Minimizing Queue Length Regret Under Adversarial Network Models

QINGKAI LIANG and EYTAN MODIANO, MIT LIDS

Stochastic models have been dominant in network optimization theory for over two decades, due to their analytical tractability. However, these models fail to capture non-stationary or even adversarial network dynamics which are of increasing importance for modeling the behavior of networks under malicious attacks or characterizing short-term transient behavior. In this paper, we focus on minimizing queue length regret under adversarial network models, which measures the finite-time queue length difference between a causal policy and an "oracle" that knows the future. Two adversarial network models are developed to characterize the adversary's behavior. We provide lower bounds on queue length regret under these adversary models and analyze the performance of two control policies (i.e., the MaxWeight policy and the Tracking Algorithm). We further characterize the stability region under adversarial network models, and show that both the MaxWeight policy and the Tracking Algorithm are throughput-optimal even in adversarial settings.

ACM Reference Format:
Qingkai Liang and Eytan Modiano. 2018. Minimizing Queue Length Regret Under Adversarial Network Models. *Proc. ACM Meas. Anal. Comput. Syst.* 2, 1, Article 11 (March 2018), 34 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Background and Motivation

Stochastic network models have been dominant in network optimization theory for over two decades, due to their analytical tractability. For example, it is often assumed in wireless networks that the variation of traffic patterns and the evolution of channel capacity follow some stationary stochastic process, such as the i.i.d. model and the ergodic Markov model. Many important network control policies (e.g., MaxWeight policy [16]) have been derived to optimize network performance (e.g., throughput) under those stochastic network dynamics.

However, non-stationary or even adversarial dynamics have been of increasing importance in recent years. For example, modern communication networks frequently suffer from Distributed Denial-of-Service (DDoS) attacks or jamming attacks [17], where traffic injections and channel conditions are controlled by some malicious entity in order to degrade network performance. As a result, it is important to develop efficient control policies that optimize network performance even in adversarial settings. However, extending the traditional stochastic network optimization framework to the adversarial setting is non-trivial because

This work was supported by NSF Grant CNS-1524317 and by DARPA I2O and Raytheon BBN Technologies under Contract No. HROO l l-l 5-C-0097.
Authors' address: Qingkai Liang; Eytan Modiano, MIT LIDS.

2476-1249/2018/3-ART11 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

many important notions and analytical tools developed for stochastic networks cannot be applied in adversarial settings. For example, the traditional stochastic network optimization focuses on long-term network performance while in an adversarial environment the network may not have any steady state or well-defined long-term time averages. Thus, typical steady-state analysis and many equilibrium-based notions such as the network throughput region cannot be used in networks with adversarial dynamics, and it is important to understand "transient" network performance within a finite time horizon in a non-stationary/adversarial environment.

In this paper, we investigate efficient network control policies that can optimize network performance (i.e., queue length) within a finite time horizon in an adversarial environment.

1.2 Main Results

We develop general adversarial network models and propose a new finite-time performance metric, referred to as *queue length regret* (the formal definition is given in Section 2.3):

$$\mathcal{R}_T^{\pi} = \sum_i Q_i^{\pi}(T) - \sum_i Q_i^{*}(T),$$

where $\sum_i Q_i^{\pi}(T)$ is the total queue length achieved by control policy π after a finite time horizon T, and $\sum_i Q_i^{*}(T)$ is the minimum queue length achieved by some "oracle" that has perfect knowledge about the future.

We first prove that it is impossible to achieve "low" queue length regret if the adversary is unconstrained. In particular, there exist some adversarial network dynamics such that the queue length regret grows at least linearly with the time horizon T under any causal control policy. This impossibility result motivates us to study constrained adversarial dynamics.

We then study two adversarial network models where the network dynamics are constrained to some "admissible" set. In particular, we first consider the (W, ϵ)-constrained adversary model, where the total arrivals are less than $(1-\epsilon)$ times of the total services within any window of W slots. Although this window-based model is relatively limited, it is widely used by existing works (e.g., [2–4, 9, 12]) due to its analytical tractability and serves as a foundation for understanding more generalized adversary models.

Observing the limitation of (W, ϵ)-constrained model. we then propose a more generalized V_T-constrained model, where the total queue length generated by the "oracle" during its sample path is upper bounded by V_T. By varying the values V_T, the proposed V_T-constrained adversary model can cover a wide range of adversarial settings: from a strictly constrained adversary to a fully unconstrained adversary.

Under the above two adversary models, we develop lower bounds on queue length regret. It is shown that no causal policy can achieve sublinear queue length regret if W or V_T grows linearly with T. We also analyze the queue length regret of two control algorithms: the MaxWeight policy [16] and the Tracking Algorithm [3, 4] under the two adversarial models. In particular, both the MaxWeight policy and the Tracking Algorithm achieve sublinear queue length regret whenever W or V_T grows sublinearly with T, yet the theoretical regret bound under the Tracking Algorithm is better than that under the MaxWeight policy. The Tracking Algorithm is also asymptotically regret-optimal under the (W, ϵ)-constrained adversary model. We summarize these results in Table 1.

Finally, based on the above analytical results and the observation that sublinear queue length regret is equivalent rate stability, we characterize the stability region under adversarial network models, and show that both the MaxWeight policy and the Tracking Algorithm are throughput-optimal even in adversarial settings.

