Tracking Drift-Plus-Penalty: Utility Maximization for Partially Observable and Controllable Networks

Bai Liu, Quang Minh Nguyen, *Graduate Student Member, IEEE*, Qingkai Liang, and Eytan Modiano, *Fellow, IEEE*

***Abstract*— Stochastic network models with all components being observable and controllable have been the focus of classic network optimization theory for decades. However, in modern network systems, it is common that the network controller can only observe and operate on some nodes (i.e., overlay nodes), and the other nodes (i.e., underlay nodes) are neither observable nor controllable. Moreover, the dynamics can be non-stochastic or even adversarial. In this paper, we focus on the network utility maximization (NUM) problem for networks with overlay-underlay structures. The network dynamics, such as packet admissions, external arrivals and control actions of underlay nodes, can be stochastic, non-stochastic or even adversarial. We propose the Tracking Drift-plus-Penalty (TDP*) algorithm that only operates on the overlay nodes and does not require direct observations of the underlay nodes, and analyze the tradeoffs between the average utility and queue backlog. We show that as long as the peak queue backlog of the network is sublinear in time horizon, TDP* can solve the NUM problem, i.e., reaching the maximum utility while preserving stability.**

***Index Terms*— Network control, resource allocation, routing, queueing theory.**

I. Introduction

NETWORK optimization has been an active research area for decades. However, most classic control algorithms like MaxWeight [1] and Drift-plus-Penalty [2] can only be applied to networks in which the controllers have instantaneous observations of the global network state (e.g., queue backlogs), and all nodes cooperatively execute the control commands. Moreover, the network dynamics like the external arrivals are usually restricted to be stochastic and time-invariant.

However, with the rapid development of information technology, modern network systems are too complex to be characterized by the aforementioned framework. For example, due to security or economic concerns, many network systems do not offer full observation and control access to the controllers [3]. Such networks can be modeled by an overlay-underlay framework, where the controller can only observe and control the overlay nodes, with the underlay nodes being "black boxes" that cannot be directly observed or controlled. The underlay nodes may apply non-stochastic policies, which leads to challenges in the design and analysis of overlay optimal policies. Another example is computer security, where some nodes may be hijacked by an adversary and become unobservable and uncontrollable. Even worse, to maximize the damage, the adversary may change its actions dynamically according to the controller's actions [4], [5], [6].

In this paper, we focus on the network utility maximization (NUM) problem. Specifically, we aim to maximize the average network utility while preserving queue stability for networks with unobservable and uncontrollable nodes. Moreover, the dynamics, such as packet admissions, external arrivals and control actions of underlay nodes, can be stochastic, non-stochastic or even adversarial. We propose an algorithm named **T**racking **D**rift-plus-**P**enalty (TDP*),[1] which, to the best of our knowledge, is the first control algorithm to solve NUM problems under such challenging network settings.

The major technical challenges addressed in this work are three-fold: 1) underlay nodes are unobservable and uncontrollable, 2) the external arrivals and underlay policies can be stochastic, non-stochastic or even adversarial, and 3) the controller aims to maximize general network utilities instead of merely stabilizing the network. In the following we briefly discuss prior works pertaining to the above challenges.

Control algorithms for overlay-underlay networks include the Threshold-based Backpressure (BP-T) algorithm [7], the Overlay Backpressure (OBP) algorithm [8], the Optimal Overlay Routing Policy (OORP) algorithm [9] and the Tracking-MaxWeight (TMW) and Truncated Upper Confidence Reinforcement Learning (TUCRL) algorithms [10]. These algorithms apply the Lyapunov optimization framework and only need to control the overlay nodes. However, they all require instantaneous observations and can only optimize the network throughput instead of general network utilities.

Manuscript received 28 November 2022; revised 28 June 2023; accepted 13 August 2023; approved by IEEE/ACM Transactions on Networking Editor L. Huang. This work was supported in part by the National Science Foundation (NSF) under Grant CNS-1524317, Grant CNS-1907905, and Grant CNS-1735463; and in part by the Office of Naval Research (ONR) under Grant N00014-20-12119. Part of the material in this paper was presented at Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2019. *(Corresponding author: Bai Liu.)*

Bai Liu, Quang Minh Nguyen, and Eytan Modiano are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: bailiu@mit.edu).

Qingkai Liang is with Celer Network, Mountain View, CA 94043 USA.

Digital Object Identifier 10.1109/TNET.2023.3307684

[1] We use TDP* to distinguish from our earlier version of TDP that required instantaneous observation of uncontrollable nodes.

An alternative model of the overlay-underlay framework is a Partially Observable Markov Decision Process (POMDP). POMDPs seek to optimize general objectives with some system states being unobservable and uncontrollable. However, most works on POMDP focus on heuristic algorithms and lack theoretical performance guarantees. Theoretical studies on POMDPs [11], [12], [13], [14], [15], [16] apply value iteration or policy search, yet are only practical for networks of small size.

There has been a significant number of studies on networks with non-stochastic and adversarial dynamics. The work of [17] proposed the Adversarial Queueing Theory (AQT) framework, which introduced the "W constraint" that restricts the volume of external arrivals during a certain time window. Using the W constraint to characterize the network dynamics, the algorithms proposed in [18] and [19] can stabilize single-hop wireless communication systems. The work of [20] proposed the Tracking Algorithm (TA) that can stabilize general multi-hop networks, under a more relaxed constraint named "V_T constraint". The V_T constraint only requires the peak queue backlog under the optimal policy to be constrained to V_T. However, the aforementioned works all require instantaneous observations of the underlay dynamics. The work of [21] proposed the MaxWeight for Networks with Unobservable Malicious Nodes (MWUM) algorithm, which is a throughput-optimal algorithm that can be applied to partially observable and controllable networks. However, the MWUM algorithm cannot optimize general network utilities. Note that the W and V_T constraints are conditions for the whole network, i.e., all external arrivals and underlay actions generated by the network should satisfy the constraints.

Classic algorithms for NUM problems [2], [22], [23], [24] require stochastic dynamics. The work of [25] extended the classic Drift-plus-Penalty algorithm and Tracking Algorithm (TA) to networks with adversarial dynamics. However, both algorithms require full observability and controllability. The conference version of this paper, [26], proposed a preliminary version of the Tracking Drift-plus-Penalty (TDP) algorithm for partially controllable networks with adversarial dynamics. This paper further extended TDP to partially observable settings, and applied the V_T constraint to further relax the constraints on the underlay dynamics.

Our main contributions are summarized below.

We first propose TDP*, which uses estimates of the state of the underlay nodes instead of direct observations and only requires to control the overlay nodes. We show that as long as the NUM problem has at least one solution satisfying the V_T constraint, TDP* can achieve maximum utility while preserving stability. Note that this condition is non-trivial for partially controllable networks, since the uncontrollable nodes may admit an excessive number of packets such that no overlay policy can stabilize the system.

We also rigorously derive the upper bounds for the gap to maximum utility and the queue backlogs, under stochastic, non-stochastic and adversarial dynamics. The bounds explicitly reveal the trade-off between the utility gap and the queue backlog, through a parameter V.

Furthermore, tuning the value of V requires the value of time horizon T in advance. In practice, such information may not be available. Thus, we extended TDP* so that the value of V is updated in an online manner during the operation.

The rest of this paper is organized as follows. We introduce the network model and discuss different types of network dynamics in detail in Section II. We introduce TDP* in Section III. In Section IV, we derive the bounds to the utility gap and queue backlog under TDP*, and show that TDP* solves the NUM problem as long as the network is stabilizable. Section VI presents simulation results and Section VII concludes the paper.

II. Model

We consider a multi-hop network with N nodes and denote the set of nodes by $\mathcal{N}$. The nodes are classified into two types: the set of overlay nodes $\mathcal{O}$ and the set of underlay nodes $\mathcal{U}$. The network has K classes of data and the data of class k is destined for sink d_k. The set of data classes is denoted by $\mathcal{K}$. The link capacity between node i and j is C_{ij}. We assume that time is slotted and the time horizon is T.

At the beginning of time slot t, a node $i \in \mathcal{N}$ has $Q_{ik}(t)$ buffered packets of class k and receives $a_{ik}(t)$ external packets of class k. For simplicity, we assume that $Q_{ik}(0) = 0$ for each i and k. The controller then admits $\gamma_{ik}(t) \in \big[0, a_{ik}(t)\big]$ packets of flow k to node i. We denote the set of $\gamma_{ik}(t)$ as $\boldsymbol{\gamma}(t)$, which can be decomposed into overlay admissions $\boldsymbol{\gamma}_o(t)$ and underlay admissions $\boldsymbol{\gamma}_u(t)$. We assume that we have instantaneous observations of $\boldsymbol{\gamma}_u(t)$. Denote by $U\big(\boldsymbol{\gamma}(t)\big) = \sum_{i,k} U_{ik}\big(\gamma_{ik}(t)\big)$ the network utility function, where each $U_{ik}\big(\gamma_{ik}(t)\big)$ is the utility gained by admitting $\gamma_{ik}(t) \in \big[0, a_{ik}(t)\big]$ packets of flow k to node i. Common network utilities include:

- Total throughput: $U\big(\boldsymbol{\gamma}(t)\big) = \sum_{i,k} \gamma_{ik}(t)$.
- Proportional fairness: $U\big(\boldsymbol{\gamma}(t)\big) = \sum_{i,k} \log\big(1 + \gamma_{ik}(t)\big)$.
- Power allocation: $U\big(\boldsymbol{\gamma}(t)\big) = -\sum_{i,k} P_{ik}\big(\gamma_{ik}(t)\big)$, where $P_{ik}(\cdot)$ is the power cost function.

For an overlay node $i \in \mathcal{O}$, we denote by $f_{ijk}(t)$ the number of packets of class k transmitted to a neighbor j as decided by the network controller under a given policy. The set of all $f_{ijk}(t)$ at time t is denoted by $\boldsymbol{f}(t)$. However, there may not be enough buffered packets (i.e., $Q_{ik}(t) + a_{ik}(t)$) to support the planned transmissions, and the actual number of packets transmitted, denoted by $\tilde{f}_{ijk}(t)$, might be less than $f_{ijk}(t)$. In this case, the controller can decide the actual transmissions arbitrarily, as long as the following constraint is satisfied.

$$\begin{cases} \sum\limits_{j \in \mathcal{N}} \tilde{f}_{ijk}(t) = Q_{ik}(t) + a_{ik}(t) \\ 0 \leqslant \tilde{f}_{ijk}(t) \leqslant f_{ijk}(t). \end{cases}$$

For an underlay node $i \in \mathcal{U}$, we denote by $\mu_{ijk}(t)$ the number of packets of class k transmitted to a neighbor j from an underlay node $i \in \mathcal{U}$ under a given underlay policy and the actual number of packets transmitted by $\tilde{\mu}_{ijk}(t)$. The set of all $\mu_{ijk}(t)$ at time t is denoted by $\boldsymbol{\mu}(t)$. The network controller cannot directly observe $Q_{ik}(t)$ or implement control policies at the underlay. We assume that by applying network inference methods (e.g., probing [27], [28]), the network controller can obtain estimates $\hat{Q}_{ik}(t)$ of underlay queue backlog Q_{ik}, and that such estimates are only available sporadically. We denote by Γ_i the set of time slots when estimates are made for node i. In other words, for an underlay node $i \in \mathcal{U}$, the network controller only has an estimate $\hat{Q}_{ik}(t)$ of queue backlog $Q_{ik}(t)$ for $t \in \Gamma_i$. We denote $\tau_{ik}(t)$ as the time slot when the most recent state estimate of class k at node i is obtained, i.e.,

$$\tau_{ik}(t) = \max_{\tau \in \Gamma_{ik}:\tau \leqslant t} \tau,$$

with which we define $L(t) \triangleq \max_i \big(t - \tau_{ik}(t)\big)$, which denotes the largest delay in underlay observations at time t and assume that the average observation delay is sublinear in the time horizon T, i.e.,

$$\sum_{t=0}^{T-1} \frac{L(t)}{T} = o(T). \tag{1}$$

This assumption is needed to control the impact of outdated underlay observations, and is not hard to satisfy. If the observations of underlay nodes occur with fixed interval, then it is easy to show that $\sum_{t=0}^{T-1} L(t)/T = \mathcal{O}(1)$. More generally, the condition is met as long as the k^{th} observation interval of underlay nodes grows slower than the order of k^{α} where $\alpha \geqslant 0$.

The estimate can be erroneous. For an underlay node $i \in \mathcal{U}$ and $t \in \Gamma_{ik}$, we define the error as $\epsilon_{ik}(t) \triangleq \hat{Q}_{ik}(t) - Q_{ik}(t)$. Our algorithm is robust to estimation errors. To guarantee the desired performance, we only need to assume that the errors grow sublinearly in time, i.e.,

$$|\epsilon_{ik}(t)| = o(t). \tag{2}$$

We further assume the system dynamics to be bounded, i.e.,

$$0 \leqslant a_{ik}(t), U_{ik}\big(\gamma_{ik}(t)\big), f_{ijk}(t), \mu_{ijk}(t) \leqslant D,\ \forall i,j,k,t \tag{3}$$

for some constant $D \geqslant 0$.

