cases} \quad \diamond \quad (12)$$
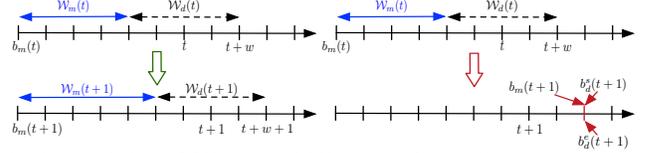
The idea of ADE is shown in Fig. 2.

Fig. 2. Evolution of $\mathcal{W}_m(t)$ and $\mathcal{W}_d(t)$. (Left) No change detected: The samples satisfy the criteria in Steps (i) and (ii). Thus, $\mathcal{W}_d(t)$ advances by one slot and $\mathcal{W}_m(t)$ increases its size by one. (Right) Change detected: The samples violate at least one of the conditions in (i) and (ii). Thus, both windows set their start and end points to $t + w + 1$.

The intuition of ADE is that if the environment is changing over time, we should rely on prediction for control. Else if the environment is stationary, then one should use the average distribution learned over time to combat the potential prediction error that may affect performance. T_l is introduced to ensure the accuracy of the empirical distribution and can be regarded as the confidence-level given to the distribution stationarity. A couple of technical remarks are also ready. (a) The term $2M\log(T_l)/\sqrt{T_l}$ is to compensate the inevitable deviation of $\hat{\boldsymbol{\pi}}^m(t)$ from the true value due to randomness. (b) In $\mathcal{W}_m(t)$, we only use the first T_l historic samples. Doing so avoids random oscillation in estimation and facilitates analysis.

Prediction is used in two ways in ADE. First, it is used in step (i) to decide whether the empirical distributions match (average prediction). Second, it is used to check whether prediction is consistent with the history (individual prediction). The reason for having this two-way utilization is to accommodate general prediction types. For example, suppose each $\hat{\boldsymbol{\pi}}(t+k)$ denotes the indicator for state $S(t+k)$, e.g., as in the look-ahead window model [22]. Then, step (ii) is loose since $e(k)$ is large, but step (i) will be useful. If $\hat{\boldsymbol{\pi}}(t+k)$ gets closer to the true distribution, both steps will be useful.

B. Predictive learning-aided control

We are now ready to present the PLC algorithm (shown in Fig. 1). The formal description is given below.

Predictive Learning-aided Control (PLC): At time t, do:
- (Estimation) Update $\boldsymbol{\pi}_a(t)$ with ADE(T_l, d, ϵ_d).
- (Learning) Solve the following empirical problem and compute the optimal Lagrange multiplier $\boldsymbol{\gamma}^*(t)$, i.e.,[9]

$$\max: \ g(\boldsymbol{\gamma}, \boldsymbol{\pi}_a(t)), \quad \text{s.t. } \boldsymbol{\gamma} \succeq \mathbf{0}, \quad (13)$$

[8]This step is evoked after we set $b_m(t') = b_d^s(t') = t' + w + 1 \geq t$ for some time t', in which case we the two windows remain unchanged until t is larger than $t' + w + 1$.

[9]The dual problem (13) is always concave [41]. Thus, it can be solved efficiently with existing convex optimization methods. Moreover, our results can be generalized to the case when the optimal Lagrange multiplier is computed only approximately. In particular, our results can be similarly proven if we obtain $\tilde{\boldsymbol{\gamma}}^*(t) = \boldsymbol{\gamma}^*(t) + err$. In this case, the utility bound remains the same and the delay bound will increase by $O(err)$.

If $\boldsymbol{\gamma}^*(t) = \infty$, set $\boldsymbol{\gamma}^*(t) = V \log(V) \cdot \mathbf{1}$. If $W_m(t-1) = T_l$ and $\boldsymbol{\pi}_a(t) \neq \boldsymbol{\pi}_a(t-1)$, set $\boldsymbol{q}(t+w+1) = \mathbf{0}$, i.e., drop all packets currently in the queues.

- (Control) At every time slot t, observe the current network state $S(t)$ and the backlog $\boldsymbol{q}(t)$. If $S(t) = s_i$, choose $x^{(s_i)} \in \mathcal{X}_i$ that solves the following:

$$\max: \quad -Vf(s_i, x) + \sum_{j=1}^{r} Q_j(t) \big[\mu_j(s_i, x) - A_j(s_i, x) \big]$$

$$\text{s.t.} \quad x \in \mathcal{X}_i, \tag{14}$$

where $Q_j(t) \triangleq q_j(t) + (\gamma_j^*(t) - \theta)^+$. Then, update the queues according to (2) with Last-In-First-Out. $\Diamond$

For readers who are familiar with the Backpressure (BP) algorithm, e.g., [1] and [44], the control component of PLC is the BP algorithm with its queue vector augmented by the empirical multiplier $\boldsymbol{\gamma}^*(t)$. Also note that packet dropping is introduced to enable quick adaptation to new dynamics if there is a distribution change. It occurs only when a long-lasting distribution ends, which avoids dropping packets frequently in a fast-changing environment.

We have the following remarks. **(i) Prediction usage:** Prediction is explicitly incorporated into control by forming an average distribution and converting the distribution estimate into a Lagrange multiplier. The intuition for having $T_l = \max(V^c, e_w^{-2})$ is that when e_w is small, we should rely on prediction as much as possible, and only switch to learned statistics when it is sufficiently accurate. **(ii) Connection with RHC:** It is interesting to see that when $W_m(t) < T_l$, PLC mimics the commonly adopted receding-horizon-control method (RHC), e.g., [28]. The main difference is that, in RHC, future *states* are predicted and are directly fed into a predictive optimization formulation for computing the current action. Under PLC, *distribution prediction* is combined with *historic state information* to compute an empirical multiplier for augmenting the controller. In this regard, PLC can be viewed as exploring the benefits of statistics whenever it finds the system stationary (and does it automatically). **(iii) Parameter selection:** The parameters in PLC can be conveniently chosen as follows. First, fix a detection error probability $\delta = V^{-\log(V)}$. Then, choose a small ϵ_d and a d that satisfies $d \geq 4 \log(V)^2 / \epsilon_d^2 + w + 1$. Finally, choose $T_l = \max(V^c, e_w^{-2})$ and θ according to (16).

While recent works [16] and [17] also design learning-based algorithms that utilize historic information, they do not consider prediction and do not provide insight on its benefits and the impact of prediction error. Moreover, [16] focuses on stationary systems and [17] adopts a frame-based scheme.

VII. PERFORMANCE ANALYSIS

This section presents the performance results of PLC. We focus on four metrics, detection efficiency, network utility, service delay, and algorithm convergence. The metrics are chosen to represent robustness, resource utilization efficiency, quality-of-service, and adaptability, respectively. Since our main objective is to investigate how these metrics behave under our algorithm, we treat network parameters M and r as constants.

A. Detection and estimation

We first look at the detection and estimation part. The following lemma summarizes the performance of ADE, which is affected by the prediction accuracy as expected.

Lemma 1. *Under* ADE(T_l, d, ϵ_d)*, we have:*

(a) Suppose at a time t, $\boldsymbol{\pi}(\tau_1) = \boldsymbol{\pi}_1$ for $\tau_1 \in \mathcal{W}_d(t)$ and $\boldsymbol{\pi}(\tau_2) = \boldsymbol{\pi}_2 \neq \boldsymbol{\pi}_1$ for all $\tau_2 \in \mathcal{W}_m(t)$ and $\max|\pi_{1i} - \pi_{2i}| > 4(w+1)e_w/d$. Then, by choosing $\epsilon_d < \epsilon_0 \triangleq \max|\pi_{1i} - \pi_{2i}|/2 - (w+1)e_w/d$ and $d > \ln\frac{4}{\delta} \cdot \frac{1}{2\epsilon_d^2} + w + 1$, if $W_m(t) \geq W_d(t) = d$, with probability at least $1 - \delta$, $b_m(t+1) = t+w+1$ and $\mathcal{W}_m(t+1) = \phi$, i.e., $W_m(t+1) = 0$.

(b) Suppose $\boldsymbol{\pi}(t) = \boldsymbol{\pi}$ $\forall t$. Then, if $W_m(t) \geq W_d(t) = d$, under ADE(T_l, d, ϵ_d) *with $d \geq \ln\frac{4}{\delta} \cdot \frac{2}{\epsilon_d^2} + w + 1$, with probability at least $1 - \delta - (w+1)MT_l^{-2\log(T_l)}$, $b_m(t+1) = b_m(t)$. $\Diamond$*

Proof. See Appendix A. $\qquad\square$

Lemma 1 shows that for a stationary system, i.e., $\boldsymbol{\pi}(t) = \boldsymbol{\pi}$, $W_m(t)$ will likely grow to a large value (Part (b)), in which case $\boldsymbol{\pi}_a(t)$ will stay close to $\boldsymbol{\pi}$ most of the time. If instead $\mathcal{W}_m(t)$ and $\mathcal{W}_d(t)$ contain samples from different distributions, ADE will reset $\mathcal{W}_m(t)$ with high probability. Note that since the first $w + 1$ slots belong to prediction, PLC detects changes $O(w)$ slots faster compared to methods without prediction. The condition $\max|\pi_{1i} - \pi_{2i}| > 4(w+1)e_w/d$ can be understood as follows. If we want to distinguish two different distributions, we want the detection threshold to be no more than half of the distribution distance. Now with prediction, we want the potential prediction error to be no more than half of the threshold, hence the factor 4. Also note that the delay involved in detecting a distribution change is nearly order-optimal, i.e., $d = O(1/\min_i |\pi_{1i} - \pi_{2i}|^2)$ time, which is known to be necessary for distinguishing two distributions [45]. Moreover, $d = O(\ln(1/\delta))$ shows that a logarithmic window size is enough to ensure a high detection accuracy.

B. Utility and delay

In this section, we look at the utility and delay performance of PLC. To state our results, we first define the following structural property of the system.

Definition 1. *A system is called polyhedral with parameter $\rho > 0$ under distribution $\boldsymbol{\pi}$ if the dual function $g(\boldsymbol{\gamma}, \boldsymbol{\pi})$ satisfies:*

$$g(\boldsymbol{\gamma}^*, \boldsymbol{\pi}) \geq g(\boldsymbol{\gamma}, \boldsymbol{\pi}) + \rho\|\boldsymbol{\gamma}_{\boldsymbol{\pi}}^* - \boldsymbol{\gamma}\|. \quad \Diamond \tag{15}$$

The polyhedral property often holds for practical systems, especially when action sets are finite (see [40] for more discussions).

1) Stationary system: We first consider stationary systems, i.e., $\boldsymbol{\pi}(t) = \boldsymbol{\pi}$. Our theorem shows that PLC achieves the near-optimal utility-delay tradeoff for stationary networks. This result is important, as any good adaptive algorithm must be able to handle stationary settings well.