Table 1. Queue Length Regret Bounds

	(W, ϵ)-Constrained Adversary	**V_T-Constrained Adversary**
Lower Bound	$\Omega(W)$	$\Omega(V_T)$
MaxWeight	$O(\sqrt{TW})$ if $\epsilon = 0$ $O(W/\epsilon^3)$ if $\epsilon > 0$	$O(V_T^{1/3}T^{2/3})$
Tracking Alg.	$O(W)$	$O(\sqrt{TV_T})$

1.3 Related Work

The study of adversarial network models dates back more than two decades ago. Rene Cruz [7] provided the first concrete example of networks with adversarial dynamics, which were later generalized by Borodin *et al.* [5] under the *Adversarial Queuing Theory* (AQT) framework. In AQT, in each time slot, the adversary injects a set of packets at some of the nodes. In order to avoid trivially overloading the system, the AQT framework imposes a stringent *window constraints*: the maximum traffic injected in every link over any window of W time slots should not exceed the amount of traffic that the link can serve during that interval. Andrews *et al.* [1] introduced a more generalized adversary model known as the *Leaky Bucket* (LB) model that differs from AQT by allowing some traffic burst during any time interval. The AQT model and the LB model have given rise to a large number of results since their introduction, most of which are about network stability under several simple scheduling policies such as FIFO (see [6] for a review of these results).

However, the AQT and the LB models assume that only packet injections are adversarial while the underlying network topology and link states remain fixed. Such a static network model does not capture many adversarial environments, such as wireless networks under jamming attacks where the adversary can control the channel states. Andrews and Zhang [3, 4] extended the AQT model to single-hop dynamic wireless networks, where both packets injections and link states are controlled by an adversary, and prove the stability of the MaxWeight algorithm in this context. Jung *et al.* [2, 9] further extended the results of [3, 4] to multi-hop dynamic networks. Our window-based (W, ϵ)-constrained model is inspired by and similar to the adversarial models used in [2–4, 9]. Moreover, we also develop a new V_T-constrained model that relaxes the window constraints and generalizes the existing window-based (W, ϵ)-constrained models.

Recently, Paschos and Tassiulas [14] considered the problem of stabilizing queues under a mixture of stochastic and adversarial traffic injections, but their results is limited to a very specific service provisioning model and only traffic injections are adversarial.

While the above-mentioned works focused on network stability, Neely [12] investigated the universal network utility maximization problem where network utility needs to be maximized subject to stability constraints under adversarial network dynamics. Algorithm (time-average) performance is measured with respect to a so-called "W-slot look-ahead policy". Such a policy has perfect knowledge about network dynamics over the next W slots but it is required that under this policy the total arrivals to each queue should not exceed the total amount of service offered to that queue during every window of W slots. As a result, it is similar to our (W, ϵ)-constrained model where stringent window constraints have to be enforced. In this paper, we not only considers the (W, ϵ)-constrained model but also develop a more general V_T-constrained model that gets rid of the window constraints.

In addition, Shakkottai *et al.* [8] also used the notion of queue regret in the multiarmed bandit problem. However, their analysis is intended for stochastic environments and cannot be carried over to adversarial environments.

In summary, our paper expands previous work in a number of fundamental ways.

First, we develop queue length regret lower bounds under both the (W, ϵ)-constrained and the V_T-constrained models. As far as we know, none of the existing works (e.g., [2–4, 9, 12]) provide lower bounds on queue length regret (or queue length), even under the restrictive (W, ϵ)-constrained model where stringent window constraints are imposed. Note that our lower bounds on queue length regret reveal fundamental limits of the system. For example, our lower bound under the V_T-constrained model reveals that if $V_T = \Omega(T)$, then no causal policy can stabilize the network even if there exists some stabilizing non-causal policy. Moreover, these lower bounds are also critical to establishing the optimality of the Tracking algorithm under the (W, ϵ)-constrained model.

Second, we provide analysis under the new V_T-constrained adversary model which generalizes the adversarial network dynamics models used by existing works. As far as we know, existing works (e.g., [2–4, 9, 12]) all use the (W, ϵ)-constrained adversary model or similar windows-based variants due to its analytical tractability. In this paper, we propose a new V_T-constrained adversary model which gets rid of the window constrains and covers the full spectrum of network dynamics. Due to the lack of window-based structure, the analysis carried out in existing works cannot be applied to the V_T-constrained model. In this paper, we develop queue length regret upper bounds for the MaxWeight policy and the Tracking algorithm under the V_T-constrained model by using a new "traffic shedding" technique, which converts a general V_T-constrained adversary to a (W, ϵ)-constrained adversary and then optimizes the regret bounds by carefully choosing the amount of traffic to shed. Such a proof technique may be used to adapt any queue length regret bounds derived under the (W, ϵ)-constrained model to that under the V_T-constrained model.

Finally, to the best of our knowledge, this is the first paper that characterizes the throughput region under arbitrary (and possibly adversarial) network dynamics, which provides a necessary and sufficient condition on network dynamics such that the network is stable. The characterization of the throughput region is based on our analysis under the new V_T-constrained model and the equivalence between sublinear queue length regret and rate stability.

1.4 Organization of This Paper

We first introduce the system model and relevant performance metrics in Section 2. We study the (W, ϵ)-constrained and V_T-constrained adversary models in Sections 3 and 4, respectively. In Section 5, we characterize the stability region under adversarial network models. Finally, simulation results and conclusions are given in Section 6 and 7, respectively.

2 SYSTEM MODEL

2.1 Asymptotic Notations

Let f and g be two functions defined on some subset of real numbers. Then $f(x) = O(g(x))$ if $\limsup_{x\to\infty} \frac{|f(x)|}{g(x)} < \infty$. Similarly, $f(x) = \Omega(g(x))$ if $\liminf_{x\to\infty} \frac{f(x)}{g(x)} > 0$. Also, $f(x) = \Theta(g(x))$ if $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$. In addition, $f(x) = o(g(x))$ if $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$, and in this case we say that $f(x)$ is sublinear in $g(x)$.