Mathematically, the queue backlogs evolve according to the following rule (we use the operator $[x]^+ \triangleq \max\{x, 0\}$)

$$Q_{ik}(t+1) = \begin{cases} \big[Q_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} f_{ijk}(t)\big]^+ \\ \quad + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t),\ i \in \mathcal{O} \\ \big[Q_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} \mu_{ijk}(t)\big]^+ \\ \quad + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t),\ i \in \mathcal{U}. \end{cases}$$

We use the network event sequence of external arrivals, underlay admissions and underlay transmissions, i.e., $\big\{\boldsymbol{a}(t), \boldsymbol{\gamma}_u(t), \boldsymbol{\mu}(t)\big\}_{0\leqslant t\leqslant T-1}$, to characterize the underlay behaviors. The policy taken by the network controller can be characterized by a function π that maps a network event sequence to an overlay action sequence, i.e.,

$$\pi : \big\{\boldsymbol{a}(t), \boldsymbol{\gamma}_u(t), \boldsymbol{\mu}(t)\big\}_{0\leqslant t\leqslant T-1} \to \big\{\boldsymbol{\gamma}_o(t), \boldsymbol{f}(t)\big\}_{0\leqslant t\leqslant T-1}.$$

Note that this definition is equivalent to policies that make decisions based on queue backlogs, since the action sequence determines the queue backlogs. It is also worthwhile to emphasize that the underlay actions $\boldsymbol{\gamma}_u(t)$ are unobservable to the controller.

The packet admissions, external arrivals and control actions of underlay nodes are generated differently under different network dynamics. The dynamics can be classified into three categories: stochastic dynamics, non-stochastic dynamics and adversarial dynamics, as introduced below.

A. Stochastic Dynamics

Under stochastic dynamics, the external arrivals $a_{ik}(t)$'s are i.i.d across time. We assume that control policies of the underlay nodes are queue agnostic (i.e. the actions are independent of the queue backlogs), such as randomized routing and shortest path protocols.

Our goal is to design an algorithm that maximizes the average network utility while keeping the network rate stable, i.e.,

$$\begin{aligned} \max_{\pi} \quad & \lim_{T\to\infty} \frac{\mathbb{E}\Big[\sum_{t=0}^{T-1} U\big(\boldsymbol{\gamma}^{\pi}(t)\big)\Big]}{T} \\ s.t. \quad & \lim_{T\to\infty} \frac{\mathbb{E}\Big[\sum_{i\in\mathcal{N},k} Q_{ik}^{\pi}(T)\Big]}{T} = 0 \end{aligned} \tag{4}$$

where we use superscript π to distinguish the variables under policy π. (e.g., $Q_{ik}^{\pi}(t)$ is the queue backlog of class k data at node i at t under policy π). We use mean rate stability to characterize the stability constraint, which implies that as $t \to \infty$, the expected queue backlog grows up to a sublinear function of t and the arrival rate is no greater than the service rate.

We assume that there exists a policy π^* that solves (4). If there are multiple policies satisfying the conditions, we arbitrarily select any one of them to be π^*. We define V_T as the maximum queue backlog under π^* during the whole process, i.e.,

$$V_T \triangleq \mathbb{E}\Bigg[\max_{0\leqslant t\leqslant T} \sum_{i\in\mathcal{N},k} Q_{ik}^{\pi*}(t)\Bigg].$$

We assume that V_T is sublinear in time horizon T, i.e., $V_T = o(T)$.

For any policy π, we use utility regret to characterize the accumulated gap between the utilities under π and π^*, defined as follows.

Definition 1: The utility regret achieved by applying policy π is defined to be

$$R_T^{\pi} = \mathbb{E}\Bigg[\sum_{t=0}^{T-1} U\big(\boldsymbol{\gamma}^{\pi*}(t)\big) - \sum_{t=0}^{T-1} U\big(\boldsymbol{\gamma}^{\pi}(t)\big)\Bigg].$$

It is straightforward to see that maximizing utility is equivalent to minimizing the utility regret. If an algorithm obtains the performance of $R_T^{\pi} = o(T)$, then its average utility converges to the maximum utility asymptotically.

B. Non-Stochastic Dynamics

Under non-stochastic dynamics, the external arrivals and underlay actions are generated arbitrarily and cannot be captured by a stochastic process. Instead, we aim to solve

$$\begin{aligned} \max_{\pi} \quad & \lim_{T\to\infty} \frac{\sum_{t=0}^{T-1} U\big(\boldsymbol{\gamma}^{\pi}(t)\big)}{T} \\ s.t. \quad & \lim_{T\to\infty} \frac{\sum_{i\in\mathcal{N},k} Q_{ik}^{\pi}(T)}{T} = 0 \end{aligned} \tag{5}$$

for any occurred network event sequence.

We assume that, there exists a policy π^* such that for each possible network event sequence, the overlay action sequence generated by π^* solves (5). If there are multiple policies satisfying the conditions, we arbitrarily select any one of them to be π^*. We define V_T to be the worst-case peak total queue backlog under π^* across network event sequences, i.e.,

$$V_T \triangleq \max_{\{\boldsymbol{a}(t),\boldsymbol{\gamma}_u(t),\boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}} \max_{0\leqslant t\leqslant T} \sum_{i\in\mathcal{N},k} Q_{ik}^{\pi^*}(t).$$

We assume that V_T is sublinear in time horizon T, i.e., $V_T = o(T)$. The definition of V_T can also be regarded as a constraint on network dynamics: the network should be benign such that the peak total queue backlog under π^* is always bounded by V_T.

We define the utility regret in a worst-case manner, i.e.,

Definition 2: The utility regret achieved by applying policy π is defined to be

$$R_T^{\pi} = \max_{\{\boldsymbol{a}(t),\boldsymbol{\gamma}_u(t),\boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}} \sum_{t=0}^{T-1} U\big(\boldsymbol{\gamma}^{\pi^*}(t)\big) - \sum_{t=0}^{T-1} U\big(\boldsymbol{\gamma}^{\pi}(t)\big).$$

Since R_T^{π} is the worst-case regret, if an algorithm obtains $R_T^{\pi} = o(T)$, then for each possible network event sequence, it converges to the maximum utility corresponding to the network event sequence.

C. Adversarial Dynamics

Under non-stochastic dynamics, network event sequences can be arbitrary but are independent of the controller's actions. However, under adversarial dynamics, the underlay nodes are controlled by an intelligent adversary, who can change packet admissions, external arrivals and control actions of underlay nodes according to the control actions up to time $t-1$ to maximize the impact on the achieved utility. For instance, in Denial-of-Service (DDoS) attack, the attacker hijacks and takes control of multiple machines in the network by planting Trojans or scanning for security holes [4]. The adversary may consider the past queue backlogs and transmission history and send a large number of requests to the most vulnerable nodes.

Similar to Section II-B, we assume that for each possible network event sequence, (5) always has a solution, and continue using the definitions of V_T and R_T^{π} of Section II-B.

However, under adversarial dynamics, the coupling between the controller and adversary brings significant challenges in solving the NUM problems formulated as (5). As analyzed in Section IV, we can calculate an optimal action sequence that maximize average utility while preserving rate stability for any **given** network event sequence. However, under adversarial dynamics, when the optimal action sequence is applied to the system, the adversary may adjust future network events so that the optimal action sequence no longer maximizes the average utility. Coping with the coupling issue remains an open problem and is beyond the scope of this paper. Nonetheless, we can show that, no matter how malicious the adversary is, for **any realized** network event sequence, the TDP* algorithm is guaranteed to maximize the utility.

For readers' convenience, we summarize the variable notations in Table I.

TABLE I
VARIABLE NOTATIONS

N	The number of queues in the queueing network
$\mathcal{N}$, $\mathcal{O}$, $\mathcal{U}$	The set of all nodes, overlay nodes, underlay nodes
$\mathcal{K}$	The set of data types
C_{ij}	The link capacity between node i and j
d_k	The destination of the data of class k
T	The time horizon
π	A policy, can be characterized as a function that maps a network event sequence to an overlay action sequence
π^*	The policy that solves the NUM problems
Γ_i	The set of time slots when an estimation of Q_{ik} was made for node $i\in\mathcal{U}$
$\tau_i(t)$	The most recent time an estimate of node i was made for node $i\in\mathcal{U}$ at t
$L(t)$	The maximum delay of estimates at t, i.e., $\max_{i\in\mathcal{U},k} t-\tau_i(t)$
$Q_{ik}^{\pi}(t)$	Under policy π, the queue backlog of class k at node $i\in\mathcal{N}$ at t
$\hat{Q}_{ik}^{\pi}(t)$	Under policy π, the estimated queue backlog of class k at node $i\in\mathcal{N}$ at $t\in\Gamma_i$
$\epsilon_{ik}(t)$	the estimation error made at t for class k at node $i\in\mathcal{U}$ i.e., $\hat{Q}_{ik}(t)-Q_{ik}(t)$
$a_{ik}^{\pi}(t)$	Under policy π, the number of external packets of class k arriving at node $i\in\mathcal{N}$ at t
$\gamma_{ik}^{\pi}(t)$	Under policy π, the number of admitted packets of class k at node $i\in\mathcal{N}$ at t
$U_{ik}\big(\gamma_{ik}^{\pi}(t)\big)$	Under policy π, the utility of class k at node $i\in\mathcal{N}$ induced by $\gamma_{ik}^{\pi}(t)$
$f_{ijk}^{\pi}(t)$, $\tilde{f}_{ijk}^{\pi}(t)$	Under policy π, the planned and actual number of packets of class k transmitted from node $i\in\mathcal{O}$ to $j\in\mathcal{N}$ at t
$\mu_{ijk}^{\pi}(t)$, $\tilde{\mu}_{ijk}^{\pi}(t)$	Under policy π, the planned and actual number of packets of class k transmitted from node $i\in\mathcal{U}$ to $j\in\mathcal{N}$ at t
$g_{ijk}^{\pi}(t)$, $\tilde{g}_{ijk}^{\pi}(t)$	In the imaginary network, under policy π, the planned and actual number of packets of class k transmitted from node $i\in\mathcal{U}$ to $j\in\mathcal{N}$ at t
$X_{ik}^{\pi}(t)$	Under policy π, the virtual queue backlog of class k at node $i\in\mathcal{U}$ at t
$Y_{ik}^{\pi}(t)$	$Q_{ik}^{\pi}(t)-X_{ik}^{\pi}(t)$ for $i\in\mathcal{U}$
V_T	The peak queue backlog during $0\leqslant t\leqslant T$ under π^*

III. Our Approach

The key challenges in solving the NUM problem are two-fold. First, the partial observability and controllability make classical algorithms such as MaxWeight [1] unusable. Second, the external arrivals and the routing actions taken by the underlay nodes may not be cooperative and may even impact the utility and destabilize the network. While some existing works attempts to solve NUM problems in partially controllable or adversarial settings, no algorithm is capable of handling unobservability, uncontrollability, and non-stochastic/adversarial dynamics together.

A. Overview

The core of our algorithm is to "track" the underlay dynamics (i.e., the queue backlog and service of the underlay nodes) and solve the NUM problem based on the tracked information.

Specifically, the controller constructs an "imaginary" system with the same topology as the real system, yet all nodes are fully observable and controllable. The actions of each underlay node $i \in \mathcal{U}$ in the imaginary system can be decided by the controller, and we use $g_{ijk}(t)$ to denote the number of packets of class k transmitted to neighbor j. However, $g_{ijk}(t)$ may differ from the actual underlay action $\mu_{ijk}(t)$, thus causing gaps in queue backlogs between the imaginary system (denoted by X_{ik}) and the real system (still denoted by Q_{ik}). Meanwhile, the controller directly observes the overlay queue backlogs in the real system, and enforce them as the overlay queue backlog in the "imaginary" system, so that each overlay node $i \in \mathcal{O}$ is always synchronized with the real system, i.e., the queue backlogs Q_{ik} and the action γ_{ik} and f_{ijk} at overlay nodes are the same across the imaginary system and the real system.

We define the gap between Q_{ik} and X_{ik} by $Y_{ik} \triangleq Q_{ik} - X_{ik}$. The queue backlog in the real system can be decomposed into the queue backlog in the imaginary system and the queue backlog gap between the two systems, i.e.,

$$\underbrace{\sum_{i\in\mathcal{N},k} Q_{ik}(t)}_{\text{backlog in the real system}} = \underbrace{\sum_{i\in\mathcal{O},k} Q_{ik}(t) + \sum_{i\in\mathcal{U},k} X_{ik}(t)}_{\text{backlog in the imaginary system}} + \underbrace{\sum_{i\in\mathcal{U},k} Y_{ik}(t)}_{\text{gap between systems}}. \tag{6}$$

Since the imaginary system is easier to control, our approach is to solve the NUM problem for the imaginary system, while controlling the gap between the two systems.