Theorem 1. *Suppose $\boldsymbol{\pi}(t) = \boldsymbol{\pi}$, the system is polyhedral with $\rho = \Theta(1)$, $e_w > 0$, and $\boldsymbol{q}(0) = \mathbf{0}$. Choose $0 < \epsilon_d < \epsilon_0 \triangleq$*

$2(w+1)e_w/d$, $d = \log(V)^3/\epsilon_d^2$, $T_l = \max(V^c, e_w^{-2})$ for $c \in (0,1)$ *and*

$$\theta = 2\log(V)^2(1 + \frac{V}{\sqrt{T_l}}). \qquad (16)$$

Then, with a sufficiently large V, PLC achieves the following:

(a) Utility: $f_{av}^{PLC} = f_{av}^{\boldsymbol{\pi}} + O(1/V)$

(b) Delay: For all but an $O(\frac{1}{V})$ fraction of traffic, the average packet delay is $D = O(\log(V)^2)$

(c) Dropping: The packet dropping rate is $O(V^{-1})$. ◊

Proof. See Appendix B. □

Here c is a constant used for deciding the learning time T_l. Choosing $\epsilon = 1/V$, we see that PLC achieves the near-optimal $[O(\epsilon), O(\log(1/\epsilon)^2)]$ utility-delay tradeoff. Moreover, prediction enables PLC to also greatly reduce the queue size (see Part (b) of Theorem 2). Our result is different from the results in [20] and [22] for proactive service settings, where delay vanishes as prediction power increases. This is because we only assume observability of future states but not pre-service, and highlights the difference between pre-service and pure prediction. Note that the performance of PLC does not depend heavily on ϵ_d in Theorem 1. The value ϵ_d is more crucial for non-stationary systems, where a low false-negative rate is critical for performance. Also note that although packet dropping can occur during operation, the fraction of packets dropped is very small, and the resulting performance guarantee cannot be obtained by simply dropping the same amount of packets, in which case the delay will still be $\Theta(1/\epsilon)$.

Although Theorem 1 has a similar form as those in [17] and [16], the analysis is very different, in that (i) prediction error must be taken into account, and (ii) PLC performs sequential detection and decision-making.

2) Piecewise stationary system: We now turn to the non-stationary case and consider the scenario where $\boldsymbol{\pi}(t)$ changes over time. In this case, we see that prediction is critical as it significantly accelerates convergence and helps to achieve good performance when each distribution only lasts for a finite time. As we know that when the distribution can change arbitrarily, it is hard to even define optimality. Thus, we consider the case when the system is piecewise stationary, i.e., each distribution lasts for a duration of time, and study how the algorithm optimizes the performance for each distribution.

The following theorem summarizes the performance of PLC in this case. In the theorem, we define $D_k \triangleq t_k + d - t^*$, where $t^* \triangleq \sup\{t < t_k + d : t \text{ is a reset point}\}$, i.e., the most recent time when a cycle with size no smaller than T_l ends (recall that reset points are marked in step (ii) of ADE and $d \geq w+1$).

Theorem 2. *Suppose $d_k \geq 4d$ and the system is polyhedral with $\rho = \Theta(1)$ for all k. Also, suppose there exists $\epsilon_0^* = \Theta(1) > 0$ such that $\epsilon_0^* \leq \inf_{k,i} |\pi_{ki} - \pi'_{k-1i}|$ and $\boldsymbol{q}(0) = \boldsymbol{0}$. Choose $\epsilon_d < \epsilon_0^*$ in ADE, and choose d, θ and T_l as in Theorem 1. Fix any distribution $\boldsymbol{\pi}_k$ with length $d_k = \Theta(V^{1+a}T_l)$ for some $a = \Theta(1) > 0$.[10] Then, under PLC with a sufficiently large V, if $\mathcal{W}_m(t_k)$ only contains samples after t_{k-1}, we achieve the following with probability $1 - O(V^{-3\log(V)/4})$:*

[10]The constant a here is introduced to show that our results hold as long as d_k is larger than $O(VT_l)$.

(a) Utility: $f_{av}^{PLC} = f_{av}^{\boldsymbol{\pi}_k} + O(1/V) + O(\frac{D_k\log(V)}{T_l V^{1+a}})$

(b) Queueing: $\bar{q}_{av} = O((\min(V^{1-c/2}, Ve_w)+1)\log^2(V) + D_k + d)$.

(c) In particular, if $d_{k-1} = \Theta(T_l V^{a_1})$ for $a_1 = \Theta(1) > 0$ and $W_m(t_{k-1})$ only contains samples after t_{k-2}, then with probability $1 - O(V^{-2})$, $D_k = O(d)$, $f_{av}^{PLC} = f_{av}^{\boldsymbol{\pi}_k} + O(1/V)$ and $\bar{q}_{av} = O(\min(V^{1-c/2}, Ve_w) + \log^2(V))$. ◊

Proof. See Appendix C. □

A few remarks are in place. (i) Theorem 2 shows that, with an increasing prediction power, i.e., a smaller e_w, it is possible to simultaneously reduce network queue size and the time it takes to achieve a desired average performance (even if we do not execute actions ahead of time). The requirement $d_k = \Theta(V^{1+a}T_l)$ can be strictly less than the $O(V^{2-c/2+a})$ requirement for RLC in [17] and the $O(V^2)$ requirement of BP for achieving the same average utility. This implies that PLC finds a good system operating point faster, a desirable feature for network algorithms. (ii) The dependency on D_k is necessary. This is because PLC does not perform packet dropping if previous intervals do not exceed length T_l. As a result, the accumulated backlog can affect decision making in the current interval. Fortunately the queues are shown to be small and do not heavily affect performance (also see simulations). (iii) To appreciate the queueing result, note that BP (without learning) under the same setting will result in an $O(V)$ queue size.

Compared to the analysis in [17], one complicating factor in proving Theorem 2 is that ADE may not always throw away samples from a previous interval. Instead, ADE ensures that with high probability, only $o(d)$ samples from a previous interval will remain. This ensures high learning accuracy and fast convergence of PLC. One interesting special case not covered in the last two theorems is when $e_w = 0$. In this case, prediction is perfect and $T_l = \infty$, and PLC always runs with $\boldsymbol{\pi}_a(t) = \frac{1}{w+1}\sum_{k=0}^{w}\hat{\boldsymbol{\pi}}(t+k)$, which is the exact average distribution. For this case, we have the following result.

Theorem 3. *Suppose $e_w = 0$ and $\boldsymbol{q}(0) = \boldsymbol{0}$. Then, PLC achieves the following:*

(a) Suppose $\boldsymbol{\pi}(t) = \boldsymbol{\pi}$ and the system is polyhedral with $\rho = \Theta(1)$. Then, under the conditions of Theorem 1, PLC achieves the $[O(\epsilon), O(\log(1/\epsilon)^2)]$ utility-delay tradeoff.

(b) Suppose $d_k \geq d\log^2(V)$ and the system is polyhedral with $\rho = \Theta(1)$ under each $\boldsymbol{\pi}_k$. Under the conditions of Theorem 2, for an interval $d_k \geq V^{1+\epsilon}$ for any $\epsilon > 0$, PLC achieves that $f_{av}^{PLC} = f_{av}^{\boldsymbol{\pi}_k} + O(1/V)$ and $\mathbb{E}\{\boldsymbol{q}(t_k)\} = O(\log^4(V))$. ◊

Proof. See Appendix D. □

The intuition is that since prediction is perfect, i.e., $\boldsymbol{\pi}_a(t) = \boldsymbol{\pi}_k$ during $[t_k + d, t_{k+1} - w]$. Therefore, a better performance can be achieved. The key challenge in this case is that PLC does not perform any packet dropping. Thus, queues can build up and one needs to show that the queues will be concentrating around $\theta \cdot \boldsymbol{1}$ even when the distribution changes.

C. Convergence time

We now consider the algorithm convergence time, which is an important evaluation metric and measures how long it takes for an algorithm to reach its steady-state. While recent works [17], [16], [46], and [47] also investigate algorithm convergence time, they do not consider utilizing prediction in learning and do not study the impact of prediction error.

To formally state our results, we adopt the following definition of convergence time from [16].

Definition 2. *Let $\zeta > 0$ be a given constant and let $\boldsymbol{\pi}$ be a system distribution. The ζ-convergence time of a control algorithm, denoted by T_ζ, is the time it takes for the effective queue vector $\boldsymbol{Q}(t)$ to get to within ζ distance of $\boldsymbol{\gamma}_{\boldsymbol{\pi}}^*$, i.e.,[11]*

$$T_\zeta \triangleq \inf\{t \,|\, ||\boldsymbol{Q}(t) - \boldsymbol{\gamma}_{\boldsymbol{\pi}}^*|| \le \zeta\}. \, \diamond \qquad (17)$$

With this definition, we have the following theorem. Recall that $w \le d = \Theta(\log(V)^2)$.

Theorem 4. *Assuming all conditions in Theorem 2, except that $\boldsymbol{\pi}(t) = \boldsymbol{\pi}_k$ for all $t \ge t_k$. If $e_w = 0$, under* PLC,

$$\mathbb{E}\{T_G\} = O(\log^4(V)). \qquad (18)$$

Else suppose $e_w > 0$. Under the conditions of Theorem 2, with probability $1 - O(\frac{1}{VT_l} + \frac{D_k}{V^2 T_l})$,

$$\mathbb{E}\{T_G\} = O(\theta + T_l + D_k + w) \qquad (19)$$
$$\mathbb{E}\{T_{G_1}\} = O(d). \qquad (20)$$

Here $G = \Theta(1)$ and $G_1 = \Theta(D_k + 2\log(V)^2(1 + V e_w))$, where D_k is defined in Theorem 2 as the most recent reset point before t_k. In particular, if $d_{k-1} = \Theta(T_l V^{a_1})$ for some $a_1 = \Theta(1) > 0$ and $\theta = O(\log(V)^2)$, then with probability $1 - O(V^{-2})$, $D_k = O(d)$, and $\mathbb{E}\{T_{G_1}\} = O(\log^2(V))$. $\diamond$

Proof. See Appendix E. $\square$

The assumption $\boldsymbol{\pi}(t) = \boldsymbol{\pi}_k$ for all $t \ge t_k$ is made to avoid the need for specifying the length of the intervals. It is interesting to compare (18), (19) and (20) with the convergence results in [16] and [17] without prediction, where it was shown that the convergence time is $O(V^{1-c/2}\log(V)^2 + V^c)$, with a minimum of $O(V^{2/3})$. Here although we may still need $O(V^{2/3})$ time for getting into an G-neighborhood (depending on e_w), getting to the G_1-neighborhood can take only an $O(\log^2(V))$ time, which is much faster compared to previous results, e.g., when $e_w = o(V^{-2})$ and $D_k = O(w)$, we have $G_1 = O(\log^2(V))$. This confirms our intuition that *prediction accelerates algorithm convergence* and demonstrates the power of (even imperfect) prediction.

VIII. Simulation

In this section, we present simulation results of PLC in a two-queue system shown in Fig. 3. Though being simple, the system models various settings, e.g., a two-user downlink transmission problem in a mobile network, a CPU scheduling problem with two applications, or an inventory control system where two types of orders are being processed.

---

[11]T_ζ is essentially the *hitting time* of the process $\boldsymbol{Q}(t)$ to the area $\{||\boldsymbol{Q}(t) - \boldsymbol{\gamma}_{\boldsymbol{\pi}}^*|| \le \zeta\}$ [48].
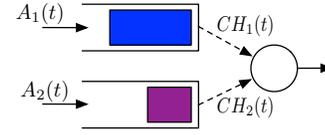

Fig. 3. A single-server two-queue system. Each queue receives random arrivals. The server can only serve one queue at a time.