2.2 Network Model

Consider a network with N queues (the set of all queues are denoted by $\mathcal{N} = \{1, \cdots, N\}$). Time is slotted with a finite horizon $\mathcal{T} = \{0, \cdots, T-1\}$. Let ω_t denote the *network event* that occurs in slot t, which indicates the current network parameters, such as a vector of conditions for each link, a vector of exogenous arrivals to each node, or other relevant information about the current network links and exogenous arrivals.

At the beginning of each time slot t, the network operator observes the current network event ω_t and chooses a control action α_t from some action space $\mathcal{D}_{\omega_t}$ that can depend on ω_t. The network event ω_t and the control action α_t together produce the service vector $\mathbf{b}(\alpha_t, \omega_t) \triangleq \mathbf{b}(t) = (b_1(t), \cdots, b_N(t))$ and the arrival vector $\mathbf{a}(\alpha_t, \omega_t) \triangleq \mathbf{a}(t) = (a_1(t), \cdots, a_N(t))$. Note that $a_i(t)$ includes both the exogenous arrivals from outside the network to queue i, and the endogenous arrivals from other queues (i.e., routed packets from other queues to queue i). Thus, the above network model accounts for both single-hop and multi-hop networks, and the control action α_t may correspond to, for example, joint routing, rate allocation and scheduling decisions in a multi-hop network. Let $\mathbf{Q}(t) = (Q_1(t), \cdots, Q_N(t))$ be the queue length vector at the beginning of slot t (before the arrivals in that slot). The queuing dynamics are

$$Q_i(t+1) = [Q_i(t) + a_i(t) - b_i(t)]^+, \ \forall i \in \mathcal{N}, t \in \mathcal{T},$$

where $[x]^+ = \max\{x, 0\}$.

We assume that the sequence of network events $\{\omega_t\}_{t=0}^{T-1}$ are generate according to an *arbitrary* process (possibly non-stationary or even adversarial), except for the following boundedness assumptions.

• Under any network event and any control action, the arrivals and the service rates in each slot are bounded by constants that are independent of the time horizon T:

$$0 \leq a_i(t) \leq A, \ 0 \leq b_i(t) \leq B, \ \forall t \in \mathcal{T}, i \in \mathcal{N}.$$

For simplicity, we assume $B \geq A$ such that both arrivals and services are upper bounded by B in each slot.

A policy π generates a sequence of control actions $\left(\alpha_0^\pi, \cdots, \alpha_{T-1}^\pi\right)$ within the time horizon. In each slot t, the queue length vector, the controlled arrival vector and the service rate vector under policy π is denoted by $\mathbf{Q}^\pi(t)$, $\mathbf{a}^\pi(t)$ and $\mathbf{b}^\pi(t)$, respectively. A *causal* policy is one that generates the current control action α_t only based on the knowledge up until the current slot t. In contrast, a *non-causal* policy may generate the current control action α_t based on knowledge of the future.

Example: An example of the above network model is the power control problem in wireless downlink systems with N links. In each slot t, the controller observes the current network events $\omega_t = \big(\mathbf{a}(t), \mathbf{s}(t)\big)$, where $\mathbf{a}(t)$ and $\mathbf{s}(t)$ correspond to the vector of exogenous arrivals and the vector of channel capacities in slot t, respectively. Then the controller takes a control action α_t as a power allocation vector $\alpha_t = (\alpha_t^{(1)}, \cdots, \alpha_t^{(N)})$, subject to an instantaneous power constraint $\alpha_t \in \mathcal{D}_{\omega_t}$, where $\alpha_t^{(i)}$ is the power allocated to link i in slot t. The constraint set $\mathcal{D}_{\omega_t}$ could be, for example, the set of power allocation vectors that satisfy the peak power constraint $\sum_i \alpha_t^{(i)} \leq \alpha_{peak}$. The service rate for each link i is determined by the rate-power function $b_i(\alpha_t, \omega_t)$. For example, one possible form of the rate-power function is $b_i(\alpha_t, \omega_t) = s_i(t) \log\Big(1 + SINR_i(\alpha_t, \omega_t)\Big)$, where $SINR_i(\alpha_t, \omega_t)$ is the signal-to-interference-plus-noise ratio over link i when power vector α_t is allocated under network event ω_t.

2.3 Performance Metrics

Our objective is to find a causal control policy that keeps the total queue length as small as possible. Note that a network with adversarial dynamics may not have any steady state or well-defined time averages. Hence, it is crucial to understand the transient behavior of the network, and the traditional equilibrium-based performance metrics may not be appropriate in an adversarial setting. As a result, we introduce the notion of *queue length regret* to measure the finite-time performance achieved by a causal control policy.

Definition 2.1 (Queue Length Regret). Given the time horizon T, the queue length regret achieved by a causal policy π under a sequence of network events $\omega_0, \cdots, \omega_{T-1}$ is defined to be

$$\mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = \sum_{i \in \mathcal{N}} Q_i^\pi(T) - \sum_{i \in \mathcal{N}} Q_i^*(T), \tag{1}$$

where $\sum_i Q_i^*(T)$ is the minimum total queue length generated by the optimal non-causal policy that knows the the entire sequence of network events $\{\omega_0, \cdots, \omega_{T-1}\}$ in advance. The worst-case queue length regret achieved by policy π is

$$\mathcal{R}_T^\pi = \sup_{\omega_0, \cdots, \omega_{T-1}} \mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big).$$

In this setup, a policy π is chosen and then the adversary selects the sequence of network events $\{\omega_0, \cdots, \omega_{T-1}\}$ that maximize the regret. Intuitively, the notion of queue length regret captures the worst-case queue length difference between a causal policy and an ideal T-slot lookahead non-causal policy. This metric also measures the "price of causality", i.e., the cost of not knowing the future.