B. Algorithm

The Tracking Drift-plus-Penalty (TDP*) algorithm enhances the classical Drift-plus-Penalty algorithm [2] and can be applied to stochastic dynamics, non-stochastic dynamics and adversarial dynamics. To minimize the queue backlog and to maximize the utility simultaneously in the imaginary system, we aim at minimizing the following Lyapunov function

$$\Phi(t) \triangleq \frac{1}{2}\sum_{i\in\mathcal{O},k} Q_{ik}^2(t) + \frac{1}{2}\sum_{i\in\mathcal{U},k} X_{ik}^2(t) + \frac{1}{2}\sum_{i\in\mathcal{U},k} Y_{ik}^{+2}(t) - V\cdot\sum_{\tau=0}^{t-1} U\big(\gamma(\tau)\big). \tag{7}$$

where $Y_{ik}^+(t) = \max\{Y_{ik}(t), 0\}$ and V is a parameter that will be used to tune the utility-backlog trade-off.

To control the growth of (7), we minimize the Lyapunov drift $\Delta\Phi(t) \triangleq \Phi(t+1) - \Phi(t)$ during each time slot. It can be shown that minimizing $\Delta\Phi(t)$ is equivalent to minimizing

$$\sum_{i\in\mathcal{O},k} Q_{ik}(t)\delta Q_{ik}(t) + \sum_{i\in\mathcal{U},k} X_{ik}(t)\delta X_{ik}(t) + \sum_{i\in\mathcal{U},k} Y_{ik}^+(t)\Delta Y_{ik}^+(t) - V\cdot U\big(\gamma^\pi(t)\big), \tag{8}$$

where $\delta Q_{ik}(t)$, $\delta X_{ik}(t)$ and $\Delta Y_{ik}(t)$ are defined as

$$\begin{cases} \delta Q_{ik}(t) \triangleq & \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} f_{ijk}(t) + \sum_{j\in\mathcal{O}} f_{jik}(t) \\ & + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t), \quad i\in\mathcal{O} \\ \delta X_{ik}(t) \triangleq & \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} g_{ijk}(t) + \sum_{j\in\mathcal{O}} f_{jik}(t) \\ & + \sum_{j\in\mathcal{U}} g_{jik}(t), \quad i\in\mathcal{U} \\ \Delta Y_{ik}(t) \triangleq & Y_{ik}(t+1) - Y_{ik}(t), \quad i\in\mathcal{U}. \end{cases}$$

The proof can be found in Appendix B and C. Note that we use δ instead of Δ for $\delta Q_{ik}(t)$ and $\delta X_{ik}(t)$ because they are not the actual one-slot changes (using $\tilde{f}_{ijk}$ and $\tilde{g}_{ijk}$) but the planned one-slot changes (using f_{ijk} and g_{ijk}).

However, for an underlay node $i \in \mathcal{U}$, the network controller does not have instantaneous access to its queue backlog $Q_{ik}(t)$ and thus the value of $Y_{ik}(t)$ is unavailable to the network controller. As discussed in Section II, the network controller can obtain estimates of Q_{ik} at certain time slots Γ_i. Therefore, the network controller can use the most recently estimated (possibly erroneous) $\hat{Q}_{ik}(t)$ to estimate $Y_{ik}(t)$, i.e.,

$$\hat{Y}_{ik}(t) = \hat{Q}_{ik}(\tau_i(t)) - X_{ik}(t), \tag{9}$$

where $\tau_i(t)$ is the most recent time when an estimation of Q_{ik} was made, i.e., $\tau_i(t) \triangleq \max_{\tau\in\Gamma_i:\tau\leqslant t}\tau$. By replacing $Y_{ik}^+(t)$ with $\hat{Y}_{ik}^+(t)$ and discarding uncontrollable variables including a_{ik} and μ_{ijk}, Eqn (8) can be fomulated as (10), shown at the bottom of the next page, where the solution is denoted by $\boldsymbol{\gamma}_o^{\pi_T}(t)$, $\boldsymbol{f}^{\pi_T}(t)$ and $\boldsymbol{g}^{\pi_T}(t)$.

For each time slot, the network controller solves (10) and applies $\boldsymbol{\gamma}_o^{\pi_T}(t)$ and $\boldsymbol{f}^{\pi_T}(t)$ to the overlay nodes in the **real network**, meanwhile using $\boldsymbol{f}^{\pi_T}(t)$ and $\boldsymbol{g}^{\pi_T}(t)$ to update $X_{ik}(t)$ for all underlay nodes $i \in \mathcal{U}$, according to

$$X_{ik}(t+1) = \Big[X_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} g_{ijk}(t)\Big]^+ + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) + \sum_{j\in\mathcal{U}} g_{jik}(t), \tag{11}$$

where, for technical reasons, we assume that in the **imaginary network**, underlay nodes can transmit dummy packets when the allotted packets to be transmitted are less than the queue backlog (i.e., $\tilde{g}_{ijk} \equiv g_{ijk}$ for $i \in \mathcal{U}$). This assumption does not affect the performance of the algorithm, as analyzed in the next section.

The complete algorithm is given in Algorithm 1.

Algorithm 1 The TDP* Algorithm

```
Input: $T$, $Q_{ik}(0)$, $\Gamma_i$ for $i \in \mathcal{U}$
Initialization: $X_{ik}(0) \leftarrow Q_{ik}(0)$ for $i \in \mathcal{N}$, $\hat{Y}_{ik}(0) \leftarrow 0$ for $i \in \mathcal{U}$
for $t \leftarrow 0, 1, \cdots, T-1$ do
    Obtain $Q_{ik}(t)$ and $a_{ik}(t)$ for $i \in \mathcal{O}$, $\gamma_{ik}(t)$ and $X_{ik}(t)$ for $i \in \mathcal{U}$
    for $i \in \mathcal{U}$ do
        if $t \in \Gamma_i$ then
            Obtain an estimation $\hat{Q}_{ik}(t)$ for $k$
        end if
        Update $\hat{Y}_{ik}(t)$ using Eqn (9) for $k$
    end for
    Solve Eqn (10) and obtain $\gamma_o^{\pi_T}(t)$, $f^{\pi_T}(t)$ and $g^{\pi_T}(t)$
    Implement $\gamma_o^{\pi_T}(t)$ and $f^{\pi_T}(t)$ to overlay nodes $\mathcal{O}$ in the real network
    Update $X_{ik}(t+1)$ using Eqn (11) for $i \in \mathcal{U}$ and $k$
end for
Output: action sequence for overlay nodes $f^{\pi_T}(t)$ for $t = 0, \cdots, T-1$, i.e., $\pi_T$
```

IV. Performance Analysis

We derive the following universal bounds for stochastic, non-stochastic and adversarial dynamics, using V_T in Theorem 1. For conciseness, we use $Q_T^{\pi_T}$ to denote the expected total queue backlog at T for stochastic dynamics, and to denote the worst-case total queue backlog at T for non-stochastic and adversarial dynamics, under policy TDP*.

Theorem 1: Under the TDP* algorithm, we have the following performance bounds:

$$\begin{cases} R_T^{\pi_T} = \mathcal{O}\Big(\dfrac{V_T^{1/2}T^{3/2} + \sigma(L,\epsilon)}{V}\Big) \\ Q_T^{\pi_T} = \mathcal{O}\Big(V_T^{1/4}T^{3/4} + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\Big) \end{cases}$$

where $\sigma(L,\epsilon)$ characterizes the impact caused by sparse observations on underlay nodes and estimation errors, and is upper bounded by

$$\sigma(L,\epsilon) = \mathcal{O}\Bigg(\sum_{t=0}^{T-1} L(t) + \sum_{t=0}^{T-1}\sum_{i\in\mathcal{U},k} |\epsilon_{ik}(\tau_{ik}(t))|\Bigg).$$

Proof: For conciseness, we use superscript π to denote the variables that are obtained under policy π. For example, $\Phi^{\pi_T}(T)$ denotes the Lyapunov value at T under TDP*, and $\sum_{i,k} Q_{ik}^{\pi^*}(T)$ denotes the total queue backlog at T under policy π^*.

The outline of the proof is as follows. We first bound the queue backlog. By Lemma 1, bounding the queue backlog can be achieved by bounding the Lyapunov value $\Phi^{\pi_T}(T)$. To bound $\Phi^{\pi_T}(T)$, we use Lemma 2, 3 and 4 to bound the one-slot drift $\Delta\Phi^{\pi_T}(t)$. By summing up $\Delta\Phi^{\pi_T}(t)$ over time and using Lemmas 5 and 6, we bound $\Phi^{\pi_T}(T)$ and thus bound the queue backlog. We then bound the utility regret by rearranging the result in Lemma 4 and reusing Lemmas 5 and 6.

We assume the occurred network event sequence to be an arbitrary sequence $\{\boldsymbol{a}(t), \boldsymbol{\gamma}_u(t), \boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}$. We denote the corresponding overlay action sequence under TDP* by $\{\boldsymbol{\gamma}_o^{\pi_T}(t), \boldsymbol{f}^{\pi_T}(t)\}_{0\leqslant t\leqslant T-1}$, and the overlay action sequence under π^* by $\{\boldsymbol{\gamma}_o^{\pi^*}(t), \boldsymbol{f}^{\pi^*}(t)\}_{0\leqslant t\leqslant T-1}$.

To prove Theorem 1, we first prove the upper bound on the queue backlog at T. With the following lemma (see Appendix A for the proof), to bound the queue backlog, it suffices to bound $\Phi^{\pi_T}(T)$.

Lemma 1:

$$\sum_{i,k} Q_{ik}^{\pi_T}(T) \leqslant \sqrt{2KN\Phi^{\pi_T}(T) + 2K^2N^2DVT}.$$

To bound $\Phi^{\pi_T}(T)$, we start by bounding $\Delta\Phi^{\pi_T}(t)$. We upper bound $Q_{ik}^2(t+1) - Q_{ik}^2(t)$, $X_{ik}^2(t+1) - X_{ik}^2(t)$ and $Y_{ik}^{+2}(t+1) - Y_{ik}^{+2}(t)$ in Lemmas 2 and 3, respectively (see Appendices B and C for the proof).

Lemma 2: For each $t = 0, \cdots, T-1$, we have

$$\begin{cases} Q_{ik}^2(t+1) - Q_{ik}^2(t) \leqslant 2Q_{ik}(t)\delta Q_{ik}(t) + 6N^2D^2, i \in \mathcal{O} \\ X_{ik}^2(t+1) - X_{ik}^2(t) \leqslant 2X_{ik}(t)\delta X_{ik}(t) + 6N^2D^2, i \in \mathcal{U} \end{cases}$$

Lemma 3: For each $i \in \mathcal{U}$, k and $t = 0, \cdots, T-1$, we have

$$\begin{aligned} &Y_{ik}^{+2}(t+1) - Y_{ik}^{+2}(t) \\ &\leqslant 2\hat{Y}_{ik}^{+}(t)\Delta Y_{ik}(t) + \big(8L(t)+6\big)N^2D^2 + 4ND|\epsilon_{ik}(\tau_{ik}(t))|. \end{aligned}$$

With Lemma 2 and Lemma 3, we can upper bound $\Delta\Phi^{\pi_T}(t)$ as follows,

$$\begin{aligned} &\Delta\Phi^{\pi_T}(t) \\ &\leqslant \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\delta Q_{ik}^{\pi_T}(t) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\delta X_{ik}^{\pi_T}(t) \\ &\quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\Delta Y_{ik}^{\pi_T}(t) - V\cdot U\big(\boldsymbol{\gamma}^{\pi_T}(t)\big) \\ &\quad + (4L(t)+9)KN^3D^2 + 2KN^2D|\epsilon_{ik}(\tau_{ik}(t))|. \qquad (12) \end{aligned}$$

$$\begin{aligned} &\boldsymbol{\gamma}_o^{\pi_T}(t), \boldsymbol{f}^{\pi_T}(t), \boldsymbol{g}^{\pi_T}(t) \\ &= \arg\min_{\boldsymbol{\gamma}_o,\boldsymbol{f},\boldsymbol{g}} \sum_{i\in\mathcal{O},k} Q_{ik}(t)\cdot\Big[\gamma_{ik} + \sum_{j\in\mathcal{O}} f_{jik} + \sum_{j\in\mathcal{U}} g_{jik} - \sum_{j\in\mathcal{N}} f_{ijk}\Big] + \sum_{i\in\mathcal{U},k} X_{ik}(t)\cdot\Big[\sum_{j\in\mathcal{O}} f_{jik} + \sum_{j\in\mathcal{U}} g_{jik} - \sum_{j\in\mathcal{N}} g_{ijk}\Big] \\ &\qquad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{+}(t)\cdot\Big[\min\Big\{\sum_{j\in\mathcal{N}} g_{ijk}, X_{ik}(t)+\gamma_{ik}(t)\Big\} - \sum_{j\in\mathcal{U}} g_{jik}\Big] - V\cdot\sum_{i\in\mathcal{O},k} U_{ik}\big(\gamma_{ik}\big), \qquad (10) \\ &\quad s.t. \quad 0 \leqslant \gamma_{ik} \leqslant a_{ik}(t), f_{ijk} \geqslant 0, \ \textstyle\sum_k f_{ijk} \leqslant C_{ij}, \ g_{ijk} \geqslant 0, \ \textstyle\sum_k g_{ijk} \leqslant C_{ij}. \end{aligned}$$