$A_j(t)$ denotes the number of arriving packets to queue j at time t. We assume $A_j(t)$ is i.i.d. being 1 or 0 with probabilities p_j and $1 - p_j$, and use $p_1 = 0.3$ and $p_2 = 0.6$. Thus, $\lambda_1 = 0.3$ and $\lambda_2 = 0.6$. Each queue has a time-varying channel condition. We denote $CH_j(t)$ the channel condition of queue j at time t. We assume that $CH_j(t) \in \mathcal{CH}_j$ with $\mathcal{CH}_1 = \{0, 1\}$ and $\mathcal{CH}_2 = \{1, 2\}$. The channel distributions are assumed to be uniform. At each time, the server determines the power allocation to each queue. We use $P_j(t)$ to denote the power allocated to queue j at time t. Then, the instantaneous service rate $q_j(t)$ gets is given by:

$$\mu_j(t) = \log(1 + CH_j(t)P_j(t)). \qquad (21)$$

We assume that $P_j(t) \in \mathcal{P} = \{0, 1, 2\}$ for $j = 1, 2$, and at each time only one queue can be served. The objective is to stabilize the system with minimum average power. It can be verified that Assumptions 1 and 2 both hold in this example.

We compare PLC with BP in two cases. The first case is a stationary system where the arrival distributions remain constant. The second case is a non-stationary case, where we change the arrival distributions during the simulation. In both cases we simulate the system for $T = 5 \times 10^4$ slots with $V \in \{20, 50, 100, 150, 200, 300\}$. We set $w + 1 = 5$ and generate prediction error by adding uniform random noise to distributions with max value $e(k)$ (specified below). We also use $\epsilon_d = 0.1$, $\delta = 0.005$ and $d = 2\ln(4/\delta)/\epsilon^2 + w + 1$. We also simplify the choice of θ and set it to $\theta = \log(V)^2$.

We first examine the long-term performance. Fig. 4 shows the utility-delay performance of PLC compared to BP in the stationary setting. There are two PLC we simulated, one is with $e_w = 0$ (PLC) and the other is with $e_w = 0.04$ (PLC-e). From the plot, we see that both PLCs achieve a similar utility as BP, but guarantee a much smaller delay. The reason PLC-e has a better performance is due to packet dropping. We observe around an average packet dropping rate of 0.06. As noted before, the delay reduction of PLC cannot be achieved by simply dropping this amount of packets.
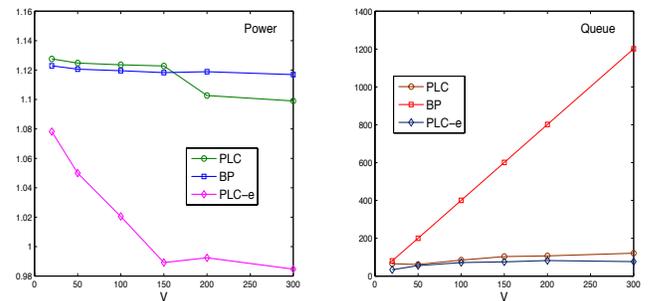

Fig. 4. Utility and delay performance comparison between PLC and BP.

Next, we take a look at the detection and convergence performance of PLC. Fig. 5 shows the performance of PLC

with perfect prediction ($e_w = 0$), PLC with prediction error ($e_w = 0.04$) and BP when the underlying distribution changes. Specifically, we run the simulation for $T = 5000$ slots and start with the arrival rates of $p_1 = 0.2$ and $p_2 = 0.4$. Then, we change them to $p_1 = 0.3$ and $p_2 = 0.6$ at time $T/2$.
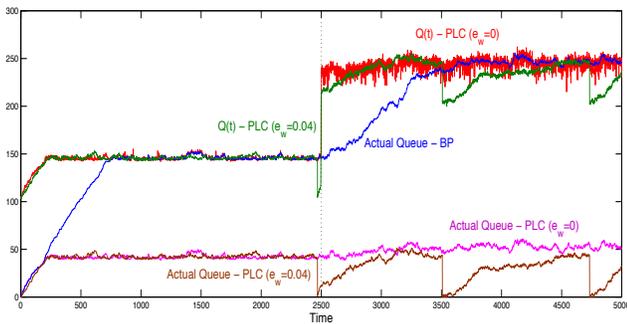

Fig. 5. Convergence comparison between PLC and BP for queue 1 under $V = 100$. PLC ($e_w = 0$) is the perfect case and PLC ($e_w = 0.04$) contains prediction error. Both versions converge much faster compared to BP.

We can see from the green and red curves that PLC quickly adapts to the change and modifies the Lagrange multiplier accordingly. By doing so, the actual queues under PLC (the purple and the brown curves) remain largely unaffected. For comparison, we see that BP takes a longer time to adapt to the new distribution and results in a larger queue size. We also see that during the 5000 slots, PLC ($e_w = 0.04$) drops packets 3 times (zero for the first half), validating the results in Lemma 1 and Theorem 1. Moreover, after the distribution change, PLC ($e_w = 0.04$) quickly adapts to the new equilibrium, despite having imperfect prediction. The fast convergence result also validates our theorem about short term utility performance under PLC. Indeed, if we look at slots during time $200-500$, and slots between $2500-3500$, we see that when BP is learning the target backlog, PLC already operates near the optimal mode. This shows the benefits of prediction and learning in stochastic network control.

IX. CONCLUSION

We investigate the problem of stochastic network optimization in the presence of imperfect state prediction and non-stationarity. Based on a novel distribution-accuracy curve prediction model, we develop the *predictive learning-aided control* (PLC) algorithm. PLC is an online algorithm that contains three main functionalities, sequential distribution estimation and change detection, dual learning, and online queue-based control. We show that PLC simultaneously achieves good long-term performance, short-term queue size reduction, accurate change detection, and fast algorithm convergence. Our results demonstrate that state prediction can help improve performance and quantify the benefits of prediction in four important metrics, i.e., utility (efficiency), delay (quality-of-service), detection (robustness), and convergence (adaptability). They provide new insight for joint prediction, learning and optimization in stochastic networks.

REFERENCES

[1] L. Georgiadis, M. J. Neely, and L. Tassiulas. *Resource Allocation and Cross-Layer Control in Wireless Networks*. Foundations and Trends in Networking Vol. 1, no. 1, pp. 1-144, 2006.

[2] Y. Chon, E. Talipov, H. Shin, and H. Cha. Mobility prediction-based smartphone energy optimization for everyday location monitoring. *ACM Sensys*, 2011.

[3] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. Effective straggler mitigation: Attack of the clones. *ACM NSDI*, 2014.

[4] X. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, . Jin, J. Rexford, and R. K. Sinha. Can accurate predictions improve video streaming in cellular networks? *ACM HotMobile*, 2015.

[5] TechCrunch. Amazon patents "anticipatory" shipping - to start sending stuff before you've bought it. *http://techcrunch.com/2014/01/18/amazon-pre-ships/*, Jan 2014.

[6] Adweek. Facebook begins prefetching to improve mobile site speed. *http://www.adweek.com/socialtimes/prefetching/644281*, Aug 2016.

[7] M. Gatzianas, L. Georgiadis, and L. Tassiulas. Control of wireless networks with rechargeable batteries. *IEEE Trans. on Wireless Communications, Vol. 9, No. 2*, Feb. 2010.

[8] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Trans. Netw.*, 15(6):1333–1344, 2007.

[9] B. Li and R. Srikant. Queue-proportional rate allocation with per-link information in multihop networks. *Proceedings of ACM Sigmetrics*, 2015.

[10] B. Ji and Y. Sang. Throughput characterization of node-based scheduling in multihop wireless networks: A novel application of the gallai-edmonds structure theorem. *Proceedings of ACM MobiHoc*, 2016.

[11] H. Zhao, C. H. Xia, Z. Liu, and D. Towsley. A unified modeling framework for distributed resource allocation of general fork and join processing networks. *Proc. of ACM Sigmetrics*, 2010.

[12] L. Jiang and J. Walrand. Stable and utility-maximizing scheduling for stochastic processing networks. *Allerton Conference on Communication, Control, and Computing*, 2009.

[13] R. Urgaonkar and M. J. Neely. Opportunistic scheduling with reliability guarantees in cognitive radio networks. *IEEE Transactions on Mobile Computing*, 8(6):766–777, June 2009.

[14] H. Su and A. El Gamal. Modeling and analysis of the role of fast-response energy storage in the smart grid. *Proc. of Allerton*, 2011.

[15] M. J. Neely R. Urgaonkar, B. Urgaonkar and A. Sivasubramaniam. Optimal power cost management using stored energy in data centers. *Proceedings of ACM Sigmetrics*, June 2011.

[16] L. Huang, X. Liu, and X. Hao. The power of online learning in stochastic network optimization. *Proceedings of ACM Sigmetrics*, 2014.

[17] L. Huang. Receding learning-aided control in stochastic networks. *IFIP Performance*, Oct 2015.

[18] J. Tadrous, A. Eryilmaz, and H. El Gamal. Proactive resource allocation: harnessing the diversity and multicast gains. *IEEE Tansactions on Information Theory*, 2013.

[19] J. Spencer, M. Sudan, and K Xu. Queueing with future information. *ArXiv Technical Report arxiv:1211.0618*, 2012.

[20] S. Zhang, L. Huang, M. Chen, and X. Liu. Proactive serving reduces user delay exponentially. *Proceedings of ACM Sigmetrics (Poster Paper)*, 2014.

[21] K. Xu. Necessity of future information in admission control. *Operations Research*, 2015.

[22] L. Huang, S. Zhang, M. Chen, and X. Liu. When Backpressure meets Predictive Scheduling. *Proceedings of ACM MobiHoc*, 2014.

[23] L. Muppirisetty, J. Tadrous, A. Eryilmaz, and H. Wymeersch. On proactive caching with demand and channel uncertainties. *Proceedings of Allerton Conference*, 2015.

[24] S. Zhao, X. Lin, and M. Chen. Peak-minimizing online ev charging: Price-of-uncertainty and algorithm robustification. *Proceedings of IEEE INFOCOM*, 2015.

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

[26] 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.

[27] 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.

[28] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew. Online algorithms for geographical load balancing. *IEEE IGCC*, 2012.

[29] J. Tadrous, A. Eryilmaz, and H. El Gamal. Pricing for demand shaping and proactive download in smart data networks. *The 2nd IEEE International Workshop on Smart Data Pricing (SDP), INFOCOM*, 2013.

[30] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong. A cost-effective recommender system for taxi drivers. *ACM KDD*, 2014.

[31] L. Huang. The value-of-information in matching with queues. *IEEE/ACM Trans. on Netwroking*, to appear.

[32] O. Simeone C. Tapparello and M. Rossi. Dynamic compression-transmission for energy-harvesting multihop networks with correlated sources. *IEEE/ACM Trans. on Networking*, 2014.

[33] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. J. Neely. Data centers power reduction: A two time scale approach for delay tolerant workloads. *IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 25, no. 1, pp. 200-211*, Jan 2014.

[34] W. Wang, K. Zhu, Lei Ying, J. Tan, and L. Zhang. Map task scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality. *IEEE/ACM Transactions on Networking*, to appear.

[35] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. Low. Real-time deferrable load control: Handling the uncertainties of renewable generation. *ACM e-Energy*, 2013.

[36] J. Walrand. *Probability in Electrical Engineering and Computer Science*. Amazon, Feb 2014.

[37] 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.

[38] 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.

[39] M. J. Neely. Energy optimal control for time-varying wireless networks. *IEEE Transactions on Information Theory 52(7): 2915-2934*, July 2006.

[40] L. Huang and M. J. Neely. Delay reduction via Lagrange multipliers in stochastic network optimization. *IEEE Trans. on Automatic Control*, 56(4):842–857, April 2011.

[41] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003.