A desirable first order characteristic of a "good" policy π is that it achieves sublinear regret $\mathcal{R}_T^\pi = o(T)$ such that $\mathcal{R}_T^\pi / T \to 0$ as the time horizon $T \to \infty$. In other words, the time-average queue growth rate asymptotically approaches the one achieved by the optimal non-causal policy. We will also demonstrate the equivalence between sublinear queue length regret and *rate stability* in Section 5.

In addition to being sublinear, the queue length regret should also have a low growth rate. A lower growth rate of regret implies that the policy has a better learning ability and can adapt to the adversarial environment faster. We define the *minimax queue length regret* as the minimal queue length regret that can be achieved over the space of causal policies. A policy is said to be *asymptotically regret-optimal* if it achieves the minimax regret up to a constant multiplicative factor; this implies that, in terms of growth rate of regret, the performance of the policy is the best possible. Finally, note that the growth rate of regret is strongly related to the notion of convergence time (see [13] for more details).

Unfortunately, the following theorem shows that in general no causal policy can achieve sublinear queue length regret for any sequence of network events.

THEOREM 2.2. *For any causal policy π, there exists a sequence of network events $\omega_0, \cdots, \omega_{T-1}$ such that the queue length regret $\mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \geq T/2$.*

PROOF. We prove this theorem by constructing a sequence of network events $\omega_0, \cdots, \omega_{T-1}$ such that the lower bound is attained. Consider the power control example mentioned in

Section 2.2 with $N = 2$ links. The constraint on power allocation is $\alpha_t^{(1)} + \alpha_t^{(2)} \leq 1$ for each $t \in \mathcal{T}$, and the rate-power function is $b_i(t) = \alpha_t^{(i)} s_i(t)$. Without loss generality, assume that the time horizon T is an even number. The exogenous arrivals and channel capacities in the first $T/2$ slots is

$$a_1(t) = a_2(t) = 2, \; s_1(t) = s_2(t) = 2, \; \forall t = 0, \cdots, T/2 - 1.$$

Under the power allocation constraint, the total number of packets that can be cleared in the first $T/2$ slots is at most T. For any causal policy π, let n_1 and n_2 be the number of packets cleared over link 1 and 2 during the first $T/2$ slots, respectively. Then it is clear that $n_1 + n_2 \leq T$, which implies that $\min\{n_1, n_2\} \leq T/2$. Define $i^* = \arg\min_{i=1,2} n_i$ (ties are broken arbitrarily). Then the queue length over link i^* after $T/2$ slots is

$$Q_{i^*}^{\pi}(T/2) = T - n_{i^*} = T - \min\{n_1, n_2\} \geq T/2.$$

In the next $T/2$ slots, the adversary can set

$$a_{i^*}(t) = 0, \; s_{i^*}(t) = 0, \; t = T/2, \cdots, T-1.$$

For the other link (its index is denoted by i'), the adversary can set

$$a_{i'}(t) = 0, \; s_{i'}(t) = 2, \; t = T/2, \cdots, T-1.$$

Since there is no capacity to clear any packet over link i^* in the last $T/2$ slots, we have

$$Q_{i^*}^{\pi}(T) = Q_{i^*}^{\pi}(T/2) \geq T/2,$$

which implies that $\sum_i Q_i^{\pi}(T) \geq T/2$.

On the other hand, the optimal non-causal policy can choose the following sequence of power allocation vectors

$$\left(\alpha_t^{(i^*)}, \alpha_t^{(i')}\right) = \begin{cases} (1, 0), \; t = 0, \cdots, T/2 - 1, \\ (0, 1), \; t = T/2, \cdots, T-1, \end{cases}$$

such that $\sum_i Q_i^*(T) = 0$, which implies that the queue length regret achieved by policy π is at least $T/2$. This completes the proof.

Remark: Note that the above construction requires the value of T. We can eliminate the dependence on the time horizon T by using the standard *Doubling Trick* (see Section 2.3.1 in [15]). The details about the doubling trick are given in Appendix A.1. □

Theorem 2.2 shows that sublinear queue length regret is not achievable if the adversary has unconstrained power in determining the network dynamics. As a result, in the following two sections, we develop two adversary models where the sequence of network events (i.e. network dynamics) that the adversary can select is constrained to some "admissible" set. In Section 3, we consider the (W, ϵ)-constrained adversary model that is an extension of the widely-known yet very stringent model used in Adversarial Queuing Theory. Next in Section 4, we develop a more relaxed adversary model called the V_T-constrained adversary. Lower bounds on queue length regret and the performance of some commonly-used algorithms are analyzed under the two adversary models.

3 (W, ϵ)-CONSTRAINED ADVERSARY MODEL

In this section, we investigate the (W, ϵ)-constrained adversary model which is an extension of the classical Adversarial Queuing Theory (AQT) [5] (similar to the models used in[2–4, 9, 12]). It has stringent constraints on the set of admissible network dynamics that the adversary can set, yet is analytically tractable, which facilitates our subsequent investigation of a more relaxed adversary model in Section 4. We first give the definition of (W, ϵ)-constrained network dynamics.