For technical exposition, we consider the following quantities

$$\begin{cases} \Delta Q_{ik}^{\pi*}(t) \triangleq & \gamma_{ik}^{\pi*}(t) - \sum_{j\in\mathcal{N}} \tilde{f}_{ijk}^{\pi*}(t) + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}^{\pi*}(t) \\ & + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t), \quad i\in\mathcal{O} \\ \Delta X_{ik}^{\pi*}(t) \triangleq & \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t) + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}^{\pi*}(t) \\ & + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t), \quad i\in\mathcal{U} \\ \widetilde{\Delta} Y_{ik}^{\pi*}(t) \triangleq & \min\left\{X^{\pi}t_{ik}(t) + \gamma_{ik}(t), \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t)\right\} \\ & - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t), \quad i\in\mathcal{U}. \end{cases}$$

Particularly, $\Delta Q_{ik}^{\pi*}(t)$ and $\Delta X_{ik}^{\pi*}(t)$ are the actual one-slot changes of the queue backlogs of both the real and imaginary systems under the policy π^*. On the other hand, $\widetilde{\Delta} Y_{ik}^{\pi*}(t)$ captures the "hypothetical" one-slot change of the system gap if, given the current state to be $X_{ik}^{\pi_T}(t)$, the actions of π^* are applied. To ease analysis, we replace the one-slot changes $\delta Q_{ik}^{\pi_T}(t)$, $\delta X_{ik}^{\pi_T}(t)$ and $\Delta Y_{ik}^{\pi_T}(t)$ in (12) with $\Delta Q_{ik}^{\pi*}(t)$, $\Delta X_{ik}^{\pi*}(t)$ and $\widetilde{\Delta} Y_{ik}^{\pi*}(t)$ using the following lemma (see Appendix D for the proof). The intuition behind is that (10) minimizes the drift.

Lemma 4:

$$\begin{aligned} &\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\delta Q_{ik}^{\pi_T}(t) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\delta X_{ik}^{\pi_T}(t) \\ &\quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\Delta Y_{ik}^{\pi_T}(t) - V\cdot U\big(\boldsymbol{\gamma}^{\pi_T}(t)\big) + \\ &\leqslant \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi*}(t) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi*}(t) \\ &\quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta} Y_{ik}^{\pi*}(t) - V\cdot U\big(\boldsymbol{\gamma}^{\pi*}(t)\big). \end{aligned}$$

With Lemma 4, $\Delta\Phi^{\pi_T}(t)$ can be further upper bounded as

$$\begin{aligned} &\Delta\Phi^{\pi_T}(t) \\ &\leqslant \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi*}(t) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi*}(t) \\ &\quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta} Y_{ik}^{\pi*}(t) - V\cdot U\big(\boldsymbol{\gamma}^{\pi*}(t)\big) \\ &\quad + (4L(t)+9)KN^3D^2 + 2KN^2D|\epsilon_{ik}(\tau_{ik}(t))|. \end{aligned} \tag{13}$$

Summing up (13) from $t=0$ to $t=T-1$ gives us an upper bound to $\Phi^{\pi_T}(T)$. To assist the analysis, we prove Lemmas 5 and 6 (see Appendices E and F for the proof), as follows.

Lemma 5:

$$\begin{aligned} &\sum_{t=0}^{T-1}\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi*}(t) + \sum_{t=0}^{T-1}\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi*}(t) \\ &= \mathcal{O}\bigg(\Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/2}\cdot T^{3/2}\bigg). \end{aligned}$$

Lemma 6:

$$\sum_{t=0}^{T-1}\sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta} Y_{ik}^{\pi*}(t) \leqslant 0.$$

By summing up (13) from $t=0$ to $t=T-1$, inserting Lemma 5 and 6, we have

$$\Phi^{\pi_T}(T) = \mathcal{O}\bigg(\Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/2}\cdot T^{3/2} + \sigma(L,\epsilon)\bigg). \tag{14}$$

By inserting (14) into the result in Lemma 1, we show that

$$\begin{aligned} &\sum_{i,k} Q_{ik}^{\pi_T}(T) \\ &= \mathcal{O}\bigg(\Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/4}\cdot T^{3/4} + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\bigg). \end{aligned} \tag{15}$$

For stochastic dynamics, by taking expectation over the network event sequences on both sides of (15), we have

$$\begin{aligned} &\mathbb{E}\bigg[\sum_{i,k} Q_{ik}^{\pi_T}(T)\bigg] \\ &= \mathcal{O}\bigg(\mathbb{E}\bigg[\Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/4}\bigg]\cdot T^{3/4} + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\bigg) \\ &= \mathcal{O}\bigg(\mathbb{E}\bigg[\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\bigg]^{1/4}\cdot T^{3/4} + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\bigg) \\ &= \mathcal{O}\Big(V_T^{1/4}T^{3/4} + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\Big), \end{aligned} \tag{16}$$

where the second equation holds by apply Jensen's inequality and the last equation holds by the definition of V_T in Section II-A.

For non-stochastic and adversarial dynamics, we consider the worst case of queue backlog, as

$$\begin{aligned} &\max_{\{\boldsymbol{a}(t),\boldsymbol{\gamma}_u(t),\boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}} \sum_{i,k} Q_{ik}^{\pi_T}(T) \\ &= \mathcal{O}\bigg(\max_{\{\boldsymbol{a}(t),\boldsymbol{\gamma}_u(t),\boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}} \Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/4}\cdot T^{3/4} \\ &\qquad + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\bigg) \\ &= \mathcal{O}\Big(V_T^{1/4}T^{3/4} + \sqrt{TV} + \sqrt{\sigma(L,\epsilon)}\Big), \end{aligned} \tag{17}$$

where the last equation holds by the definition of V_T in Section II-B.

Equations (16) and (17) complete the bound on the queue backlog for all types of dynamics.

We now bound the utility regret. By rearranging the result in Lemma 4, we have

$$\begin{aligned} &V\cdot U\big(\boldsymbol{\gamma}^{\pi*}(t)\big) - V\cdot U\big(\boldsymbol{\gamma}^{\pi_T}(t)\big) \\ &\leqslant \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi*}(t) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi*}(t) \\ &\quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta} Y_{ik}^{\pi*}(t) - \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\delta Q_{ik}^{\pi_T}(t) \\ &\quad - \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\delta X_{ik}^{\pi_T}(t) - \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\Delta Y_{ik}^{\pi_T}(t), \end{aligned}$$

which, by inserting Lemma 2 and Lemma 3 to the last three terms, can be further upper bounded as

$$\begin{aligned} & V\cdot U\big(\boldsymbol{\gamma}^{\pi*}(t)\big) - V\cdot U\big(\boldsymbol{\gamma}^{\pi_T}(t)\big) \\ & \leqslant \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi*}(t) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi*}(t) \\ & \quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta} Y_{ik}^{\pi*}(t) \\ & \quad + \frac{1}{2}\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)^2 - \frac{1}{2}\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t+1)^2 \\ & \quad + \frac{1}{2}\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)^2 - \frac{1}{2}\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t+1)^2 \\ & \quad + \frac{1}{2}\sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T+}(t)^2 - \frac{1}{2}\sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T+}(t+1)^2 \\ & \quad + (4L(t)+9)KN^3D^2 + 2KN^2D|\epsilon_{ik}(\tau_{ik}(t))|. \end{aligned} \tag{18}$$

Summing up (18) from $t=0$ to time $t=T-1$, applying Lemma 5 and Lemma 6, we have

$$\begin{aligned} & V\cdot U\big(\boldsymbol{\gamma}^{\pi*}(t)\big) - V\cdot U\big(\boldsymbol{\gamma}^{\pi_T}(t)\big) \\ & = \mathcal{O}\bigg(\Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/2}\cdot T^{3/2} + \sigma(L,\epsilon)\bigg) \\ & \quad + \frac{1}{2}\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(0)^2 - \frac{1}{2}\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(T)^2 \\ & \quad + \frac{1}{2}\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(0)^2 - \frac{1}{2}\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(T)^2 \\ & \quad + \frac{1}{2}\sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T+}(0)^2 - \frac{1}{2}\sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T+}(T)^2 \\ & \leqslant \mathcal{O}\bigg(\Big(\max_{0\leqslant t\leqslant T}\sum_{i,k} Q_{ik}^{\pi*}(t)\Big)^{1/2}\cdot T^{3/2} + \sigma(L,\epsilon)\bigg), \end{aligned} \tag{19}$$

where the inequality holds because we assume the initial queue backlogs are zero.

For stochastic dynamics, similar to the analysis in bounding the queue backlog, by dividing by V and taking expectation over the network event sequences on both sides of (19), applying Jensen's inequality, and using Definition 1, we have $R_T^{\pi_T} = \mathcal{O}\Big(V_T^{1/2}T^{3/2}/V + \sigma(L,\epsilon)/V\Big)$. For non-stochastic and adversarial dynamics, also similar to the analysis in bounding the queue backlog, by dividing by V on both sides of (19), considering the worst case of network event sequences, and using Definition 2, we also obtain $R_T^{\pi_T} = \mathcal{O}\Big(V_T^{1/2}T^{3/2}/V + \sigma(L,\epsilon)/V\Big)$. Therefore, we obtain the bound on the utility regret for all types of dynamics. □

With Theorem 1, we can easily derive that the TDP* algorithm can solve the NUM problem, i.e., maximizing the utility while keeping stable queues, as in Theorem 2.

Theorem 2: For a network with stochastic / non-stochastic dynamics, the TDP* algorithm solves the NUM problem defined by (4) / (5), respectively.

Proof: By the assumptions made in Section II, we have $V_T = o(T)$. By (1) and (2), we have $\sigma(L,\epsilon) = o(T^2)$. We assume that there exists a number $0 \leqslant \alpha < 1$ such that $V_T = \mathcal{O}(T^\alpha)$ and $\sigma(L,\epsilon) = \mathcal{O}(T^{2\alpha})$. We choose $V = T^\beta$ with $(1+\alpha)/2 < \beta < 1$.

For stochastic dynamics, we have

$$\begin{cases} R_T^{\pi_T}/T = \mathcal{O}\Big(T^{\frac{1+\alpha}{2}-\beta} + T^{2\alpha-\beta-1}\Big) \\ \mathbb{E}\Big[\sum_{i\in\mathcal{N},k} Q_{ik}^{\pi}(T)\Big] = \mathcal{O}\Big(T^{\frac{3+\alpha}{4}} + T^{\frac{1+\beta}{2}} + T^{\alpha}\Big) = o(T), \end{cases} \tag{20}$$

which shows that as $T\to\infty$, the average utility converges to the maximum utility, while the queue backlog remains rate stable. Thus TDP* solves (4) for stochastic dynamics.

For non-stochastic dynamics, we have

$$\begin{cases} R_T^{\pi_T}/T = \mathcal{O}\Big(T^{\frac{1+\alpha}{2}-\beta} + T^{2\alpha-\beta-1}\Big) \\ \displaystyle\max_{\{\boldsymbol{a}(t),\boldsymbol{\gamma}_u(t),\boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}} \sum_{i,k} Q_{ik}^{\pi_T}(T) \\ \qquad = \mathcal{O}\Big(T^{\frac{3+\alpha}{4}} + T^{\frac{1+\beta}{2}} + T^{\alpha}\Big) = o(T), \end{cases} \tag{21}$$

which shows for any network event sequence, the average utility converges to the maximum utility and the queue backlog remains rate stable. Thus TDP* solves (5) for non-stochastic dynamics. □

Theorem 2 presents a strong result that if a solution to NUM problems (i.e., maximizing utility while preserving stability) with stochastic or non-stochastic dynamics exists, the TDP* algorithm is almost equivalent to the optimal policy. For adversarial dynamics, the results in (21) still holds. The queue backlog always remains rate stable. However, the average utility only converges to the maximum utility for a **given** network event sequence. Due to the coupling between the controller and adversary, when TDP* is actually applied to the system, the adversary may change the incoming network events to undermine the utility achieved by TDP*. Nonetheless, the results in (21) show that, no matter how malicious the adversary is, for **any realized** network event sequence, the TDP* algorithm is guaranteed to maximize the utility.