[42] L. Huang and M. J. Neely. Max-weight achieves the exact $[O(1/V), O(V)]$ utility-delay tradeoff under Markov dynamics. *arXiv:1008.0200v1*, 2010.

[43] J. Liu, A. Eryilmaz, N. Shroff, and E. Bentley. Heavy-ball: A new approach for taming delay and convergence in wireless network optimization. *Proceedings of IEEE INFOCOM*, 2016.

[44] L. Huang and M. J. Neely. The optimality of two prices: Maximizing revenue in a stochastic network. *IEEE/ACM Transactions on Networking*, 18(2):406–419, April 2010.

[45] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6, 4-22.*, 1985.

[46] M. J. Neely. Energy-aware wireless scheduling with near optimal backlog and convergence time tradeoffs. *Proceedings of IEEE INFOCOM*, 2016.

[47] J. Liu. Achieving low-delay and fast-convergence in stochastic network optimization: A nesterovian approach. *Proceedings of ACM Sigmetrics*, 2016.

[48] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. monograph in preparation, http://www.stat.berkeley.edu/ aldous/RWG/book.html.

[49] Albert Bifet and Ricard Gavald. Learning from time-changing data with adaptive windowing. *SIAM International Conference on Data Mining*, 2007.

[50] F. Chung and L. Lu. Concentration inequalities and martingale inequalities - a survey. *Internet Math.*, 3 (2006-2007), 79–127.

[51] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association 58 (301): 13-30*, 1963.

APPENDIX A - PROOF OF LEMMA 1

(Proof of Lemma 1) We prove the performance of $\mathtt{ADE}(T_l, d_\epsilon)$ with an argument inspired by [49]. We make use of the following concentration result.

Theorem 5. *[50] Let $X_1, ..., X_n$ be independent random variables with $Pr\{X_i = 1\} = p_i$, and $Pr\{X_i = 0\} = 1 - p_i$. Consider $X = \sum_{i=1}^{n} X_i$ with expectation $\mathbb{E}\{X\} = \sum_{i=1}^{n} p_i$. Then, we have:*

$$Pr\{X \le \mathbb{E}\{X\} - m\} \le e^{\frac{-m^2}{2\mathbb{E}\{X\}}}, \tag{22}$$

$$Pr\{X \ge \mathbb{E}\{X\} + m\} \le e^{\frac{-m^2}{2(\mathbb{E}\{X\} + m/3)}}. \quad \Diamond \tag{23}$$

Proof. (Lemma 1) (**Part (a)**) In this case, it suffices to check condition (i) in $\mathtt{ADE}$. Define

$$\tilde{\pi}_i^d(t) \triangleq \frac{1}{d}\left(\sum_{\tau = (t+w-d)_+}^{t-1} 1_{[S(\tau) = s_i]} + \sum_{\tau \in \mathcal{W}_w(t)} \pi_i(\tau) \right),$$

i.e., $\tilde{\pi}_i^d(t)$ is defined with the true distributions in $\mathcal{W}_d(t)$. Denote $\epsilon_1 = (w+1)e_w/d$, we see then $\|\tilde{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^d(t)\| \le \epsilon_1$. Thus, for any $\epsilon > 0$, we have:

$$\begin{aligned}
&\Pr\{\|\hat{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} \le \epsilon\} \\
\le\ &\Pr\{\|\tilde{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} \le \epsilon + \epsilon_1\} \\
\le\ &\Pr\{|\tilde{\pi}_i^d(t) - \hat{\pi}_i^m(t)| \le \epsilon + \epsilon_1\}. \tag{24}
\end{aligned}$$

Choose $\epsilon = \frac{1}{2}\max|\pi_{1i} - \pi_{2i}| - 2\epsilon_1 > 0$ and let $\epsilon_0 = \epsilon + \epsilon_1$. Fix $\alpha \in (0,1)$ and consider $i \in \arg\max_i |\pi_{1i} - \pi_{2i}|$. We have:

$$\begin{aligned}
&\Pr\{|\tilde{\pi}_i^d(t) - \hat{\pi}_i^m(t)| \le \epsilon_0\} \\
&\le \Pr\{\{|\tilde{\pi}_i^d(t) - \pi_{1i}| \ge \alpha\epsilon_0\} \\
&\qquad \cup\{|\hat{\pi}_i^m(t) - \pi_{2i}| \ge (1-\alpha)\epsilon_0\}\} \\
&\le \Pr\{|\tilde{\pi}_i^d(t) - \pi_{1i}| \ge \alpha\epsilon_0\} \\
&\qquad + \Pr\{|\hat{\pi}_i^m(t) - \pi_{2i}| \ge (1-\alpha)\epsilon_0\}. \tag{25}
\end{aligned}$$

Here the first inequality follows because if we have both $\{|\tilde{\pi}_i^d(t) - \pi_{1i}| < \alpha\epsilon_0\}$ and $\{|\hat{\pi}_i^m(t) - \pi_{2i}| < (1-\alpha)\epsilon_0\}$, and $|\tilde{\pi}_i^d(t) - \hat{\pi}_i^m(t)| \le \epsilon_0$, we must have:

$$\begin{aligned}
|\pi_{1i} - \pi_{2i}| &\le |\tilde{\pi}_i^d(t) - \pi_{1i}| + |\hat{\pi}_i^m(t) - \pi_{2i}| + |\tilde{\pi}_i^d(t) - \hat{\pi}_i^m(t)| \\
&= 2\epsilon_0 < |\pi_{1i} - \pi_{2i}|,
\end{aligned}$$

which contradicts the fact that i achieves $\max_i |\pi_{1i} - \pi_{2i}|$. Using (25) and Hoeffding inequality [51], we first have:

$$\begin{aligned}
&\Pr\{|\hat{\pi}_i^m(t) - \pi_{2i}| \ge (1-\alpha)\epsilon_0\} \\
&\qquad \le 2\exp(-2((1-\alpha)\epsilon_0)^2 W_m(t)). \tag{26}
\end{aligned}$$

For the first term in (25), we have:

$$\begin{aligned}
&\Pr\{|\tilde{\pi}_i^d(t) - \pi_{1i}| \ge \alpha\epsilon_0\} \\
&\qquad \le 2\exp(-2(\alpha\epsilon_0)^2 (W_d(t) - w - 1)). \tag{27}
\end{aligned}$$

Equating the above two probabilities and setting the sum equal to δ, we have $\alpha = \frac{\sqrt{W_m(t)/(W_d(t) - w - 1)}}{1 + \sqrt{W_m(t)/(W_d(t) - w - 1)}}$, and

$$\epsilon_0 = \sqrt{\ln\frac{4}{\delta}} \cdot \frac{1 + \sqrt{(W_d(t) - w - 1)/W_m(t)}}{\sqrt{2(W_d(t) - w - 1)}}. \tag{28}$$

In order to detect the different distributions, we can choose $\epsilon_d < \epsilon_0$, which on the other hand requires that:

$$\begin{aligned}
\epsilon_d &\overset{(*)}{\le} \sqrt{\ln\frac{4}{\delta}} \cdot \sqrt{\frac{1}{2(d - w - 1)}} < \epsilon_0 \\
&\Rightarrow\ d > \ln\frac{4}{\delta} \cdot \frac{1}{2\epsilon_d^2} + w + 1. \tag{29}
\end{aligned}$$

Here (*) follows because $W_d(t) = d \le W_m(t)$. This shows that whenever $W_d(t) = d \le W_m(t)$ and the windows are loaded with non-coherent samples, error will be detected with probability $1 - \delta$.

(Part (b)) Note that for any time t, the distribution will be declared changed if $\|\hat{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} > \epsilon_d$. Choose $\epsilon_d = 2\epsilon_1$. Similar to the above, we have:

$$\Pr\{\|\hat{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} \ge \epsilon_d\} \tag{30}$$

$$\leq \Pr\big\{\|\tilde{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} \geq \epsilon_d - \epsilon_1\big\}$$
$$\leq \Pr\big\{\|\hat{\boldsymbol{\pi}}^d(t) - \boldsymbol{\pi}\|_{TV} \geq \alpha\epsilon_d/2\big\}$$
$$+ \Pr\big\{\|\hat{\boldsymbol{\pi}}^m(t) - \boldsymbol{\pi}\|_{TV} \geq (1-\alpha)\epsilon_d/2\big\}.$$

Using the same argument as in (25), (26) and (27), we get:

$$\Pr\big\{\|\hat{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} \geq \epsilon_d\big\} \leq \delta.$$

This shows that step (i) declares change with probability δ.

Next we show that step (ii) does not declare a distribution change with high probability. To do so, we use Theorem 5 with $m = 2\log(T_l)\sqrt{T_l}$ to have that, when $W_m(t) \geq T_l$,

$$\Pr\big\{\|\pi_i^m(t) - \pi_i\| > \frac{2\log(T_l)}{\sqrt{T_l}}\big\} \leq e^{-2\log(T_l)^2} = T_l^{-2\log(T_l)}.$$

Using the union bound, we get

$$\Pr\big\{\|\boldsymbol{\pi}^m(t) - \boldsymbol{\pi}\| > \frac{2M\log(T_l)}{\sqrt{T_l}}\big\} \leq M T_l^{-2\log(T_l)}. \quad (31)$$

Thus, part (b) follows from the union bound over k. $\qquad\square$

APPENDIX B - PROOF OF THEOREM 1

(Proof of Theorem 1) Here we prove the utility-delay performance of PLC for a stationary system. We sometimes omit the $\boldsymbol{\pi}$ when it is clear. For our analysis, define:

$$g_{s_i}(\boldsymbol{\gamma}) = \inf_{x^{(s_i)} \in \mathcal{X}_i} \Big\{ V f(s_i, x^{(s_i)}) \qquad (32)$$
$$+ \sum_j \gamma_j \big[A_j(s_i, x^{(s_i)}) - \mu_j(s_i, x^{(s_i)}) \big] \Big\},$$

to be the dual function when there is only a single state s_i. It is clear from equations (8) and (32) that:

$$g(\boldsymbol{\gamma}) = \sum_i \pi_i g_{s_i}(\boldsymbol{\gamma}). \qquad (33)$$

We will also make use of the following results.

Lemma 2. *[40] Suppose the conditions in Theorem 1 hold. Then, under* PLC *with $\boldsymbol{Q}(t) = \boldsymbol{q}(t)$, there exist constants $G, \eta_1 = \Theta(1)$, i.e., both independent of V, such that whenever $\|\boldsymbol{q}(t) - \boldsymbol{\gamma}^*\| > G$,*

$$\mathbb{E}_{\boldsymbol{\pi}}\{\|\boldsymbol{q}(t+1) - \boldsymbol{\gamma}^*\| \mid \boldsymbol{q}(t)\} \leq \|\boldsymbol{q}(t) - \boldsymbol{\gamma}^*\| - \eta_1. \quad \diamond \quad (34)$$

Lemma 3. *[16] If $\|\boldsymbol{\pi}_a(t) - \boldsymbol{\pi}\|_{TV} \leq \epsilon$ and (5) holds for $\boldsymbol{\pi}_a(t)$, then $\boldsymbol{\gamma}^*(t)$ satisfies:*

$$\|\boldsymbol{\gamma}^*(t) - \boldsymbol{\gamma}^*\| \leq b_0 V \epsilon, \qquad (35)$$

where $b_0 = \Theta(1)$. $\diamond$

Lemma 4. *[46] Suppose $Z(t)$ is a real-value random process with initial value z_0 that satisfies:*

1) $|Z(t+1) - Z(t)| \leq Z_{\max}$ *where* $Z_{\max} > 0$
2) $\mathbb{E}\{Z(t+1) - Z(t) \mid Z(t)\} \leq z(t)$ *where $z(t) = Z_{\max}$ when $Z(t) < Z_u$ and $z(t) = -\eta$ with $0 \leq \eta \leq Z_{\max}$ when $Z(t) \geq Z_u$ for some constant Z_u.*

Then, there exist constants $r_z = \Theta(1)$, $0 < \rho_z < 1$, and $D = \frac{(e^{r_z Z_{\max}} - \rho_z)e^{r_z Z_u}}{1-\rho_z}$, such that for every slot t,

$$\mathbb{E}\{e^{r_z Z(t)}\} \leq D + (e^{r_z z_0} - D)\rho_z^t. \quad \diamond \quad (36)$$

Now we prove Theorem 1.