Definition 3.1 ((W, ϵ)-Constrained Dynamics). Given a window size $W \in [1, T]$ and a load factor $\epsilon \in [0, 1]$, a sequence of network events $\omega_0, \cdots, \omega_{T-1}$ is (W, ϵ)-constrained if there exists a (possibly non-causal) policy π such that for any $t = 0,\ W,\ 2W, \cdots$

$$\sum_{\tau=t}^{t+W-1} a_i^\pi(\tau) \le (1-\epsilon) \sum_{\tau=t}^{t+W-1} b_i^\pi(\tau),\ \forall i \in \mathcal{N}. \tag{2}$$

Any network satisfying the above is called a (W, ϵ)**-constrained network**. In other words, the time horizon is divided into frames of size W slots, and it is required that there exists a (possibly non-causal) policy such that during every frame the total amount of arrivals to each queue is less than or equal to $1-\epsilon$ times of the total amount of services offered to that queue.

Denote by $\mathcal{A}_T(W, \epsilon)$ the set of all sequences of network events $\{\omega_0, \cdots, \omega_{T-1}\}$ that are (W, ϵ)-constrained. Then the (W, ϵ)**-constrained adversary** can only select the sequence of network events from the constrained set $\mathcal{A}_T(W, \epsilon)$. In this context, the worst-case queue length regret achieved by a causal policy π is defined to be

$$\mathcal{R}_T^\pi = \sup_{\{\omega_0, \cdots, \omega_{T-1}\} \in \mathcal{A}_T(W, \epsilon)} \mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big),$$

where $\mathcal{R}_T^\pi(\cdot)$ is given in (1).

In the following, we first provide a lower bound on queue length regret under the (W, ϵ)-constrained adversary model (Section 3.1), and then analyze the worst-case queue length regret achieved by several common control policies (Section 3.2). Note that throughout this section we mainly focus on the dependence of queue length regret on W, ϵ and T while **treating the number of users N a constant**.

3.1 Lower Bound on Queue Length Regret

The following theorem provides a lower bound on queue length regret under the (W, ϵ)-constrained adversary model.

THEOREM 3.2. *For any causal policy π, there exists a sequence of network events $\{\omega_0, \cdots, \omega_{T-1}\} \in \mathcal{A}_T(W, \epsilon)$ such that $\mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \ge \max\Big\{(1-2\epsilon)W/2,\ 0\Big\}$.*

PROOF. For any given causal policy, we construct a sequence of network events such that the lower bound is attained. The construction is similar to the one used in the proof of Theorem 2.2. The difference is that the constructed sequence of network events are also (W, ϵ)-constrained here. See Appendix A.2 for the detailed proof. □

Remarks: Note that if ϵ is some small ($\epsilon < 1/2$) constant independent of the window size W, then the above lower bound is of order $\Omega(W)$. If the window size W is comparable with the time horizon T, i.e., $W = \Theta(T)$, no causal policy can achieve sublinear (worst-case) queue length regret under the (W, ϵ)-constrained adversary model. On the other hand, if

$W = o(T)$, there *might* exist some causal policy that attains sublinear queue length regret, which we investigate in the next section. In particular, we show that the above regret lower bound can be asymptotically attained by some causal policy and thus the minimax queue length regret in (W, ϵ)-constrained networks is $\Theta(W)$.

3.2 Algorithm Performance in (W, ϵ)-Constrained Networks

In this section, we analyze the worst-case queue length regret achieved by two network control algorithms under the (W, ϵ)-constrained adversary model. The first is the famous MaxWeight policy [16] that was proved to be throughput-optimal in stochastic networks. The second is a generalized version of the Tracking Algorithm [3, 4] that was originally proposed in Adversarial Queuing Theory.

3.2.1 **MaxWeight**. In each slot t, the MaxWeight algorithm simply observes the current network event ω_t and chooses the control action as follows:

$$\alpha_t^{MW} = \arg \max_{\alpha_t \in \mathcal{D}_{\omega_t}} \sum_i Q_i(t)\Big(b_i(\alpha_t, \omega_t) - a_i(\alpha_t, \omega_t)\Big). \tag{3}$$

The solution to (3) depends on the particular network model. For example, in a single-hop wireless network with primary interference, the solution to (3) just corresponds to the one that serves the queue with the largest product of queue length and service rate; in a input-queued switch with crossbar constraints, solving (3) is equivalent to solving the Maximum Weight Matching problem [10].

The following theorem gives the performance of the MaxWeight policy in (W, ϵ)-constrained networks.

THEOREM 3.3. *Under the (W, ϵ)-constrained adversary model, the worst-case queue length regret achieved by the MaxWeight algorithm is $O(\sqrt{TW})$ for any $\epsilon \geq 0$.*

Moreover, in the special case where $\epsilon > 0$ and $a_{\min} > 0$, a better queue length regret bound of $O\Big(\frac{W}{\epsilon^3 a_{\min}}\Big)$ can be achieved by the MaxWeight algorithm, where $a_{\min}$ is the minimum arrival to each queue in each slot.

PROOF. The proof is based on the Lyapunov drift analysis. However, instead of considering the one-slot drift as in the traditional stochastic analysis, we find upper bounds on the W-slot drift and make sample-path arguments. See Appendix A.3 for details. □

There are several important observations about Theorem 3.3. First, sublinear worst-case queue length regret could be achieved by the MaxWeight policy under the (W, ϵ)-constrained adversary model as long as $W = o(T)$. Noticing that sublinear regret cannot be achieved by any causal policy if $W = \Omega(T)$ (Theorem 3.2), we have the following corollary.