V. Online TDP*

If the time horizon T is unknown, it is hard set up an appropriate static V. The controller can dynamically estimate the time horizon using the doubling trick: in the beginning the time horizon estimate is $\hat{T} = T_0$. Every time the actual time elapsed exceed $\hat{T}$, the controller doubles the estimate, i.e., $\hat{T} \leftarrow 2\hat{T}$. Meanwhile, the controller uses the estimated $\hat{T}$ to decide the value of V dynamically. As defined in the proof of Theorem 2, there exists a number $0 \leqslant \alpha < 1$ such that $V_T = \mathcal{O}(T^\alpha)$. Given an estimate $\hat{T}$ to the time horizon, we choose $V = \hat{T}^\beta$ with $(1+\alpha)/2 < \beta < 1$. Under the doubling trick, the value of $\hat{T}$ at time t is

$$\hat{T} = \begin{cases} T_0, & 0 \leqslant t < T_0 \\ 2^k T_0, & 2^{k-1}T_0 \leqslant t < 2^k T_0,\ k\geqslant 1, \end{cases} \tag{22}$$

and the corresponding value of V at time t is

$$V(t) = \begin{cases} T_0^\beta, & 0 \leqslant t < T_0 \\ 2^{k\beta}T_0^\beta, & 2^{k-1}T_0 \leqslant t < 2^k T_0,\ k\geqslant 1. \end{cases} \tag{23}$$

We replace the static V in (10) with $V(t)$ defined in (23), and all other operations of TDP* remain unchanged. We can show that the online TDP* still solves the NUM problems, as stated below.

Theorem 3: When applying the doubling trick to estimate the time horizon T, the results in Theorem 2 still hold.

Proof: We first analyze the queue backlog. The analysis for the queue backlog in the proof of Theorem 1 still holds, with the only difference of replacing the static V in (15), (16) and (17) with the largest $V(t)$. From (22), for any real time horizon T, the estimated time horizon $\hat{T}$ is at most $2T$. Thus, the largest $V(t)$ is bounded as $\mathcal{O}\left(T^{\beta}\right)$. By replacing the V with T^{β} and inserting $V_T = \mathcal{O}(T^{\alpha})$ in (15), (16) and (17), the bounds for queue backlogs in (20) and (21) still hold. Therefore, TDP* stabilizes the system for all types of dynamics.

We then analyze the utility regret. We consider an arbitray network event sequence $\{\boldsymbol{a}(t), \boldsymbol{\gamma}_u(t), \boldsymbol{\mu}(t)\}_{0\leqslant t\leqslant T-1}$. By replacing the V with $V(t)$ in (18), summing it up from $t = 0$ to time $t = T-1$, and applying Lemma 5 and 6, we have

$$\sum_{t=0}^{T-1} V(t)\cdot\left(U\left(\boldsymbol{\gamma}^{\pi^*}(t)\right) - U\left(\boldsymbol{\gamma}^{\pi_T}(t)\right)\right) \leqslant C\cdot V_T^{1/2}T^{3/2} \tag{24}$$

for any $T > 0$, where C is a constant. For conciseness, we define $R^{\pi_T}_{t_0:t_1} \triangleq \sum_{t=t_0}^{t_1-1}\left(U\left(\boldsymbol{\gamma}^{\pi^*}(t)\right) - U\left(\boldsymbol{\gamma}^{\pi_T}(t)\right)\right)$. By inserting $V(t)$ defined in (23) into (24), we have for every $K\geqslant 1$,

$$\begin{aligned} T_0^{\beta}\cdot R^{\pi_T}_{T_0} &+ \sum_{j=1}^{K}\left(2^jT_0\right)^{\beta}\cdot R^{\pi_T}_{2^{j-1}T_0:2^jT_0} \\ &\leqslant\ C\cdot V^{1/2}_{2^KT_0}\cdot\left(2^KT_0\right)^{3/2}. \end{aligned} \tag{25}$$

Since $\beta < 1$, for any $1\leqslant k\leqslant K$, we have

$$\left(2^KT_0\right)^{\beta} = 2^{k\beta}\cdot\left(2^{K-k}T_0\right)^{\beta} \leqslant 2^k\cdot\left(2^{K-k}T_0\right)^{\beta},$$

with which we can bound $\left(2^KT_0\right)^{\beta}\cdot R^{\pi_T}_{2^KT_0}$ (with $K\geqslant 1$) as

$$\begin{aligned} &\left(2^KT_0\right)^{\beta}\cdot R^{\pi_T}_{2^KT_0} \\ &= \left(2^KT_0\right)^{\beta}\cdot R^{\pi_T}_{T_0} + \sum_{j=1}^{K}\left(2^KT_0\right)^{\beta}\cdot R^{\pi_T}_{2^{j-1}T_0:2^jT_0} \\ &\leqslant 2^K\cdot\left(T_0\right)^{\beta}\cdot R^{\pi_T}_{T_0} + \sum_{j=1}^{K}2^{K-j}\cdot\left(2^jT_0\right)^{\beta}\cdot R^{\pi_T}_{2^{j-1}T_0:2^jT_0} \\ &= \sum_{k=0}^{K-1}2^{K-k-1}\cdot\left(T_0^{\beta}\cdot R^{\pi_T}_{T_0} + \sum_{j=1}^{k}\left(2^jT_0\right)^{\beta}\cdot R^{\pi_T}_{2^{j-1}T_0:2^jT_0}\right) \\ &\quad + T_0^{\beta}\cdot R^{\pi_T}_{T_0} + \sum_{j=1}^{K}\left(2^jT_0\right)^{\beta}\cdot R^{\pi_T}_{2^{j-1}T_0:2^jT_0}. \end{aligned} \tag{26}$$

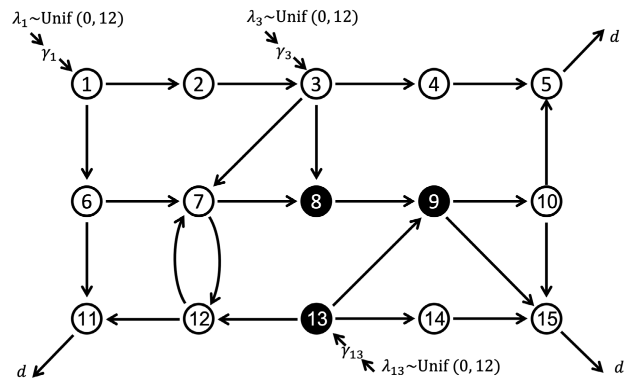


Fig. 1. The model of the 15-node network with stochastic dynamics.

By inserting (25) into (26), we have

$$\begin{aligned} \left(2^KT_0\right)^{\beta}\cdot R^{\pi_T}_{2^KT_0} &\leqslant \sum_{k=0}^{K-1}2^{K-k-1}\cdot C\cdot V^{1/2}_{2^kT_0}\cdot\left(2^kT_0\right)^{3/2} \\ &\quad + C\cdot V^{1/2}_{2^KT_0}\cdot\left(2^KT_0\right)^{3/2} \\ &\leqslant 4C\cdot V^{1/2}_{2^KT_0}\cdot(2^KT_0)^{3/2}, \end{aligned}$$

which leads to the result that for any $T = 2^KT_0$ with $K\geqslant 1$, we have

$$R^{\pi_T}_T/T \leqslant 4C\cdot V_T^{1/2}\cdot T^{1/2-\beta} \leqslant C'\cdot T^{\frac{1+\alpha}{2}-\beta},$$

where C' is a constant, and the second inequality holds by using the fact that $V_T = \mathcal{O}(T^{\alpha})$. We suppose the limit of $R^{\pi_T}_T/T$ exists as $T\to\infty$, then every subsequence converges to the same limit. Since $(1+\alpha)/2-\beta < 0$, the subsequence of $R^{\pi_T}_T/T$ with $T = 2^KT_0$ converges to zero. Therefore, $R^{\pi_T}_T/T$ also converges to zero. By applying similar analysis as the end of the proof of Theorem 1, we can show that the average utility regret converges to zero for all types of dynamics.

Combining the analysis on the queue backlog and the utility regret, the TDP* algorithm solves the NUM problem for all types of dynamics, thus completes the proof. □

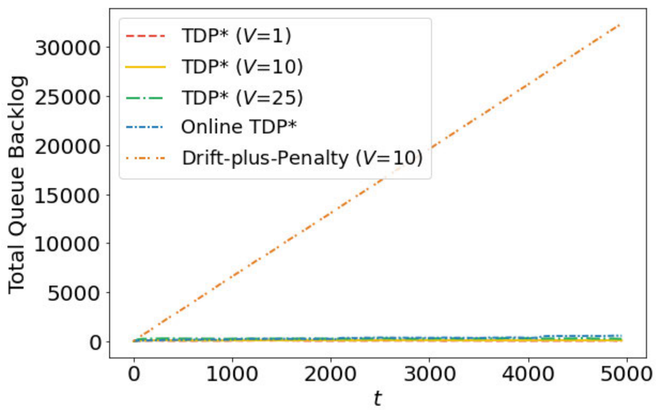
VI. Numerical Experiments

We conduct numerical experiments on two network systems to validate the performance analysis of TDP*. We study a complex system of 15 nodes with stochastic dynamics and a system of 12 nodes with adversarial dynamics to show the performance of our algorithm under different dynamics.

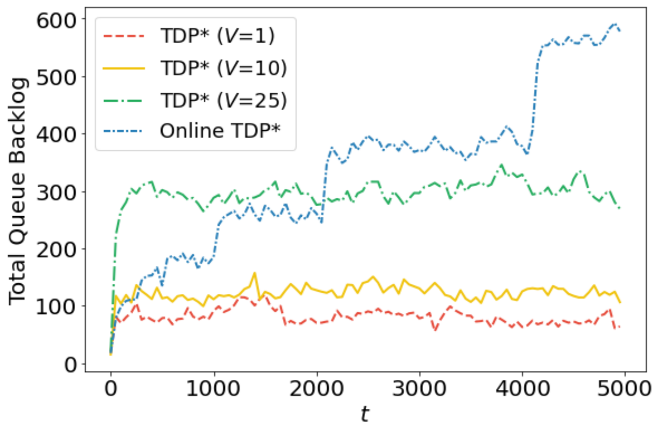
A. 15-Node Network With Stochastic Dynamics

We study a 15-node queueing network as in Figure 1. The system consists of 12 overlay nodes and 3 underlay nodes (node 8, 9 and 13). All link capacities (including the links $5\to d$, $11\to d$ and $15\to d$) are 5. For simplicity, there is only a single class of traffic, and all packets can leave the system via any of the three sink nodes (5, 11 and 15) leading to the destination d.

At the beginning of each time slot, external packets arrive at nodes 1, 3 and 13 according to a uniform distribution



(a) All policies.



(b) Stabilizing policies.

Fig. 2. Queue backlog evolution under different policies for the 15-node network with stochastic dynamics.

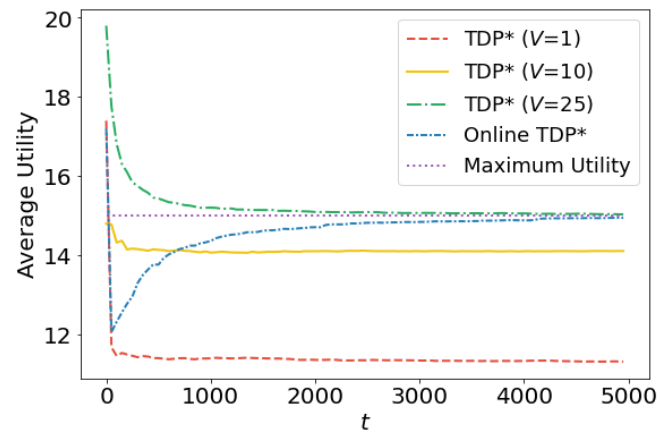


Fig. 3. Utility evolution under TDP* for the 15-node network with stochastic dynamics.

between 0 and 12, i.e.,

$$a_1(t), a_3(t), a_{13}(t) \sim \text{Unif}\{0, \cdots, 12\}.$$

Node 1 and 3 are overlay nodes and their packet admissions, denoted by γ_1 and γ_3 respectively, can be decided by the controller. Node 13 is uncontrollable and applies a greedy admission policy that admits all incoming packets, i.e., $\gamma_{13}(t) = a_{13}(t)$.

The controller then decides, for all overlay nodes, which neighbors to relay the buffered packets. The underlay nodes 8, 9 and 13 transmit packets on each outgoing link according to a uniform distribution between 0 and 5, i.e.,

$$\mu_{8\to 9}(t), \mu_{9\to 15}(t), \mu_{9\to 10}(t),$$
$$\mu_{13\to 9}(t), \mu_{13\to 12}(t), \mu_{13\to 14}(t) \sim \text{Unif}\{0, \cdots, 5\}.$$

We aim to maximize the throughput, i.e., $\sum_{t=0}^{T-1}\left(\gamma_1^\pi(t) + \gamma_3^\pi(t) + \gamma_{13}^\pi(t)\right)$. The expected number of external arrivals at each time slot is $12 \times 0.5 \times 3 = 18$ (packets), while the total service rate is $C_{5\to d} + C_{11\to d} + C_{15\to d} = 15$ (packets). Therefore, to keep the entire network rate stable, the controller cannot greedily admit packets.

In the simulation, we first compare the evolution of the queue backlog. We implemented TDP* with different parameter V's. We then implemented the online TDP* which uses the doubling trick to adjust the value of V's dynamically. We also directly applied the traditional Drift-plus-Penalty algorithm (with $V = 10$) to the overlay nodes as a baseline method. The results are in Figure 2.