Proof. (Theorem 1) (**Part (a) - Utility**) Define a Lyapunov function $L(t) \triangleq \frac{1}{2}\sum_j q_j(t)^2$. Then, define the one-slot Lyapunov drift $\Delta(t) \triangleq \mathbb{E}\{L(t+1) - L(t) \mid \boldsymbol{q}(t)\}$. Using the queueing dynamic equation (2), we have:

$$\Delta(t) \leq B - \sum_j q_j(t)\mathbb{E}\{\mu_j(t) - A_j(t) \mid \boldsymbol{q}(t)\}. \qquad (37)$$

Here $B \triangleq \delta_{\max}^2$, and the expectation is taken over $\boldsymbol{\pi}$ and the potential randomness in action selection. Adding to both sides the term $V\mathbb{E}\{f(t) \mid \boldsymbol{q}(t)\}$, we first obtain:

$$\Delta(t) + V\mathbb{E}\{f(t) \mid \boldsymbol{q}(t)\} \leq B \qquad (38)$$
$$+ \sum_j \mathbb{E}\{V f(t) - q_j(t)[\mu_j(t) - A_j(t)] \mid \boldsymbol{q}(t)\}.$$

Now add to both sides the term $\Delta_1(t) \triangleq \mathbb{E}\{(\gamma_j^*(t) - \theta)^+[\mu_j(t) - A_j(t)] \mid \boldsymbol{q}(t)\}$, we get:

$$\Delta(t) + V\mathbb{E}\{f(t) \mid \boldsymbol{q}(t)\} + \Delta_1(t) \qquad (39)$$
$$\leq B + \sum_j \mathbb{E}\{V f(t) + Q_j(t)[\mu_j(t) - A_j(t)] \mid \boldsymbol{q}(t)\}.$$
$$= B + g(\boldsymbol{Q}(t))$$
$$\leq B + f_{\text{av}}^{\boldsymbol{\pi}}. \qquad (40)$$

The last inequality holds as follows. Define a convexified dual function as:

$$\tilde{g}_{s_i}(\boldsymbol{\gamma}) = \inf_{x_k^{(s_i)} \in \mathcal{X}_i} \Big\{ \sum_k a_k^{(s_i)} V f(s_i, x_k^{(s_i)})$$
$$+ \sum_k a_k^{(s_i)} \sum_j \gamma_j \big[A_j(s_i, x_k^{(s_i)}) - \mu_j(s_i, x_k^{(s_i)}) \big] \Big\},$$

and $\tilde{g}(\boldsymbol{\gamma}) = \sum_i \pi_i \tilde{g}_{s_i}(\boldsymbol{\gamma})$. We have $\tilde{g}(\boldsymbol{\gamma}) = g(\boldsymbol{\gamma})$. To see this, suppose one $x_k^{(s_i)}$ maximizes the term $V f(s_i, x_k^{(s_i)}) + \sum_j \gamma_j \big[A_j(s_i, x_k^{(s_i)}) - \mu_j(s_i, x_k^{(s_i)}) \big]$. Then, the optimal choice of $\{a_k^{(s_i)}\}$ values is to set $a_k^{(s_i)} = 1$ and the other values zero, in which case we obtain $\tilde{g}(\boldsymbol{\gamma}) = g(\boldsymbol{\gamma})$. From the definition of $\tilde{g}(\boldsymbol{\gamma})$, we can exactly view the probabilities $\{a_k^{(s_i)}\}$ as specifying a stationary and randomized policy that achieves the minimum for a given $\boldsymbol{\gamma}$. Hence, the value achieved must be no larger than any other such policies, including the ones that achieve the optimal utility subject to stability, which is guaranteed to exist [1], resulting in $g(\boldsymbol{Q}(t)) \leq f_{\text{av}}^{\boldsymbol{\pi}}$.

Taking an expectation over $\boldsymbol{q}(t)$, carrying out a telescoping sum from $t = 0$ to $t = T-1$, and dividing both sides by VT, we obtain:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{f(t)\} \leq f_{\text{av}}^{\boldsymbol{\pi}} + B/V - \frac{1}{VT}\sum_{t=0}^{T-1}\mathbb{E}\{\Delta_1(t)\}. \quad (41)$$

To prove the utility performance, it remains to show that the last term is $O(1)$ in the limit, i.e.,

$$\lim_{T \to \infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_j \mathbb{E}\{(\gamma_j^*(t) - \theta)^+[\mu_j(t) - A_j(t)]\} = O(1). \quad (42)$$

To prove (42), consider the system evolution over the timeline. From the detection algorithm, we see that the timeline is divided into intervals separated by reset points. Moreover, since $\mathcal{W}_m(t)$ and $\mathcal{W}_d(t)$ are restarted, and $\boldsymbol{q}(t)$ is set to zero at reset points, these intervals form renewal cycles with initial
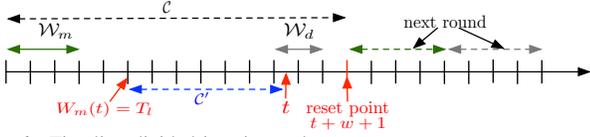
backlog $\boldsymbol{q}(t) = \boldsymbol{0}$ (see Fig. 6).

Fig. 6. Timeline divided into intervals.

Label the cycles by $\{\mathcal{C}_k, k = 0, 1, ...\}$. We thus have:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_j \mathbb{E}\{(\gamma_j^*(t) - \theta)^+ [\mu_j(t) - A_j(t)]\} \quad (43)$$

$$= \frac{\mathbb{E}\{\text{cost}\}}{\mathbb{E}\{\text{length}\}} \triangleq \frac{\mathbb{E}\{\sum_{t \in \mathcal{C}_k} \sum_j (\gamma_j^*(t) - \theta)^+ [\mu_j(t) - A_j(t)]\}}{\mathbb{E}\{|\mathcal{C}_k|\}}.$$

Below, we omit the index k and use $d_m \triangleq \max_t\{b_d^s(t) - b_m(t)\}$ to denote the size of $\mathcal{C}$. Also, let c_0 be such that $e_w = \Theta(V^{-c_0/2})$, and write $T_l = V^{c_1}$ where $c_1 \triangleq \max(c c_0)$. Since $e_w > 0$, we have $T_l < \infty$.

We first show that the probability for having a small d_m (w.r.t. T_l) is small. Denote the event $\mathcal{E}_1$ that $d_m \geq V^{2+c_1}$ and $W_m(t) = T_l$ at $t = T_l + d - w$ slots from the beginning of $\mathcal{C}$, i.e., step (i) of ADE does not declare any change before $W_m(t) = T_l$ occurs. Using Lemma 1, $T_l \geq V^c$, and the fact that for a large V, $d = \log(V)^3/\epsilon_d^2 \geq \frac{2}{\epsilon_d^2} \ln(4/\delta) + w + 1$ for $\delta = V^{-\log(V)}$, we have that:

$$\Pr\{\mathcal{E}_1^c\} \leq \delta T_l + V^{2+c_1} \cdot (\delta + (w+1)MT_l^{-2\log(T_l)})$$
$$\leq V^{-c_2 \log(V)} \leq V^{-3}. \quad (44)$$

Here $\mathcal{E}_1^c$ denotes the complementary event of $\mathcal{E}_1$. Therefore, with probability at least $1 - V^{-3}$, $d_m \geq V^{2+c_1}$, which implies that:

$$\mathbb{E}\{|\mathcal{C}|\} \geq V^{2+c_1}/2. \quad (45)$$

Conditioning on $\mathcal{E}_1$, we see that PLC will compute the empirical multiplier with statistics in $\mathcal{W}_m(t)$ after $t = T_l + d - w$ slots, and use it until a new change is declared (see Fig. 6). Denote this period of time, i.e., after the multiplier is computed until the cycle ends, by $\mathcal{C}'$ and its length by d_m' (see Fig. 6). We have $d_m' = d_m - O(V^{c_1}) - w - 1 = \Omega(V^{2+c_1})$ time (the first V^{c_1} slots are for learning and the last $O(w+1)$ slots are not contained in both $\mathcal{W}_m(t)$ and $\mathcal{W}_d(t)$ due to ADE).

Denote another event $\mathcal{E}_2 \triangleq \{\|\hat{\boldsymbol{\pi}}^m(t) - \boldsymbol{\pi}\| \leq \frac{4\log(V)}{V^{c_1/2}}\}$, where t is when $W_m(t) = T_l$. That is, the distribution $\hat{\boldsymbol{\pi}}^m(t)$ is close to the true distribution, for $t \in \mathcal{C}'$ (Note that $\hat{\boldsymbol{\pi}}^m(t)$ remains constant during $\mathcal{C}'$). Using Theorem 5, we have that:

$$\Pr\{\mathcal{E}_2\} \geq 1 - Me^{-4\log(V)^2}. \quad (46)$$

Thus,

$$\Pr\{\mathcal{E}_2 | \mathcal{E}_1\} = \frac{\Pr\{\mathcal{E}_2 \cap \mathcal{E}_1\}}{\Pr\{\mathcal{E}_1\}} \geq 1 - Me^{-4\log(V)^2} - V^{-3}$$
$$\geq 1 - 2V^{-3}. \quad (47)$$

With $\mathcal{E}_1$ and $\mathcal{E}_2$, we now bound $\mathbb{E}\{\text{cost}\}$, where cost $\triangleq \sum_{t \in \mathcal{C}} \sum_j (\gamma_j^*(t) - \theta)^+ [\mu_j(t) - A_j(t)]$. First, when $\mathcal{E}_1^c$ takes place, we either have $d_m \leq V^{2+c_1}$, denoted by $\mathcal{E}_{1a}^c$, or $d_m \geq V^{2+c_1}$ but step (i) of ADE declares changes before $W_m(t) = T_l$, denoted by $\mathcal{E}_{1b}^c$. Given $\mathcal{E}_{1a}^c$, the cost is no more

than $V \log(V) \delta_{\max} V^{2+c_1}$. For $\mathcal{E}_{1b}^c$, we first see that:

$$\Pr\{\mathcal{E}_{1b}^c\}$$
$$= \Pr\{d_m \geq V^{2+c_1} \text{and}$$
at least one change declared in the first $T_l + d - w$ slots$\}$
$$\leq \delta(T_l + d - w) \leq V^{-\log(V)/2}. \quad (48)$$

Also, given $\mathcal{E}_{1b}^c$, if we denote the first time a change is declared by T_{1b}, we have:

$$\mathbb{E}\{|\mathcal{C}| \, | \, \mathcal{E}_{1b}^c\} \leq T_{1b} + \mathbb{E}\{|\mathcal{C}| \, | \, d_m \geq V^{2+c_1} - T_{1b}\}$$
$$\leq T_l + d + 2\mathbb{E}\{|\mathcal{C}|\} \leq 3\mathbb{E}\{|\mathcal{C}|\}. \quad (49)$$