COROLLARY 3.4. *Under the (W, ϵ)-constrained adversary model, sublinear worst-case queue length regret is achievable if and only if $W = o(T)$.*

Second, the $O(\sqrt{TW})$ queue length regret bound could be much larger than the lower bound in Theorem 3.2 when W is significantly smaller than T.

Third, if $a_{\min} > 0$ and the system is in the sub-critical regime ($\epsilon > 0$), then the performance bound of the MaxWeight policy is $O\Big(\frac{W}{\epsilon^3 a_{\min}}\Big)$, which could be significantly better than the $O(\sqrt{TW})$ bound when W is much smaller than T and $\epsilon, a_{\min}$ is not too small. This is analogous to the performance of the MaxWeight policy in stochastic networks: *strong* *stability*[1] can be achieved if the system is strictly inside the stability region (sub-critically loaded) while only *rate stability* can be achieved if the system is on the boundary of the stability region (critically-loaded) [11].

Finally, it should be noted that in order to derive a better regret bound in the sub-critical regime ($\epsilon > 0$), we require an additional assumption that $a_{\min} > 0$ and the obtained bound is inversely proportional to the value of $a_{\min}$. It is still unknown whether a better regret bound could be obtained in the sub-critical regime without such an assumption. We conjecture that the regret bound of $O(W/\epsilon^3)$ may hold for MaxWeight even without the assumption that $a_{\min} > 0$, since no evidence shows that there is any discontinuity at $a_{\min} = 0$. On other hand, if the assumption that $a_{\min} > 0$ is not satisfied, then the $O(\sqrt{TW})$ bound can be applied, which is sufficient to ensure that all of the subsequent results about MaxWeight (e.g., Corollary 3.4) hold true.

3.2.2 **Tracking Algorithm**. The original Tracking Algorithm was proposed in [3, 4] to solve a scheduling problem under an adversary model similar to the (W, ϵ)-constrained adversary. However, it only works for a very specific network model: (i) the network has to be single-hop where the arrival vector is independent of the control action, and (ii) the control action has to satisfy the primary interference constraints, i.e., only one link incident on the same node can be activated in each slot. Next, we extend the original Tracking Algorithm to accommodate the general network model considered in this paper.

Let Ω be the set of all possible network events that could happen in each slot. In order for the Tracking Algorithm to work, the cardinality of Ω has to be finite (otherwise it could be discretized into a finite set as in [3]). For example, in a single-hop network, suppose each network event ω_t corresponds to a couple $(\mathbf{a}(t), \mathbf{s}(t))$ where $\mathbf{a}(t)$ is a vector of exogenous packet arrivals in slot t and $\mathbf{s}(t)$ a vector of link states in slot t. For any link i and time t, assume that $0 \le a_i(t) \le B$ and $a_i(t)$ is an integer, and each link only has a finite number of S states. Then $|\Omega| = (SB)^N$.

We maintain a virtual queue for each physical queue i and each type of network event $\omega \in \Omega$. In particular, let $q_{i,\omega}(t)$ be the virtual queue length in slot t associated with link i and network event w, which corresponds to the "debts" the Tracking Algorithm "owned" to the optimal (non-causal) policy over link i under network event ω. Here, the "debts" correspond to the queue length difference between the Tracking Algorithm and the optimal policy, and the goal of the Tracking Algorithm is to track the queue length trajectory under the optimal policy. In addition, the "optimal" policy corresponds to any sequence of control actions that satisfies the window constraints (2). Note that the optimal sequence of actions cannot be calculated online. Instead, it is calculated at the end of every window of W slots and the debt owned during this window will be updated at the beginning of the *next* window. In each slot, the Tracking Algorithm just picks the control action that clears as much debt as possible.

The detailed algorithm description is shown in Algorithm 1. The virtual debt queues are updated at two times. First, the virtual debt queues are updated in each slot t, after we observe the network event ω_t and an action α_t is taken:

$$q_{i,\omega_t}(t+1) = \left[q_{i,\omega_t}(t) + a_i(\omega_t, \alpha_t) - b_i(\omega_t, \alpha_t)\right]^+, \tag{4}$$

[1] A queuing system is strongly stable if $\sum_i Q_i(t) \le B$ for some constant B as $t \to \infty$. A system is rate-stable if $\sum_i Q_i(t)/t \to 0$ as $t \to \infty$.

and the cleared debt in slot t by action α_t is defined to be

$$\Delta q_{i,\omega_t}(\alpha_t) = q_{i,\omega_t}(t) - q_{i,\omega_t}(t+1).$$

In step 3, the Tracking Algorithm just picks the action α_t^{TA} that maximizes the total cleared debt in slot t. Note that during the above procedure, only the virtual queues associated with network event ω_t are updated while the virtual queues associated with any other types of network events remain unchanged. Second, the virtual debt queues are also updated every W slots (at the end of each window), in order to add the debt "owned" to the optimal actions during the past window. Such a procedure is shown in steps 5-6. The optimal sequence of control actions $\{\alpha_\tau^*\}_{\tau=t-W+1}^{t}$ during the past window $[t-W+1,\ t]$ is first calculated, and then the corresponding debts are added to each virtual queue. In particular, the debts owned to the optimal actions in the past window for each virtual queue $q_{i,\omega}$ is $\sum_{\tau\in\mathcal{B}_\omega}\Big(b_i(\omega,\alpha_\tau^*) - a_i(\omega,\alpha_\tau^*)\Big)$, where $\mathcal{B}_\omega$ is the set of slots during the past window when network event ω happens.