From Figure 2a, we can see that under the traditional Drift-plus-Penalty algorithm, the average queue backlog grows linearly in time. Therefore, traditional Drift-plus-Penalty might not be capable of stabilizing the network. We then focus on the performance of stabilizing policies in Figure 2b. It can be seen that under different choice of V, all TDP* algorithms stabilize the system. The larger V is, the greater the queue backlog grows. For online TDP* with the doubling trick, the value of $V(t)$ grows when time elapsed doubles, which leads to the step increase in the curve.

We then compare the utility evolution in Figure 3.

Since the aggregated service capability is 15 packets per time slot, the maximum throughput that still keeps rate stability is also 15, which serves as an upper bound. From Figure 3, we can see that larger average utility can be achieved by choosing a larger V. If the time horizon $T = 5000$ is known in advance, the controller can choose $V = 25$ and obtain an almost optimal average utility. If the time horizon is unknown, online TDP* can be applied. As can be seen from the figure, online TDP* gradually converges to the maximum throughput. Note that from Figure 2b, the queue backlog also grows larger under online TDP*.

We finally study the trade-off between the queue backlog and the utility. We conducted experiments under different values of V, and obtained the final total queue backlog and the average utility for each V. The results are in Figure 4. From Figure 4, to achieve a greater utility, the controller needs to choose a larger V, which leads to larger queue backlog, which matches the results in Theorem 1.

B. 12-Node Network With Adversarial Dynamics

We study a 12-node queueing network as in Figure 5. The system consists of 8 overlay nodes and 4 underlay nodes (node 2, 3, 4 and 6). All link capacities (including the links $9 \to d$ and $12 \to d$) are 5. For simplicity, there is only a single class of traffic, and all packets can leave the system via either of the two sink nodes (9 and 12) leading to d.

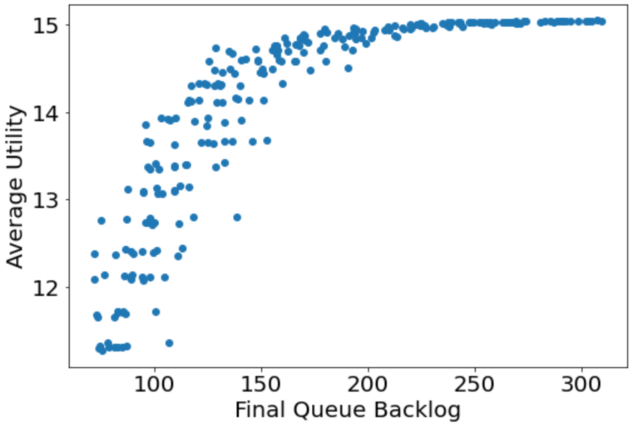


Fig. 4. The relationship between the queue backlog and the utility under TDP* for the 15-node network with stochastic dynamics.

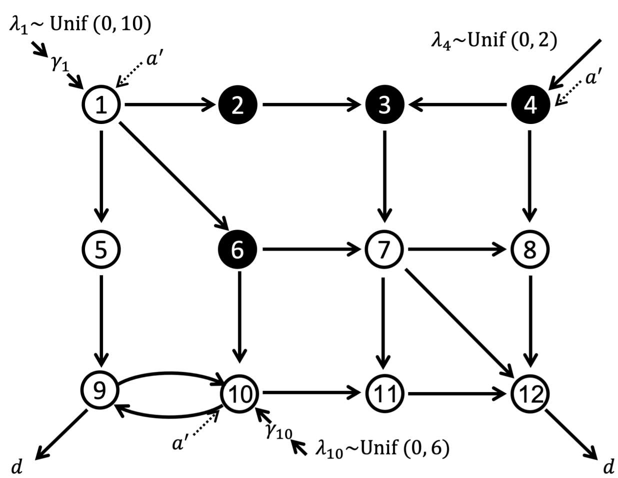


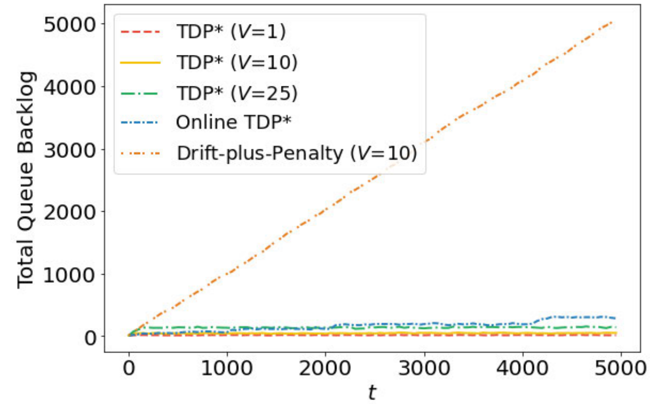
Fig. 5. The model of the 12-node network with adversarial dynamics.

At the beginning of each time slot, external packets arrive at nodes 1, 4 and 10 according to uniform distributions of

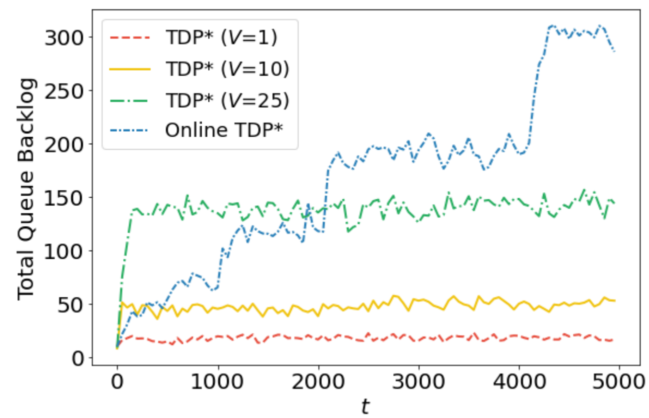
$$\begin{cases} a_1(t) \sim \text{Unif}\{0, \cdots, 10\} \\ a_4(t) \sim \text{Unif}\{0, 1, 2\} \\ a_{10}(t) \sim \text{Unif}\{0, \cdots, 6\} \end{cases}$$

Node 1 and 10 are overlay nodes and their packet admissions, denoted by γ_1 and γ_{10} respectively, can be decided by the controller. Node 4 is uncontrollable and applies a greedy admission policy that admits all incoming packets, i.e., $\gamma_4(t) = a_4(t)$. Moreover, an adversary attempts to inject at each time slot $a' = 2$ packets into the network through node 1, 4 or 10. In an attempt to destabilize the network, the adversary chooses to inject the a' packets into the node with the **largest** queue.

The controller then decides, for all overlay nodes, which neighbors to relay the buffered packets. Meanwhile, the underlay nodes, controlled by an adversary, try their best to destabilize the network. Node 4 and 6 apply the "join the longest queue" (JLQ) policy that transmits 5 packets to the neighboring node with the larger queue size and transmits nothing to the other neighboring node. JLQ, in contrast to the stabilizing "join the shortest queue" (JSQ) policy, is adversarial since the node with the larger queue is more heavily loaded and hence, easier to destabilize. Node 3 simply transmits 5 packets to node 7 at each time slot. Node 2 transmits 5 packets to node 3 for the first $T/2$ time slots, but starting at $T/2$, it only transmits 1 packet to node 3.



(a) All policies.



(b) Stabilizing policies.

Fig. 6. Queue backlog evolution under different policies for the 12-node network with adversarial dynamics.

The expected number of external arrivals at each time slot is $10 \times 0.5 + 2 \times 0.5 + 6 \times 0.5 + a' = 11$ (packets), while the total service rate is $C_{9\to d} + C_{12\to d} = 10$ (packets). Therefore, to keep the entire network stable, the controller cannot greedily admit packets. Moreover, starting at $T/2$, the service rate of node 2 drops sharply, which requires the algorithm to sense the change in time and alter the policy accordingly.

In the simulation, we first compare the evolution of the queue backlog. Similar to Section VI-A, we implemented TDP* with different parameter V's, the online TDP* which uses the doubling trick to adjust the value of V's dynamically, and the traditional Drift-plus-Penalty algorithm (with $V = 10$) to the overlay nodes. The results are shown in Figure 6.

From Figure 6, we can see that directly applying the traditional Drift-plus-Penalty algorithm cannot stabilize the network. Among the stability policies shown in Figure 6b, larger V's lead to larger queue backlogs.

We then compare the utility evolution in Figure 7. Since the aggregated service capability is 10 packets per time slot,

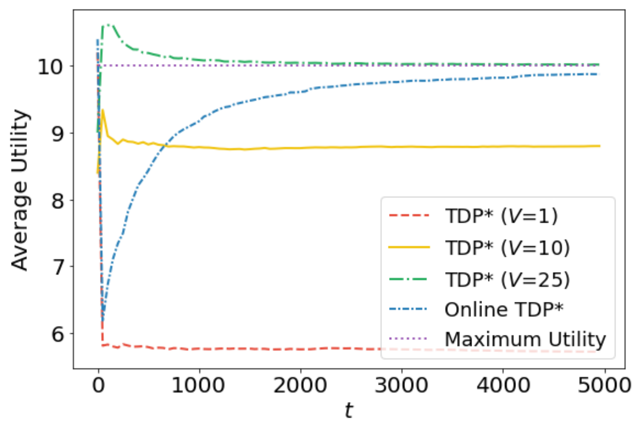


Fig. 7. Utility evolution under TDP* for the 12-node network with adversarial dynamics.

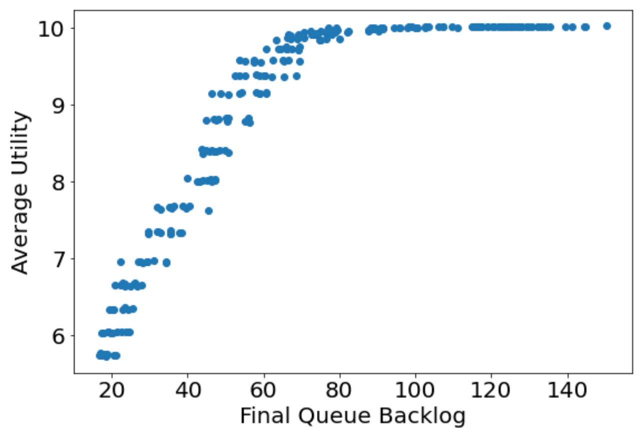


Fig. 8. The relationship between the queue backlog and the utility under TDP* for the 12-node network with adversarial dynamics.

the maximum throughput that still keeps rate stability is also 10, which serves as an upper bound. From Figure 7, larger throughput can be achieved by selecting larger V's. For the given time horizon $T = 5000$, an almost optimal throughput can be achieved by choosing $V = 25$. Similar to the result in Section VI-B, online TDP* also converges to the maximum throughput.

We finally study the relationship between queue backlog and utility by collecting the final total queue backlog and the average utility under different values of V, as depicted in Figure 8. From Figure 8, we can conclude that choosing larger value of V improves the utility, yet the queue backlogs also grow larger, which matches the results in Theorem 1.

VII. Conclusion

In this paper, we focus on networks with unobservable and uncontrollable nodes, under stochastic, non-stochastic and adversarial dynamics. We propose the TDP* algorithm that only needs to operate on overlay nodes with indirect state information. We rigorously derive the bounds on the utility gap and queue backlog, which explicitly reveal the trade-offs between utility and queue backlog. We further show that as long as the NUM problem is solvable, TDP* can maximize the network utility while keeping the queue backlogs stable.

A possible direction for future work is to focus on network inference, i.e., how can the controller develop general methods to estimate the states of the underlay nodes (e.g., queue backlog) more accurately and efficiently. Moreover, as a function of only network admissions, the network utility is a relatively limited objective. To optimize more general objective functions for networks (e.g., ones that capture certain end-to-end performance objectives) may be an interesting and important direction.

Appendix A
Proof of Lemma 1

We have

$$\begin{aligned}
&\sum_{i,k} Q_{ik}^{\pi_T}(T) \\
&= \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(T) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(T) + \sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T}(T) \\
&\leqslant \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(T) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(T) + \sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T+}(T) \\
&\leqslant \sqrt{KN + K|\mathcal{U}|} \\
&\quad \cdot \sqrt{\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T 2}(T) + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T 2}(T) + \sum_{i\in\mathcal{U},k} Y_{ik}^{\pi_T+2}(T)} \\
&= \sqrt{KN + K|\mathcal{U}|} \cdot \sqrt{2\Phi^{\pi_T}(T) + 2V \cdot \sum_{t=0}^{T-1} U\big(\gamma(t)\big)} \\
&\leqslant \sqrt{2KN\Phi^{\pi_T}(T) + 2K^2N^2DVT}, \qquad (27)
\end{aligned}$$

where the second inequality utilizes Cauchy–Schwarz inequality, the second equation holds by inserting the definition of $\Phi(T)$ as in (7), and the last equation holds by using $|\mathcal{U}| \leqslant N$ and $U_{ik}(t) \leqslant D$.