The first step follows because after the first declaration, the requirement for any additional declaration is removed, and the second step follows because $T_{1b} \leq T_l + d - w$ and $\mathbb{E}\{|\mathcal{C}| \, | \, d_m \geq V^{2+c_1} - T_{1b}\} \leq 2\mathbb{E}\{|\mathcal{C}|\}$. Thus,

$$\mathbb{E}\{\text{cost} | \mathcal{E}_1^c\} \quad (50)$$
$$\leq V \log(V) \delta_{\max} \cdot \left(\frac{V^{2+c_1}}{V^{c_2 \log(V)}} + \frac{3\mathbb{E}\{|\mathcal{C}|\}}{V^{\log(V)/2}}\right).$$

Here we have used Lemma 1 in [40] and the learning step in PLC that ensures $\boldsymbol{\gamma}^*(t) = O(V \log(V))$. On the other hand,

$$\mathbb{E}\{\text{cost} | \mathcal{E}_1, \mathcal{E}_2^c\} \leq V \log(V) \delta_{\max} \mathbb{E}\{|\mathcal{C}| \, | \, \mathcal{E}_1, \mathcal{E}_2^c\}. \quad (51)$$

Let us now bound $\mathbb{E}\{|\mathcal{C}| \, | \, \mathcal{E}_1, \mathcal{E}_2^c\}$. Define $\mathcal{E}_{2a}^c = \{y \in (\sigma, 2\sigma]\}$ and $\mathcal{E}_{2b}^c = \{y > 2\sigma\}$, where $y \triangleq \|\hat{\boldsymbol{\pi}}^m(t) - \boldsymbol{\pi}\|$ and $\sigma \triangleq \frac{4\log(V)}{V^{c_1/2}}$. We have:

$$\Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_2^c\} \quad (52)$$
$$= \Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_{2a}^c\} \Pr\{\mathcal{E}_{2a}^c | \mathcal{E}_1, \mathcal{E}_2^c\}$$
$$+ \Pr\{\|\hat{\boldsymbol{\pi}}^m - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_{2b}^c\} \Pr\{\mathcal{E}_{2b}^c | \mathcal{E}_1, \mathcal{E}_2^c\}.$$

Let us relate $\Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_{2a}^c\}$ to $\Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_2\}$. Consider a $\hat{\boldsymbol{\pi}}^d(t)$ is such that $\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d$ given $\mathcal{E}_1, \mathcal{E}_2$. We note that there exist i and j such that $\pi_i^d(t) \leq \pi_i$ and $\pi_j^d(t) \geq \pi_j$. Then, we can always change $\hat{\boldsymbol{\pi}}^d(t)$ to $\tilde{\boldsymbol{\pi}}^d(t)$ by having one more sample for j and one less sample for i (this can be ensured with high probability since $d = O(\log^3(V))$). Since $\sigma = O(V^{-c_1/2})$ and $\epsilon_d = O(1/\log^3(V))$, we will have $\|\hat{\boldsymbol{\pi}}^m(t) - \tilde{\boldsymbol{\pi}}^d(t)\| > \epsilon_d$ given $\mathcal{E}_1, \mathcal{E}_{2a}^c$. Therefore,

$$\Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_{2a}^c\}$$
$$\geq P_0 \triangleq c_0 \Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \boldsymbol{\pi}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_2\}.$$

Here $c_0 = \min_i \pi_i / \max_j \pi_j$. This shows that the probability of having a change declared under $\mathcal{E}_1, \mathcal{E}_{2a}^c$ is more than a constant factor of that under $\mathcal{E}_1, \mathcal{E}_2$. As a result, using (52) and the fact that $\Pr\{\mathcal{E}_{2a}^c | \mathcal{E}_1, \mathcal{E}_2^c\} \geq 1 - O(V^{-3})$,

$$\Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d \, | \, \mathcal{E}_1, \mathcal{E}_2^c\} \geq P_1,$$

where $P_1 = c_1 P_0$ and $c_1 \geq c_0(1 - O(V^{-3}))$. Thus,

$$\mathbb{E}\{|\mathcal{C}| \, | \, \mathcal{E}_1, \mathcal{E}_2^c\} \leq d/P_1. \quad (53)$$

This is obtained by considering only testing for changes at multiples of d slots.

On the other hand, it can be shown that

$$\mathbb{E}\{|\mathcal{C}| \, | \, \mathcal{E}_1, \mathcal{E}_2\} \geq \Theta(1/P_1). \quad (54)$$

This is so since, conditioning on $\mathcal{E}_1, \mathcal{E}_2$, samples in $\mathcal{W}_d(t)$

evolves according to a Markov chain, with each state being a sequence of d samples. Moreover, the total mass of the set of states resulting in $\|\hat{\boldsymbol{\pi}}^m(t) - \hat{\boldsymbol{\pi}}^d(t)\| > \epsilon_d$ is P_0/c_0 and that after V^{2+c_1} time, the first $\mathcal{W}_d(t)$ is drawn with the steady state probability (due to $S(t)$ being i.i.d.). Thus, the Markov chain is in steady state from then, showing that the time it takes to hit a violating state is $\Theta(1/P_1)$. Combining (54) with (53), we conclude that:

$$\mathbb{E}\{|\mathcal{C}| \,|\, \mathcal{E}_1, \mathcal{E}_2^c\} \leq d\mathbb{E}\{|\mathcal{C}| \,|\, \mathcal{E}_1, \mathcal{E}_2\} \leq 2d\mathbb{E}\{|\mathcal{C}|\}. \quad (55)$$

The last inequality follows since $\Pr\{\mathcal{E}_1, \mathcal{E}_2\} \geq 1 - 2V^{-3}$.

Now consider the event $\mathcal{E}_1 \cap \mathcal{E}_2$. Using the fact that $T_l = \Theta(V^{c_1})$, $\Pr\{\mathcal{E}_2 \cap \mathcal{E}_1\} \geq 1 - O(V^{-3})$, and using almost verbatim arguments as in the proofs of Lemmas 8 and 9 in [17], it can be shown that:[12]

$$\mathbb{E}\Big\{\sum_{t \in \mathcal{C}'} [\mu_j(t) - A_j(t)] \,|\, \mathcal{E}_1, \mathcal{E}_2\Big\} \quad (56)$$
$$\leq \mathbb{E}\{q_{js} - q_{je} \,|\, \mathcal{E}_1, \mathcal{E}_2\} + \delta_{\max}(1 + b_1\mathbb{E}\{|\mathcal{C}| \,|\, \mathcal{E}_1, \mathcal{E}_2\}/V^{\log V}),$$

where $b_1 = \Theta(1)$, and q_{js} and q_{je} denote the beginning and ending sizes of queue j during $\mathcal{C}'$, respectively.

We first bound the $\mathbb{E}\{q_{js}\}$. Conditioning on $\mathcal{E}_1$, we see that there will be $T_l + d - w$ time until $W_m(t) = T_l$. Thus, $\mathbb{E}\{q_{js}\} \leq \delta_{\max} b_2(V^{c_1} + d - w)$ for some $b_2 = \Theta(1)$. Combining (50), (51), and (56), we obtain:

$$\mathbb{E}\{\text{cost}\}$$
$$\leq V\log(V)\delta_{\max} \cdot \Big(\frac{V^{2+c_1}}{V^{c_2\log(V)}} + \frac{3\mathbb{E}\{|\mathcal{C}|\}}{V^{\log(V)/2}}\Big) \quad (57)$$
$$+ V\log(V)\delta_{\max}\mathbb{E}\{|\mathcal{C}| \,|\, \mathcal{E}_1, \mathcal{E}_2^c\} \cdot Me^{-4\log(V)^2}$$
$$+ (\delta_{\max}b_2(V^{c_1} + d - w) + w + 1)\delta_{\max}V\log(V)$$
$$+ V\log(V)\delta_{\max}(1 + b_1\mathbb{E}\{|\mathcal{C}| \,|\, \mathcal{E}_1, \mathcal{E}_2\}/V^{\log V}).$$

The term $(w+1)\delta_{\max}V\log(V)$ in the last $w+1$ slots after a change detection. Combining (57) with (43), (45), and (55), we obtain (42).

(Part (b) - Delay) From the above, we see that the event $\mathcal{E}_1 \cap \mathcal{E}_2$ happens with probability at least $1 - O(1/V^3)$. Hence, we only need to show that most packets that arrive during the $\mathcal{C}'$ intervals experience small delay, conditioning on $\mathcal{E}_1 \cap \mathcal{E}_2$.

Denote t_s and t_e the beginning and ending slots of $\mathcal{C}'$. Using (47) and Lemma 3, we get that with probability at least $1 - 2V^{-3}$,

$$\|\boldsymbol{\gamma}^*(t) - \boldsymbol{\gamma}^*\| \leq d_\gamma \triangleq 4b_0 V^{1-c_1/2}\log(V). \quad (58)$$

Define

$$\hat{\boldsymbol{\theta}} \triangleq \boldsymbol{\gamma}^* - (\boldsymbol{\gamma}^*(t) - \boldsymbol{\theta})_+, \quad (59)$$

we see from Lemma 2 that whenever $\|\boldsymbol{q}(t) - \hat{\boldsymbol{\theta}}\| > G$, which is equivalent to $\|\boldsymbol{Q}(t) - \boldsymbol{\gamma}^*\| > G$,

$$\mathbb{E}\{\|\boldsymbol{q}(t+1) - \hat{\boldsymbol{\theta}}\| \,|\, \boldsymbol{q}(t)\} \leq \|\boldsymbol{q}(t) - \hat{\boldsymbol{\theta}}\| - \eta,$$

for the same $G = \Theta(1)$ and $\eta = \Theta(1) < \eta_1$ in Lemma 2.[13] Using (58) and θ in (16), we see that $\hat{\boldsymbol{\theta}} = \Theta(d_\gamma \log(V) + \log(V)^2)$. Therefore, using Theorem 4 in [16], if we assume

---

[12] The fact that the event holds with probability almost 1 enables an analysis similar to that without conditioning.