Algorithm 1 Tracking Algorithm (TA)

1: Initialize $q_{i,\omega}(0) = 0$ for any $i \in \mathcal{N}$ and $\omega \in \Omega$.
2: **for** $t = 0, \cdots, T-1$ **do**
3: Choose the control action α_t^{TA} that clears as much total debt as possible:

$$\alpha_t^{TA} = \arg\max_{\alpha_t \in \mathcal{D}_{\omega_t}} \sum_i \Delta q_{i,\omega_t}(\alpha_t),$$

and update virtual debt queues according to (4).
4: **if** $\mod(t, W) = W-1$ **then**
5: Compute the sequence of optimal control actions $\{\alpha_\tau^*\}_{\tau=t-W+1}^{t}$ in the past window $[t-W+1,\ t]$, which is any solution that satisfies

$$\sum_{\tau=t-W+1}^{t} a_i(\omega_\tau, \alpha_\tau^*) \le (1-\epsilon) \sum_{\tau=t-W+1}^{t} b_i(\omega_\tau, \alpha_\tau^*),\ \forall i \in \mathcal{N}.$$

6: For each $i \in \mathcal{N}$ and $\omega \in \Omega$, update

$$q_{i,\omega}(t+1) = q_{i,\omega}(t) + \sum_{\tau\in\mathcal{B}_\omega}\Big(b_i(\omega,\alpha_\tau^*) - a_i(\omega,\alpha_\tau^*)\Big),$$

where $\mathcal{B}_\omega = \{\tau | t-W+1 \le \tau \le t,\ \omega_\tau = \omega\}$.
7: **end if**
8: **end for**

The following theorem gives the worst-case queue length regret achieved by the Tracking Algorithm under the (W,ϵ)-constrained adversary model.

THEOREM 3.5. *Assume that the size of the network event space Ω is finite. Then under the (W,ϵ)-constrained adversary model, the worst-case queue length regret achieved by the Tracking Algorithm is $O(W)$ for any $\epsilon \ge 0$.*

PROOF. We first present an upper bound on the virtual queue length, which is given in Lemma 3.6. This lemma shows that the queue length difference between the Tracking Algorithm and the optimal policy is at most $O(W)$.

Lemma 3.6. *For any $t \in \mathcal{T}$ and any type of network event $\omega \in \Omega$, we have*

$$\sum_i q_{i,\omega}(t) \leq NBW.$$

The proof to Lemma 3.6 is presented in Appendix A.4. Intuitively, since the Tracking Algorithm clears as much debt as possible in each slot, it can emulate the behavior of the optimal policy in the past, and the $O(W)$ gap is due to the delayed debt updated.

Now we prove Theorem 3.5. Let $\mathbf{Q}(t)$, $\mathbf{a}(t)$ and $\mathbf{b}(t)$ be the physical queue length vector, the arrival vector and the service vector in slot t under the Tracking Algorithm. Also let $\mathbf{Q}^*(t)$, $\mathbf{a}^*(t)$ and $\mathbf{b}^*(t)$ be the queue length vector, the arrival vector and the service vector in slot t under the optimal policy. Without loss of generality, let $T = RW$ for some positive integer R. For each $i \in \mathcal{N}$, let τ_i be the last time t when $Q_i(t) = 0$. Assume that τ_i is contained in frame r and let $t_i = (r+1)W$ (i.e., the beginning of frame $r+1$). Clearly we have $t_i - \tau_i \leq W$ and thus

$$Q_i(t_i) \leq Q_i(\tau_i) + \sum_{t=\tau_i}^{t_i-1} a_i(t) \leq WB. \tag{5}$$

Then it follows that

$$\begin{aligned}
Q_i(T) &= Q_i(t_i) + \sum_{t=t_i}^{T-1} \Big(a_i(t) - b_i(t)\Big) \\
&\leq WB + \sum_{t=t_i}^{T-1} \Big(a_i(t) + a_i^*(t) - a_i^*(t) - b_i(t)\Big) \\
&\leq WB + \sum_{t=t_i}^{T-1} \Big(a_i(t) + b_i^*(t) - a_i^*(t) - b_i(t)\Big) \\
&= WB + \sum_{\omega \in \mathcal{W}} \sum_{t \in \mathcal{T}_\omega} \Big[\Big(b_i^*(t) - a_i^*(t)\Big) - \Big(b_i(t) - a_i(t)\Big)\Big].
\end{aligned}$$

Here, the first inequality is due to (5) and the second inequality is due to Equation (2). The last equality regroups time slots according to the type of network event that occurred in each slot, where we define

$$\mathcal{T}_\omega = \{t | t_i \leq t \leq T-1,\ \omega_t = \omega\},\ \forall \omega \in \mathcal{W}.$$

Note that by the definition of the virtual queue and Lemma 3.6, we have for any $i \in \mathcal{N}$

$$\begin{aligned}
&\sum_{t \in \mathcal{T}_\omega} \Big[\Big(b_i^*(t) - a_i^*(t)\Big) - \Big(b_i(t) - a_i(t)\Big)\Big] \\
\leq & q_{i,\omega}(T) - q_{i,\omega}(t_i) \leq NBW.
\end{aligned}$$

Then it follows that

$$Q_i(T) \leq WB + |\Omega| NBW.$$

Note that the above inequality holds for all sequence of networks events and that $Q_i^*(T) \geq 0$. Then we can conclude that the worst-case queue length regret achieved by the Tracking Algorithm is

$$\mathcal{R}_T = \sum_i Q_i(T) - \sum_i Q_i^*(T) \leq NWB + |\Omega| N^2 BW = O(W).$$

This completes the proof to Theorem 3.5. □

There are several important observations about Theorem 3.5. First, sublinear worst-case queue length regret could be achieved by the Tracking Algorithm in (W, ϵ)-constrained networks as long as $W = o(T)$. Moreover, the queue length regret bound of the Tracking Algorithm is better than that of the MaxWeight policy, in terms of their dependence on W, ϵ and T.