Appendix B
Proof of Lemma 2

We first upper bound $Q_{ik}^2(t+1) - Q_{ik}^2(t)$ for $i \in \mathcal{O}$. Writing down the update rule for $Q_{ik}^2(t)$, we have that

$$\begin{aligned}
Q_{ik}(t+1) &= \Big[Q_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} f_{ijk}(t)\Big]^+ \\
&\quad + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t) \\
&\leqslant \Big[Q_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} f_{ijk}(t)\Big]^+ \\
&\quad + \sum_{j\in\mathcal{O}} f_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t).
\end{aligned}$$

It is easy to show that for $x, y, z \geqslant 0$, the inequality

$$\left([x-y]^+ + z\right)^2 \leqslant x^2 + y^2 + z^2 + 2x(z-y)$$

holds. By replacing x with $Q_{ik}(t) + \gamma_{ik}(t)$, y with $\sum_{j\in\mathcal{N}} f_{ijk}(t)$ and z with $\sum_{j\in\mathcal{O}} f_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t)$, we upper bound $Q_{ik}^2(t+1)$ as

$$\begin{aligned} Q_{ik}^2(t+1) \leqslant\, & Q_{ik}^2(t) + \Big(\sum_{j\in\mathcal{N}} f_{ijk}(t)\Big)^2 \\ & + \Big(\sum_{j\in\mathcal{O}} f_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t)\Big)^2 \\ & + 2\gamma_{ik}(t)\delta Q_{ik}(t) + 2Q_{ik}(t)\delta Q_{ik}(t) \\ \leqslant\, & Q_{ik}^2(t) + 2Q_{ik}(t)\delta Q_{ik}(t) + 6N^2D^2, \end{aligned} \tag{28}$$

where the last inequality holds by utilizing (3).

We then upper bound $X_{ik}^2(t+1) - X_{ik}^2(t)$ for $i\in\mathcal{U}$. With

$$\begin{aligned} X_{ik}(t+1) =\, & \Big[X_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} g_{ijk}(t)\Big]^+ \\ & + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) + \sum_{j\in\mathcal{U}} g_{jik}(t) \\ \leqslant\, & \Big[X_{ik}(t) + \gamma_{ik}(t) - \sum_{j\in\mathcal{N}} g_{ijk}(t)\Big]^+ \\ & + \sum_{j\in\mathcal{O}} f_{jik}(t) + \sum_{j\in\mathcal{U}} g_{jik}(t), \end{aligned}$$

by applying similar techniques as (28), we have

$$X_{ik}^2(t+1) \leqslant X_{ik}^2(t) + 2X_{ik}(t)\delta X_{ik}(t) + 6N^2D^2. \tag{29}$$

Appendix C
Proof of Lemma 3

and the erroneous estimates of $Y_{ik}(t)$ as $\tilde{Y}_{ik}(t) = \hat{Y}_{ik}(t) + \epsilon_{ik}(\tau_{ik}(t))$

To avoid confusion, we define that $\Delta Y_{ik}^+(t) \triangleq Y_{ik}^+(t+1) - Y_{ik}^+(t)$. Both $\Delta Y_{ik}(t)$ and $\Delta Y_{ik}^+(t)$ are bounded as the following lemma (see Appendix G for the proof).

Lemma 7: For each $i\in\mathcal{U}$, $t = 0,\cdots,T-1$ and k, we have

$$-2ND \leqslant \Delta Y_{ik}(t), \Delta Y_{ik}^+(t) \leqslant 2ND,$$

Since $Y_{ik}^{+2}(t+1) - Y_{ik}^{+2}(t)$ can be decomposed as

$$\begin{aligned} & Y_{ik}^{+2}(t+1) - Y_{ik}^{+2}(t) \\ & = \Big(Y_{ik}^+(t) + \Delta Y_{ik}^+(t)\Big)^2 - Y_{ik}^{2+}(t) \\ & = 2Y_{ik}^+(t)\Delta Y_{ik}^+(t) + \Big(\Delta Y_{ik}^+(t)\Big)^2, \end{aligned} \tag{30}$$

upper bounding $Y_{ik}^+(t)\Delta Y_{ik}^+(t)$ suffices and we have that

$$\begin{aligned} & Y_{ik}^+(t)\Delta Y_{ik}^+(t) \\ & \leqslant Y_{ik}^+(t)\cdot\max\{\Delta Y_{ik}(t), -Y_{ik}^+(t)\} \\ & = Y_{ik}^+(t)\Delta Y_{ik}(t) + \max\{0, -Y_{ik}^{+2}(t) - Y_{ik}^+(t)\Delta Y_{ik}(t)\} \\ & \leqslant Y_{ik}^+(t)\Delta Y_{ik}(t) + \max\{0, -Y_{ik}^{+2}(t) + 2NDY_{ik}^+(t)\} \\ & = Y_{ik}^+(t)\Delta Y_{ik}(t) + \max\{0, -(Y_{ik}^{+2}(t) - ND)^2 + N^2D^2\} \\ & \leqslant Y_{ik}^+(t)\Delta Y_{ik}(t) + N^2D^2, \end{aligned} \tag{31}$$

where the first inequality comes from the fact that $Y_{ik}^+(t) \geqslant 0$ and $\Delta Y_{ik}^+(t) \leqslant \max\{\Delta Y_{ik}(t), -Y_{ik}^+(t)\}$. The second inequality holds because $Y_{ik}^+(t)\geqslant 0$ and $\Delta Y_{ik}(t) \geqslant -2ND$.

By inserting (31) into (30) and utilizing Lemma 7, we have that

$$\begin{aligned} & Y_{ik}^{+2}(t+1) - Y_{ik}^{+2}(t) \\ & \leqslant 2Y_{ik}^+(t)\Delta Y_{ik}(t) + \big(\Delta Y_{ik}^+(t)\big)^2 + 2N^2D^2 \\ & \leqslant 2Y_{ik}^+(t)\Delta Y_{ik}(t) + 6N^2D^2 \\ & = 2\hat{Y}_{ik}^+(t)\Delta Y_{ik}(t) + 6N^2D^2 + 2\big(Y_{ik}^+(t) - \hat{Y}_{ik}^+(t)\big)\cdot\Delta Y_{ik}(t) \\ & \leqslant 2\hat{Y}_{ik}^+(t)\Delta Y_{ik}(t) + 6N^2D^2 \\ & \quad + 2\big((t-\tau_{ik}(t))\cdot 2ND + |\epsilon_{ik}(t)|\big)\cdot 2ND \\ & \leqslant 2\hat{Y}_{ik}^+(t)\Delta Y_{ik}(t) + \big(8L(t)+6\big)N^2D^2 + 4ND|\epsilon_{ik}(\tau_{ik}(t))|, \end{aligned}$$

which completes the proof.

Appendix D
Proof of Lemma 4

Since π_T is obtained by solving (10), i.e. $(\boldsymbol{\gamma}_o, \mathbf{f}, \mathbf{g}) = (\boldsymbol{\gamma}_o^{\pi_T}(t), \mathbf{f}^{\pi_T}(t), \mathbf{g}^{\pi_T}(t))$ minimizes (10), substituting $(\boldsymbol{\gamma}_o, \mathbf{f}, \mathbf{g}) = (\boldsymbol{\gamma}_o^{\pi*}(t), \tilde{\mathbf{f}}^{\pi*}(t), \tilde{\boldsymbol{\mu}}(t))$ would result in sub-optimal objective. That is:

$$\begin{aligned} & \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\cdot\Big[\gamma_{ik}^{\pi_T}(t) - \sum_{j\in\mathcal{N}} f_{ijk}^{\pi_T}(t) + \sum_{j\in\mathcal{O}} f_{jik}^{\pi_T}(t) \\ & \quad + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t)\Big] \\ & \quad + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\cdot\Big[\sum_{j\in\mathcal{O}} f_{jik}^{\pi_T}(t) + \sum_{j\in\mathcal{U}} g_{jik}^{\pi_T}(t) - \sum_{j\in\mathcal{N}} g_{ijk}^{\pi_T}(t)\Big] \\ & \quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T +}(t)\cdot\Big[\min\Big\{X_{ik}^{\pi_T}(t) + \gamma_{ik}(t), \sum_{j\in\mathcal{N}} g_{ijk}^{\pi_T}(t)\Big\} \\ & \quad - \sum_{j\in\mathcal{U}} g_{jik}^{\pi_T}(t)\Big] - V\cdot U(\gamma^{\pi_T}(t)) \\ & \leq \sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\cdot\Big[\gamma_{ik}^{\pi*}(t) - \sum_{j\in\mathcal{N}} \tilde{f}_{ijk}^{\pi*}(t) + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}^{\pi*}(t) \\ & \quad + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t)\Big] \\ & \quad + \sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\cdot\Big[\sum_{j\in\mathcal{O}} \tilde{f}_{jik}^{\pi*}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t) - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t)\Big] \\ & \quad + \sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T +}(t)\cdot\Big[\min\Big\{X_{ik}^{\pi_T}(t) + \gamma_{ik}(t), \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t)\Big\} \\ & \quad - \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t)\Big] - V\cdot U(\gamma^{\pi*}(t)). \end{aligned} \tag{32}$$

We now conduct the following operations on both sides of (32):

- Add $\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\cdot\gamma_{ik}(t)$
- Add $\sum_{i\in\mathcal{U},k} \hat{Y}_{ik}^{\pi_T +}(t)\cdot\big(\sum_{j\in\mathcal{U}}\tilde{\mu}_{jik}(t) - \sum_{j\in\mathcal{N}}\tilde{\mu}_{ijk}(t)\big)$

After the operations, by using the notions of $\delta Q_{ik}^{\pi_T}(t), \delta X_{ik}^{\pi_T}(t)$ and $\Delta Y_{ik}^{\pi_T}(t)$ defined in Section III-B, and $\Delta Q_{ik}^{\pi*}(t), \delta X_{ik}^{\pi*}(t)$ and $\Delta Y_{ik}^{\pi*}(t)$ defined in Section IV, the inequality (32) is equivalent to

$$\begin{aligned}&\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\delta Q_{ik}^{\pi_T}(t)+\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\delta X_{ik}^{\pi_T}(t)\\&\quad+\sum_{i\in\mathcal{U},k}\hat{Y}_{ik}^{\pi_T+}(t)\Delta Y_{ik}^{\pi_T}(t)-V\cdot U(\gamma^{\pi_T}(t))\\&\leqslant\sum_{i\in\mathcal{O},k} Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi *}(t)+\sum_{i\in\mathcal{U},k} X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi *}(t)\\&\quad+\sum_{i\in\mathcal{U},k}\hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta}Y_{ik}^{\pi *}(t)-V\cdot U(\gamma^{\pi *}(t)),\end{aligned}$$

which completes the proof.

Appendix E
Proof of Lemma 5

From the queue dynamics and the definitions of $\Delta Q_{ik}^{\pi *}(t)$ and $\Delta X_{ik}^{\pi *}(t)$ in Section IV, we have for $i\in\mathcal{O}$:

$$\begin{aligned}Q_{ik}^{\pi *}(t+1)&=\Big(Q_{ik}^{\pi *}(t)+\gamma_{ik}^{\pi *}(t)-\sum_{j\in\mathcal{N}} f_{ijk}^{\pi *}(t)\Big)^{+}\\&\quad+\sum_{j\in\mathcal{O}}\tilde{f}_{jik}^{\pi *}(t)+\sum_{j\in\mathcal{U}}\tilde{\mu}_{jik}(t)\\&=Q_{ik}^{\pi *}(t)+\gamma_{ik}^{\pi *}(t)-\sum_{j\in\mathcal{N}}\tilde{f}_{ijk}^{\pi *}(t)\\&\quad+\sum_{j\in\mathcal{O}}\tilde{f}_{jik}^{\pi *}(t)+\sum_{j\in\mathcal{U}}\tilde{\mu}_{jik}(t)\\&=Q_{ik}^{\pi *}(t)+\Delta Q_{ik}^{\pi *}(t),\end{aligned}\tag{33}$$

and for $i\in\mathcal{U}$,

$$\begin{aligned}Q_{ik}^{\pi *}(t+1)&=\Big(Q_{ik}^{\pi *}(t)+\gamma_{ik}(t)-\sum_{j\in\mathcal{N}}\mu_{ijk}(t)\Big)^{+}\\&\quad+\sum_{j\in\mathcal{O}}\tilde{f}_{jik}^{\pi *}(t)+\sum_{j\in\mathcal{U}}\tilde{\mu}_{jik}(t)\\&=Q_{ik}^{\pi *}(t)+\gamma_{ik}(t)-\sum_{j\in\mathcal{N}}\tilde{\mu}_{ijk}(t)\\&\quad+\sum_{j\in\mathcal{O}}\tilde{f}_{jik}^{\pi *}(t)+\sum_{j\in\mathcal{U}}\tilde{\mu}_{jik}(t)\\&=Q_{ik}^{\pi *}(t)+\Delta X_{ik}^{\pi *}(t).\end{aligned}\tag{34}$$

We define $M\triangleq T \bmod H$ and there exists an integer J such that $T=JH+M$. Next, we provide bounds on the multi-slot changes of the queue backlogs under π^*.