[13] This is due to conditioning on $\mathcal{E}_1 \cap \mathcal{E}_2$.

that $\mathcal{C}'$ never ends,

$$\mathbb{E}\{T_G(\boldsymbol{q}(t))\} \leq b_3 d_q/\eta, \quad (60)$$

where $b_3 = \Theta(1)$, $d_q = \|\hat{\boldsymbol{\theta}} - \boldsymbol{q}(t_s)\|$ and $T_G(\boldsymbol{q}(t)) \triangleq \inf\{t - t_s : \|\boldsymbol{q}(t) - \hat{\boldsymbol{\theta}}\| \leq G\}$. Note that this is after $W_m(t) = T_l$ in PLC, which happens after $T_c = d - w + T_l$ slots from the beginning of the interval. By Markov inequality,

$$\Pr\{T_G(\boldsymbol{q}(t)) + T_c > (b_3 d_q/\eta + d - w + T_l)V\} = \frac{1}{V}. \quad (61)$$

Denote $\mathcal{E}_3 \triangleq \{T_G(\boldsymbol{q}(t)) + T_c(t) \leq (b_3 d_q + d - w + T_l)V\}$ and let t^* the first time after t_s that $Y(t) \triangleq \|\boldsymbol{q}(t) - \hat{\boldsymbol{\theta}}\| \leq G$. Following an argument almost identical to the proof of Theorem 1 in [40], we obtain that:

$$\sum_{t=t^*}^{t_e} \frac{\nu\eta}{2}\mathbb{E}\{e^{\nu Y(t)}\} \leq (t_e - t^*)e^{2\nu\sqrt{r}\delta_{\max}} + e^{\nu Y(t^*)}, \quad (62)$$

where $\nu \triangleq \frac{\eta}{\delta_{\max}^2 + \delta_{\max}\eta/3} = \Theta(1)$. Define $b_4 \triangleq 2e^{2\nu\sqrt{r}\delta_{\max}}/\nu\eta = \Theta(1)$ and $b_5 \triangleq e^{\nu Y(t^*)} \leq e^{\nu G} = \Theta(1)$, and choose $m = \log(V)^2$. We have from (62) that:

$$\frac{1}{t_e - t_s}\sum_{t=t_s}^{t_e} \Pr\{Y(t) > G + m\} \quad (63)$$
$$\leq \frac{1}{t_e - t_s}\Big(\sum_{t=t^*}^{t_e}\Pr\{Y(t) > G + m\} + (t^* - 1 - t_s)\Big)$$
$$\leq \big[(b_4(t_e - t^*))V^{-\log(V)} + b_5 + (t^* - t_s)\big]/(t_e - t_s)$$
$$= O\Big(\frac{(b_3 d_q + d - w + T_l)V + 1}{V^{2+c_1}}\Big) = O(1/V).$$

Thus, the above implies that, given the joint event $\mathcal{E}_1 \cap \mathcal{E}_2$, which happens with probability $1 - O(1/V^3)$, the fraction of packets enter and depart from each $q_j(t)$ when $\|\boldsymbol{q}(t) - \hat{\boldsymbol{\theta}}\| \leq G$ is given by $(1 - O(1/V))(1 - O(1/V))$, i.e., $1 - O(\frac{1}{V})$. This means that they enter and depart when $q_j(t) \in [\hat{\theta} - G - \log(V)^2, \hat{\theta} + G + \log(V)^2]$ (due to LIFO), which implies that their average delay in the queue is $O(\log(V)^2)$.

(Part (c) - Dropping) First, conditioning on $\mathcal{E}_{1a}^c$, which happens probability $V^{-\log(V)/2}$, we see that the algorithm drops at most $O(V^{2+c_1})$ packets in this case.

Now consider when $\mathcal{E}_1$ takes place, and denote as above by t_s and t_e the starting and ending timeslots of a cycle. In this case, from the rules of ADE, we see that rule (ii) is inactive, since if it is satisfied at time T_l, it remains so because $\hat{\boldsymbol{\pi}}^m(t)$ remains unchanged until the cycle ends. Hence, the only case when an interval ends is due to violation rule (i). Let us suppose the interval ends because at some time t', we have $\|\hat{\boldsymbol{\pi}}^m(t') - \hat{\boldsymbol{\pi}}^d(t')\| > \epsilon_d$. We know then PLC drops all packets at time $t' + w + 1$, i.e., $\boldsymbol{q}(t' + w + 1)$.

We now bound $\mathbb{E}\{\boldsymbol{q}(t' + w + 1)\}$. To do so, consider the time $t^* = t' - 2d$. We see then $\boldsymbol{q}(t^*)$ and all queue sizes before t^* are independent of $\hat{\boldsymbol{\pi}}^d(t')$. Also, $\sum_j q_j(t' + w + 1) \leq \sum_j q_j(t^*) + r(2d + w + 1)\delta_{\max}$.

Consider the time interval from when $W_m(t) = T_l$ untill t^* and consider two cases, (i) $e_w = \Omega(V^{-c/2})$ and (ii) $e_w = O(V^{-c/2})$. In the first case, we see that $T_l = V^c$. Thus, $q_j(t_s + T_l) \leq \delta_{\max}V^c$.

In the second case, since $e_w = O(V^{-c/2})$, $T_l = e_w^{-2}$. We have from Lemma 3 that before time T_l, the estimated multi-

plier $\|\boldsymbol{\gamma}^*(t) - \boldsymbol{\gamma}^*\| \leq Ve_w = O(V^{1-c/2})$. As a result, using the definition of $\hat{\boldsymbol{\theta}}$ in (59) and denoting $Z(t) = \|(\boldsymbol{q}(t) - \hat{\boldsymbol{\theta}})_+\|$, we see that whenever $Z(t) \geq G$, $\mathbb{E}\{|Z(t+1) - Z(t)| \mid Z(t)\} \leq -\eta$. It can also be checked that the other conditions in Lemma 4 are satisfied by $Z(t)$, and $\boldsymbol{q}(t_s) = \boldsymbol{0}$ and $Z(0) = 0$. Thus,

$$\mathbb{E}\{Z(T_l)\} \leq G + \sqrt{\boldsymbol{\delta}} \,_{\max} + O(1). \qquad (64)$$

Thus, $\mathbb{E}\{\boldsymbol{q}(t_s + T_l)\} = O(V^{1-c_1/2})$. Combining the two cases, we have $\mathbb{E}\{\boldsymbol{q}(t_s + T_l)\} = O(V^{1-c_1/2} + V^c) = O(V)$.

After $t_s + T_l$, the distribution $\hat{\boldsymbol{\pi}}^m(t)$ is used to compute the multiplier. Since $T_l = \max(V^c, e_w^{-2})$, we see that the argument above similarly holds. Thus, using Lemma 4, we see that $\mathbb{E}\{\boldsymbol{q}(t^*)\} = O(V)$, which implies $\mathbb{E}\{\boldsymbol{q}(t' + w + 1)\} = O(V + d)$. Therefore, packets will be dropped no more than every V^{2+c_1} slots, and at every time we drop no more than $O(V)$ packets on average.

Finally, consider given $\mathcal{E}_{1b}^c$. Using (45) and (49), we note that conditioning on $\mathcal{E}_{1b}^c$, the cycle lasts no more than $3\mathbb{E}\{|\mathcal{C}|\}$ on average, which means that the number of packets dropped is at most $O(\mathbb{E}\{|\mathcal{C}|\})$ every cycle on average. Moreover, using (48), we see that this happens with probability $O(V^{-3})$.

The result follows by combining the above cases. $\qquad\square$

APPENDIX C - PROOF OF THEOREM 2

In the following proof, we will refer to some steps in Appendix B, to avoid restating the steps.

(Proof of Theorem 2) We first have the following lemma to show that if each $d_k \geq 4d$, then ADE keeps only $o(d)$ samples (timeslots) from the previous distribution in $\mathcal{W}_m(t)$ after change detection. This step is important, as if $\mathcal{W}_m(t)$ contains too many samples from a previous distribution interval, the distribution estimation $\hat{\boldsymbol{\pi}}^m(t)$ can be inaccurate and lead to a high false-negative rate, which in turn affects performance during $\mathcal{I}_k$. The proof of the lemma is given at the end of this section.

Lemma 5. *Under the conditions of Theorem 2, with probability $1 - O(V^{-3\log(V)/4})$ only $o(d)$ samples from $\mathcal{I}_{k-1}$ remain in $\mathcal{W}_m(t) \cup \mathcal{W}_d(t)$ for $t \geq t_k + d$.* $\Diamond$

We now prove Theorem 2.

Proof. (Theorem 2) We first have from Lemma 1 that with probability at least $1 - V^{-3\log(V)/4}$ ($\delta = V^{-3\log(V)/4}$), distribution change will be detected before $t_k + d - w$. Denote this event by $\mathcal{E}_4$.
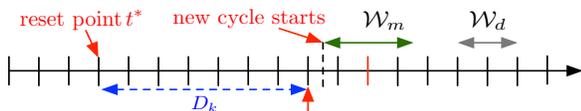

Fig. 7. Intervals in a non-stationary system.

(Part (a) - Utility) Using Lemma 5, we see that $o(d)$ samples will remain in $\mathcal{W}_m(t)$. This implies that when V is large and $\mathcal{W}_m(t) = d$, with probability $1 - O(V^{-\log(V)/2})$,

$$|\hat{\pi}_i^m(t) - \pi_{ki}| \leq \epsilon_d/8, \; \forall i \qquad (65)$$

where $\hat{\boldsymbol{\pi}}^m(t)$ is the distribution in window $\mathcal{W}_m(t)$ (can contain timeslots from the previous interval). This shows that the

empirical distribution of $\mathcal{W}_m(t)$ is close to the true distribution even though it may contain samples from $\mathcal{I}_{k-1}$. Thus, as $W_m(t)$ increases, $\hat{\boldsymbol{\pi}}^m(t)$ will only become closer to $\boldsymbol{\pi}_k$, so that (65) holds whenever $\mathcal{W}_d(t) \subset \mathcal{I}_k$. Denote (65) event $\mathcal{E}_5$.

Now use an argument similar to the proof of Lemma 1, we can show that:

$$\Pr\{\|\hat{\boldsymbol{\pi}}^d(t) - \hat{\boldsymbol{\pi}}^m(t)\|_{TV} \geq \epsilon_d\} \leq V^{-\log(V)/3}.$$

Hence, for each cycle $\mathcal{C} \subset \mathcal{I}_k$, if we denote $\mathcal{E}_6$ the event that ADE does not declare any distribution change in steps (i) and (ii) for $V^{1+a}T_l \log(V)$ slots, and $\mathcal{E}_2$ before equation (46) holds (note that (46) only requires that $W_m(t) = T_l$), we see that

$$\Pr\{\mathcal{E}_6\} \geq 1 - V^{-2}. \qquad (66)$$

This implies that $\mathcal{I}_k$ mostly likely only contains one cycle $\mathcal{C}$.

Therefore, conditioning on $\mathcal{E}_4 \cap \mathcal{E}_5 \cap \mathcal{E}_6$, which happens with probability $1 - O(V^{-2})$ and implies that for cycle $\mathcal{C}'$, $\boldsymbol{q}(t_s) = \Theta(D_k + T_l + d - w)$, we have:

$$\mathbb{E}\{\text{cost}\} \leq r(D_k + T_l + d - w)b_2\delta_{\max}V\log(V)$$
$$+ V\log(V)\delta_{\max}(1 + b_1\mathbb{E}\{|\mathcal{C}|\}/V^{\log V}).$$

Applying the argument in the proof of Theorem 1, we see that $\frac{1}{d_k}\sum_{t=0}^{d_k-1}\mathbb{E}\{\Delta_1(t)\} = O(\frac{D_k\log(V)}{T_l V^a})$. Hence, the result follows.

(Part (b) - Queue) From the above, we see that at time t_k, $\boldsymbol{q}(t_k) = O(D_k)$. We also know that the current cycle $\mathcal{C}$ will start no later than $t_k + d - w$ with probability $1 - O(V^{-3\log(V)/4})$, in which case $\boldsymbol{q}(t_s) = O(D_k + d - w)$.

Since the system is polyhedral with ρ, using an argument similar to the proof for Part (c) of Theorem 1 (which is possible since the argument there applies to finite intervals), if we define $\tilde{\boldsymbol{\theta}} = \Theta((\min(V^{1-c/2}, Ve_w)+1)\log^2(V) + D_k + d - w)$ and $Z(t) = \|(\boldsymbol{q}(t) - \tilde{\boldsymbol{\theta}})_+\|$, then throughout $t \in [t_s, t_{k+1} - 1]$,

$$\mathbb{E}\{Z(t)\} \leq G + \sqrt{\boldsymbol{\delta}} \,_{\max} + O(1). \qquad (67)$$

Therefore, Part (b) follows. $\qquad\square$

Here we provide the proof for Lemma 5.