From (33) and by telescoping, we have for $i\in\mathcal{O}$:

$$\sum_{t=jH}^{(j+1)H-1}\Delta Q_{ik}^{\pi *}(t)=Q_{ik}^{\pi *}((j+1)H)-Q_{ik}^{\pi *}(jH),$$

which leads to

$$|\sum_{t=jH}^{(j+1)H-1}\Delta Q_{ik}^{\pi *}(t)|\leqslant\max\left\{Q_{ik}^{\pi *}((j+1)H),Q_{ik}^{\pi *}(jH)\right\}\tag{35}$$

$$\leqslant\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t).\tag{36}$$

Similarly, from (34) and by telescoping, we have for $i\in\mathcal{U}$:

$$|\sum_{t=jH}^{(j+1)H-1}\Delta X_{ik}^{\pi *}(t)|\leqslant\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t).\tag{37}$$

Now, back to the main proof, we have the following decomposition for $i\in\mathcal{O}$ and $k\in\mathcal{K}$,

$$\begin{aligned}&\sum_{t=0}^{T-1}Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi *}(t)\\&=\sum_{j=0}^{J-1}\Bigg[Q_{ik}^{\pi_T}(jH)\sum_{t=jH}^{(j+1)H-1}\Delta Q_{ik}^{\pi *}(t)\\&\quad+\sum_{t=jH}^{(j+1)H-1}\big(Q_{ik}^{\pi_T}(t)-Q_{ik}^{\pi_T}(jH)\big)\cdot\Delta Q_{ik}^{\pi *}(t)\Bigg]\\&\quad+\sum_{t=JH}^{T-1}Q_{ik}^{\pi_T}(t)\Delta Q_{ik}^{\pi *}(t)\\&\leqslant\sum_{j=0}^{J-1}\Bigg[2NDT\sum_{t=jH}^{(j+1)H-1}\Delta Q_{ik}^{\pi *}(t)\\&\quad+\sum_{t=jH}^{(j+1)H-1}2NDH\cdot 2ND\Bigg]+M\cdot 2NDT\cdot 2ND\\&\overset{(36)}{\leqslant}\sum_{j=0}^{J-1}2NDT\cdot\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t)+8N^2D^2HT\\&=2JNDT\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t)+8N^2D^2HT\\&\leqslant\frac{2NDT^2}{H}\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t)+8N^2D^2HT,\end{aligned}\tag{38}$$

where inequalities hold by using (3), and the fact that $M\leqslant H$ and $J\leqslant T/H$.

Similarly, we show that for $i\in\mathcal{U}$ and $k\in\mathcal{K}$,

$$\begin{aligned}&\sum_{t=0}^{T-1}X_{ik}^{\pi_T}(t)\Delta X_{ik}^{\pi *}(t)\\&\quad\leqslant\frac{2NDT^2}{H}\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t)+8N^2D^2HT.\end{aligned}\tag{39}$$

Summing up (38) and (39) over all nodes and traffic classes, we have

$$\begin{aligned}&\sum_{t=0}^{T-1}\sum_{i\in\mathcal{O},k}Q_{ik}^{\pi_T}(t)\delta Q_{ik}^{\pi *}(t)+\sum_{t=0}^{T-1}\sum_{i\in\mathcal{U},k}X_{ik}^{\pi_T}(t)\delta X_{ik}^{\pi *}(t)\\&\quad\leqslant\frac{4KN^2DT^2}{H}\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t)+16KN^3D^2\ HT.\end{aligned}$$

Taking $H=c\sqrt{\frac{T\cdot\max_{0\leqslant t\leqslant T}\sum_{i,k}Q_{ik}^{\pi *}(t)}{ND}}$ where c is any positive constant that makes H an integer completes the proof.

Appendix F
Proof of Lemma 6

From the definition of $\widetilde{\Delta} Y_{ik}^{\pi*}(t)$ in Section IV, we have for $i \in \mathcal{U}$,

$$\widetilde{\Delta} Y_{ik}^{\pi*}(t) = \min\Big\{X_{ik}^{\pi_T}(t)+\gamma_{ik}(t), \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t)\Big\} - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t)$$
$$\leqslant \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t) - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t) = 0$$

Therefore, we have for $i \in \mathcal{U}$,

$$\hat{Y}_{ik}^{\pi_T+}(t)\widetilde{\Delta} Y_{ik}^{\pi*}(t) \leqslant 0$$

Summing up the above inequality from $t = 0$ to T over $i \in \mathcal{U}$ and $k \in \mathcal{K}$ completes the proof.

Appendix G
Proof of Lemma 7

Here we fix an i and a t arbitrarily. We first discuss the range of $\Delta Y_{ik}(t)$. From the definition of $\Delta Y_{ik}(t)$, we have

$$\begin{aligned}
&\Delta Y_{ik}(t)\\
&= Q_{ik}(t+1) - Q_{ik}(t) - \big(X_{ik}(t+1) - X_{ik}(t)\big)\\
&= a_{ik}(t) - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t) + \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) + \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t)\\
&\quad - a_{ik}(t) + \sum_{j\in\mathcal{N}} \tilde{g}_{ijk}(t) - \sum_{j\in\mathcal{O}} \tilde{f}_{jik}(t) - \sum_{j\in\mathcal{U}} g_{jik}(t)\\
&= \sum_{j\in\mathcal{U}} \tilde{\mu}_{jik}(t) - \sum_{j\in\mathcal{N}} \tilde{\mu}_{ijk}(t) + \sum_{j\in\mathcal{N}} \tilde{g}_{ijk}(t) - \sum_{j\in\mathcal{U}} g_{jik}(t).
\end{aligned}$$

By applying (3), we have

$$-2ND \leqslant \Delta Y_{ik}(t) \leqslant 2ND. \tag{40}$$

With (40) at hand, we first have

$$\begin{aligned}
\Delta Y_{ik}^{+}(t) &= \max\{Y_{ik}(t+1), 0\} - Y_{ik}^{+}(t)\\
&= \max\{Y_{ik}(t+1) - Y_{ik}^{+}(t), -Y_{ik}^{+}(t)\}\\
&\leqslant \max\{Y_{ik}(t+1) - Y_{ik}(t), -Y_{ik}^{+}(t)\}\\
&= \max\{\Delta Y_{ik}(t), -Y_{ik}^{+}(t)\} \leqslant 2ND.
\end{aligned} \tag{41}$$

For the lower bound $Y_{ik}^{+}(t)$, we have

$$\begin{aligned}
\Delta Y_{ik}^{+}(t) &= Y_{ik}^{+}(t+1) - \max\{Y_{ik}(t), 0\}\\
&= \min\{Y_{ik}^{+}(t+1) - Y_{ik}(t), Y_{ik}^{+}(t+1)\}\\
&\geqslant \min\{Y_{ik}(t+1) - Y_{ik}(t), Y_{ik}^{+}(t+1)\}\\
&= \min\{\Delta Y_{ik}(t), Y_{ik}^{+}(t+1)\} \geqslant -2ND.
\end{aligned} \tag{42}$$

Combining (40), (41) and (42) completes the proof.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *Proc. 29th IEEE Conf. Decis. Control*, Dec. 1990, pp. 2130–2132.

[2] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, Apr. 2008.

[3] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain, "Overlay networks: An Akamai perspective," *Adv. Content Del., Streaming, Cloud Services*, vol. 51, no. 4, pp. 305–328, 2014.

[4] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[5] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: Classification and state-of-the-art," *Comput. Netw.*, vol. 44, no. 5, pp. 643–666, Apr. 2004.

[6] W. G. Halfond, J. Viegas, and A. Orso, "A classification of SQL-injection attacks and countermeasures," in *Proc. IEEE Int. Symp. Secure Softw. Eng.*, Mar. 2006, vol. 1, pp. 13–15.

[7] G. S. Paschos and E. Modiano, "Throughput optimal routing in overlay networks," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2014, pp. 401–408.

[8] N. M. Jones, G. S. Paschos, B. Shrader, and E. Modiano, "An overlay architecture for throughput optimal multipath routing," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2615–2628, Oct. 2017.

[9] A. Rai, R. Singh, and E. Modiano, "A distributed algorithm for throughput optimal routing in overlay networks," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, May 2019, pp. 1–9.

[10] Q. Liang and E. Modiano, "Optimal network control in partially-controllable networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 397–405.

[11] E. J. Sondik, "The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs," *Oper. Res.*, vol. 26, no. 2, pp. 282–304, Apr. 1978.

[12] H.-T. Cheng, "Algorithms for partially observable Markov decision processes," Ph.D. thesis, Univ. British Columbia, 1988.

[13] N. L. Zhang and W. Liu, "Planning in stochastic domains: Problem characteristics and approximation," Hong Kong Univ. Sci. Technol., Tech. Rep. HKUST-CS96-31, 1996.

[14] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, nos. 1–2, pp. 99–134, May 1998.

[15] A. R. Cassandra, M. L. Littman, and N. Lianwen Zhang, "Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes," 2013, *arXiv:1302.1525*.

[16] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *J. Artif. Intell. Res.*, vol. 15, pp. 319–350, Nov. 2001.

[17] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson, "Adversarial queueing theory," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, 1996, pp. 376–385.

[18] M. Andrews and L. Zhang, "Scheduling over nonstationary wireless channels with finite rate sets," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1694–1704.

[19] M. Andrews and L. Zhang, "Scheduling over a time-varying user-dependent channel with applications to high-speed wireless data," *J. ACM*, vol. 52, no. 5, pp. 809–834, Sep. 2005.

[20] Q. Liang and E. Modiano, "Minimizing queue length regret under adversarial network models," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 1, pp. 31–32, Jan. 2019.

[21] B. Liu and E. Modiano, "Optimal control for networks with unobservable malicious nodes," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 49, no. 3, pp. 18–19, 2021.

[22] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, Jan. 1997.

[23] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Apr. 1998.

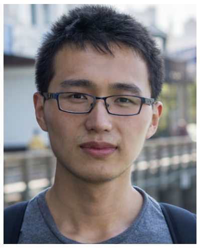
[24] S. H. Low and D. E. Lapsely, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[25] Q. Liang and E. Modiano, "Network utility maximization in adversarial environments," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 594–602.

[26] Q. Liang and E. Modiano, "Optimal network control with adversarial uncontrollable nodes," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2019, pp. 101–110.

[27] J. Han, D. Watson, and F. Jahanian, "Topology aware overlay networks," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2005, pp. 2554–2565.

[28] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 29–40, Jan. 2004.



Bai Liu received the B.E. degree (Hons.) from Tsinghua University, Beijing, China, in 2017, and the M.S. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2019, where he is currently pursuing the Ph.D. degree with the Laboratory for Information and Decision Systems. His current research interests include learning and control problems in networked systems, with the application of reinforcement learning, stochastic optimization, and inference methods.

Quang Minh Nguyen (Graduate Student Member, IEEE) received the dual B.S. degree (Hons.) in computer science and applied mathematics from the National University of Singapore, Singapore, in 2020, and the M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2022, where he is currently pursuing the Ph.D. degree with the Laboratory for Information and Decision Systems. His current research interests include the optimal control of networked systems, with a focus on wireless scheduling and software-defined networking, machine learning, information theory, and stochastic optimization.

Qingkai Liang received the B.E. degree (Hons.) in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2015 and 2018, respectively. He is currently the Co-Founder of Celer Network. His current research interests include learning and control problems that arise in networked systems, especially on online learning algorithms in adversarial networks, which have been successfully applied in Raytheon BBN Technologies and Bell Labs.

Eytan Modiano (Fellow, IEEE) received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 1989 and 1992, respectively.

He is currently the Richard C. Maclaurin Professor with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS), MIT. Prior to joining the Faculty at MIT in 1999, he was a Naval Research Laboratory Fellow from 1987 to 1992, a National Research Council Post-Doctoral Fellow from 1992 to 1993, and a member of the Technical Staff with the MIT Lincoln Laboratory from 1993 to 1999. His current research interests include the modeling, analysis, and design of communication networks and protocols. He is an Associate Fellow of AIAA. He had served on the IEEE Fellows Committee in 2014 and 2015. He received the Infocom Achievement Award in 2020 for contributions to the analysis and design of cross-layer resource allocation algorithms for wireless, optical, and satellite networks. He was a co-recipient of the 2018 Infocom Best Paper Award, the 2018 MobiHoc Best Paper Award, the 2016 MobiHoc Best Paper Award, the 2013 Wiopt Best Paper Award, and the 2006 Sigmetrics Best Paper Award. He was the Technical Program Co-Chair for IEEE Wiopt in 2006, IEEE Infocom in 2007, ACM MobiHoc in 2007, and DRCN in 2015. He was the General Co-Chair of Wiopt in 2021. He was the Editor-in-Chief of IEEE/ACM TRANSACTIONS ON NETWORKING from 2017 to 2020 and served as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE/ACM TRANSACTIONS ON NETWORKING.