Proof. (Lemma 5) Consider time $t = t_k - w$. We have the following cases.

(i) $W_d(t) = d$ and $W_m(t) < d$. Since $d_{k-1} \geq 4d$, we see that the change point t_k will be detected with probability at least $1 - \delta$ at time $t' \leq t + d$, because $\mathcal{W}_d(t')$ will contain samples from $\boldsymbol{\pi}_k$ while $\mathcal{W}_m(t')$ will contain samples from $\boldsymbol{\pi}_{k-1}$ (Note that although this is conditioning on $W_d(t) = d$ and $W_m(t) < d$, since at this point no statistical comparison will be assumed, it is independent of the realizations in the two windows). Moreover, all samples from $\mathcal{I}_{k-1}$ will be removed and not remain in $W_m(t)$ and $W_d(t)$, while at most $w + 1$ samples from $\mathcal{I}_k$ will be discarded.

(ii) $W_d(t) = d$ and $W_m(t) \geq d$. In this case, if a change is declared, then we turn to case (iii). Otherwise, since the samples in $\mathcal{W}_m(t)$ are drawn from $\boldsymbol{\pi}_{k-1}$, we have:

$$\Pr\{\|\hat{\boldsymbol{\pi}}^m(t) - \boldsymbol{\pi}_{k-1}\| \leq \epsilon_d/2\} \geq 1 - V^{-3\log(V)/4}. \qquad (68)$$

Now suppose no change is detected till time $t + d$. Then $W_m(t + d) \geq d$. Denote $\mathcal{E}_7 \triangleq \{\|\hat{\boldsymbol{\pi}}^m(t) - \boldsymbol{\pi}_{k-1}\| \leq \epsilon_d/2\}$.

Conditioning on $\mathcal{E}_7$ and using (68), we have:

$$\Pr\{\|\hat{\boldsymbol{\pi}}^m(t+d) - \boldsymbol{\pi}_{k-1}\| \le \frac{\epsilon_d}{2} \,|\, \mathcal{E}_7\} \ge 1 - 2V^{-3\log(V)/4}. \quad (69)$$

The inequality follows since $\Pr\{\mathcal{E}_7\} \ge 1 - V^{-3\log(V)/4}$. Now $\mathcal{W}_d(t+d)$ contains only samples from $\boldsymbol{\pi}_k$, in which case we similarly have:

$$\Pr\{\|\hat{\boldsymbol{\pi}}^d(t+d) - \boldsymbol{\pi}_k\| \le \epsilon_d/2\} \le 1 - V^{-3\log(V)/4}. \quad (70)$$

Since the state realizations in (69) and (70) are independent, we conclude that with probability $1 - 3V^{-3\log(V)/4}$, a change will be declared before t_k and all samples from $\mathcal{I}_{k-1}$ will be removed and not remain in $\mathcal{W}_m(t) \cup \mathcal{W}_d(t)$.

(iii) $W_d(t) < d$. We argue that with high probability, at most $o(d)$ samples can remain at time $t_k + 2d - W_d(t)$. First, note that $W_d(t) < d$ only occurs when a detection has been declared at a time $t + w - d \le t' \le t$. Thus, if $t + w - t' = o(d)$, then we are done. Otherwise suppose $t + w - t' = \alpha d$ for $\alpha = \Theta(1)$. If they are removed, then at time $t' + 2d$, $\mathcal{W}_m(t' + 2d)$ contains samples with mixed distribution $\boldsymbol{\pi}' = \alpha\boldsymbol{\pi}_{k-1} + (1-\alpha)\boldsymbol{\pi}_k$ and $\mathcal{W}_d(t' + 2d)$ containing samples with distribution $\boldsymbol{\pi}_k \ne \boldsymbol{\pi}'$. Similar to case (i), the condition $W_d(t) < d$ is independent of the state realizations in the two windows. Using Lemma 1 (it can be checked that the conditions in the lemma are satisfied), we see that this will be detected by ADE with probability $1 - \delta$ with a large V.

Combining all three cases completes the proof. $\qquad \square$

APPENDIX D - PROOF OF THEOREM 3

(Proof of Theorem 3) We prove Theorem 3 here.

Proof. (**Part (a) - Stationary**) The results follow from the fact that when $e_w = 0$, PLC is equivalent to OLAC in [16] with perfect statistics. Hence the results follow from Theorems 1 and 2 in [16].

(**Part (b) - Non-Stationary**) We first see that at time t, ADE detects distribution change in time $t + w$ through step (ii) with probability 1. Then, after time $t_k + d - w$, $\boldsymbol{\pi}_a(t) = \boldsymbol{\pi}_k$ and we see that whenever $Z(t) \triangleq \|\boldsymbol{q}(t) - \boldsymbol{\theta}\| > G$ for $\theta = 2\log^2(V)$ and $G = \Theta(1)$,

$$\mathbb{E}\{Z(t+1) \,|\, \boldsymbol{q}(t)\} \le Z(t) - \eta. \quad (71)$$

Denote $b_6 = \frac{1}{r_z}\log(e^{r_z r\delta_{\max} - \rho_z}/(1 - \rho_z))$. We want to show via induction that for all k,

$$\mathbb{E}\{\sum_j q_j(t_k)\} \le q_{th} \triangleq 2r\log^2(V) + b_6 + 2G + \delta_{\max}. \quad (72)$$

First, it holds for time zero. Suppose it holds for interval $\mathcal{I}_k$. We now show that it also holds for interval $k+1$.

To do so, first we see that during time $[t_k, t_k + d - w]$, there can be an increment of $q_j(t)$ since $\boldsymbol{\pi}_a(t)$ during this interval is a mixed version of $\boldsymbol{\pi}_{k-1}$ and $\boldsymbol{\pi}_k$. Thus,

$$\mathbb{E}\{\sum_j q_j(t_k + d)\} \le q'_{th} \triangleq q_{th} + \delta_{\max}. \quad (73)$$

Using Lemma 4, we have:

$$\mathbb{E}\{e^{r_z Z(t_{k+1} - d)}\}$$
$$\le \frac{e^{r_z r\delta_{\max} - \rho_z}}{1 - \rho_z} e^{r_z G} + (e^{r_z q'_{th}} - b_6 e^{r_z G})\rho_z^{d_k - 2d}.$$

Using the definition of q_{th} and the fact that $d_k \ge d\log^2(V)$, we have that for a large V, $(e^{r_z q_{th}} - b_6 e^{r_z G})\rho^{d_k - 2d} \le G$. Thus,

$$\mathbb{E}\{Z(t_{k+1} - d)\} \le b_6 + 2G, \quad (74)$$

which implies $\mathbb{E}\{\sum_j q_j(t_{k+1} - d)\} \le 2r\log^2(V) + b_6 + 2G$. It thus follows that $\mathbb{E}\{\sum_j q_j(t_{k+1} - d)\} \le q_{th} \le b_7 \log^4(V)$ for some $b_7 = \Theta(1)$.

Having established this result, using an argument similar to that in the proof of Theorem 2, we have:

$$\mathbb{E}\{\text{cost}\} \le b_7 \log^4(V) \cdot V\log(V)$$
$$+ V\log(V)\delta_{\max}(1 + b_1\mathbb{E}\{|\mathcal{C}|\}/V^{\log V}).$$

Using $d_k \ge V^{1+\epsilon}$, we see that Part (b) follows. $\qquad \square$

APPENDIX E - PROOF OF THEOREM 4

(Proof of Theorem 4) Here we prove the convergence results. We sometimes drop the subscript k when it is clear.

Proof. (Theorem 4) First, when $e_w = 0$, we see that for any interval $\mathcal{I}_k$, for all time $t \ge t_k + d$, $\boldsymbol{\pi}_a(t) = \boldsymbol{\pi}_k$, and $\boldsymbol{\gamma}^*(t) = \boldsymbol{\gamma} - \boldsymbol{\theta}$. Using Lemma 5 in [16] and the fact that $d = O(\log^2(V))$, we have:

$$\mathbb{E}\{T_G\} = \mathbb{E}\{\mathbb{E}\{T_G \,|\, \boldsymbol{q}(t_k)\}\}$$
$$\overset{(*)}{=} \mathbb{E}\{\Theta(\|\boldsymbol{q}(t_k) - \boldsymbol{\theta}\|)\} \overset{(**)}{=} \Theta(\log^4(V)).$$

Here (*) follows from Lemma 5 in [16] and (**) follows from (72).

Consider the other case $e_w > 0$. Using Lemma 5, we see that with probability at least $1 - V^{-3}$, PLC detects distribution change before time $t_k + d$. Recall the event $\mathcal{E}_1$ that ADE does not declare change in the first V^{2+c_1} slots from the proof of Theorem 1, where c_1 is such that $T_l = V^{c_1}$. Note that this implies $\{d_m \ge V^{2+c_1}\}$). From (44), we know that:

$$\Pr\{\mathcal{E}_1\} \ge 1 - V^{-3}. \quad (75)$$

Conditioning on $\mathcal{E}_1$, the time it takes to achieve $\|\boldsymbol{Q}(t) - \boldsymbol{\gamma}^*\| \le G$ is no more than the sum of (i) the time it takes to reach $W_m(t) = T_l$, and (ii) the time it takes to go from the estimated multiplier $\boldsymbol{\gamma}^*(t) - \boldsymbol{\theta}$ to $\boldsymbol{\gamma}^*$. Denote $\mathcal{E}_8(t) = \{\|\boldsymbol{\pi}^m(t) - \boldsymbol{\pi}\|_{TV} \le 2M\log(T_l)T_l^{-1/2}\}$. When $W_m(t) = T_l$, we have

$$\Pr\{\mathcal{E}_8(t)\} \ge 1 - O(MT_l^{-2\log(T_l)}), \quad (76)$$

in which case $\|\boldsymbol{\gamma}^*(t) - \boldsymbol{\gamma}^*\| = \Theta(\frac{V\log(V)}{\sqrt{T_l}})$. As in the proof of Theorem 2, we see that when $W_m(t) = T_l$, $\boldsymbol{q}(t) = O(D_k + T_l + d)$, which implies that $\|\boldsymbol{Q}(t) - \boldsymbol{\gamma}^*\| = \Theta((1 + \frac{V}{\sqrt{T_l}})\log^2(V) + T_l + D_k + d)$. Using Lemma 5 in [16] again, we see that if ADE always does not declare change,

$$\mathbb{E}\{T_G\} = O(\theta + T_l + D_k + d). \quad (77)$$

Using Markov inequality, we see that:

$$\Pr\{T_G \ge V^{2+c_1}\} \le O(V^{-1-c_1} + D_k V^{-2-c_1}). \quad (78)$$

Thus, with probability $1 - O(V^{-1-c_1} + D_k V^{-2-c_1})$, convergence occurs before V^{2+c_1}. This proves (19).

To prove (20), define $G_1 = \Theta(D_k + 2\log(V)^2(1 + Ve_w))$. Then, we see from Lemma 5 that with probability $1 - O(V^{-3\log(V)/4})$, distribution change will be detected before

$t' \leq t_k + d$. At that time, we have $\|\boldsymbol{\gamma}^*(t) - \boldsymbol{\gamma}^*\| = O(Ve_w)$. Combining this with the fact that $\boldsymbol{q}(t') = O(D_k + d)$, we see that (20) follows. This completes the proof. $\qquad\square$