Second, the Tracking Algorithm is asymptotically regret-optimal under the (W, ϵ)-constrained adversary model (if ϵ is a small constant), since the queue length regret achieved by the Tracking Algorithm has the same order as the lower bound in Theorem 3.2. This also implies that $\Theta(W)$ is also the growth rate of the minimax queue length regret in (W, ϵ)-constrained networks.

Corollary 3.7. *Suppose ϵ is a small constant independent of T and W. Then under the (W, ϵ)-constrained adversary model, the minimax queue length regret is $\Theta(W)$.*

Third, the Tracking Algorithm needs to maintain a virtual queue for each type of network events while the size of the network event space Ω may be exponential in the number of users N. As a result, the Tracking Algorithm may not be a practical algorithm. The purpose of presenting the Tracking Algorithm is to demonstrate that the lower bound in Theorem 3.2 could be asymptotically achieved by a causal policy. Note that Andrews and Zhang [3, 4] proposed methods to get rid of the exponential dependence on N, at the expense of much more involved algorithms. Their methods may be adapted to our scenario to achieve a better dependence on N, but it is left for future work since the focus of this paper is the scaling of queue length regret with T while N is treated as a constant.

4 V_T-CONSTRAINED ADVERSARY MODEL

The aforementioned (W, ϵ)-constrained model is relatively restrictive, where the stringent constraints (2) have to be satisfied for every window of W slots. In this section, we consider a general adversary model where the window constraints (2) are relaxed.

The new adversary model is parameterized by the inherent variation in the sequence of network events, which is measured as follows. Given a sequence of network events $\omega_0, \cdots, \omega_{T-1}$ and a (possibly non-causal) policy, we define

$$V^{\pi}\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = \max_{t \le T} \sum_i Q_i^{\pi}(t).$$

The above function measures the peak queue length achieved by policy π during its sample path. We further define

$$V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = \min_{\pi} V^{\pi}\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big),$$

i.e., the minimum peak queue length that could be achieved by any (possibly non-causal) policy under the sequence of network events $\omega_0, \cdots, \omega_{T-1}$. Note that $V(\cdot)$ only depends on $\{\omega_0, \cdots, \omega_{T-1}\}$ and measures the inherent variations in the sequence of network events.

Now we define the notion of V_T-constrained dynamics where the value of $V(\cdot)$ is constrained by some budget V_T.

Definition 4.1 (V_T-Constrained Dynamics). Given some $V_T \in [0, NTB]$, a sequence of network events $\omega_0, \cdots, \omega_{T-1}$ is V_T-constrained if

$$V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \le V_T.$$

Any network satisfying the above is called a **V_T-constrained network**. Intuitively, the V_T-constrained model requires that the total queue length generated by the "oracle" during its sample path should be upper bounded by V_T.

Denote by $\mathcal{V}_T$ the set of all possible sequences of network events that are V_T-constrained. A **V_T-constrained adversary** can only select the sequence of network events from the set $\mathcal{V}_T$. In this context, the worst-case queue length regret achieved by a causal policy π is defined as

$$\mathcal{R}_T^\pi = \sup_{\{\omega_0,\cdots,\omega_{T-1}\}\in\mathcal{V}_T} \mathcal{R}_T^\pi\Big(\{\omega_0,\cdots,\omega_{T-1}\}\Big).$$

where $\mathcal{R}_T^\pi(\cdot)$ is given in (1). Note that we restrict the range of V_T to $[0, NTB]$ since the peak queue length within T slots is at most NTB. Any larger value of V_T has the same effect as $V_T = NTB$. Note also that the larger V_T is, the more variations the network could have. By varying the value of V_T from 0 to NTB, the above V_T-constrained adversary model covers the full spectrum of network dynamics. If $V_T = 0$, then the arrivals should be less than or equal to the services for each queue *in every slot*, and network dynamics is stringently constrained. If $V_T = NTB$ (which corresponds to the maximum total queue length that could be build up during T slots), the network dynamics is completely unconstrained.

In the following, we will first provide a lower bound on queue length regret under the V_T-constrained adversary model in Section 4.1 and then analyze the regret achieved by the MaxWeight policy and the Tracking Algorithm in Section 4.2. Some important discussions are provided in Section 4.3.

4.1 Lower Bound on Queue Length Regret

The following theorem provides a lower bound on the queue length regret under the V_T-constrained adversary model.

THEOREM 4.2. *For any causal policy π, there exists a sequence of network events $\{\omega_0,\cdots,\omega_{T-1}\}\in\mathcal{V}_T$ such that $\mathcal{R}_T^\pi\Big(\{\omega_0,\cdots,\omega_{T-1}\}\Big) \geq cV_T$, where c is some constant independent of T and V_T.*

PROOF. For any given causal policy, we construct a sequence of network events such that the lower bound is attained. The construction is similar to the one used in the proof of Theorem 2.2. The difference is that the constructed sequence of network events are V_T-constrained here. See Appendix A.5 for the detailed proof. □

If $V_T = \Omega(T)$, then no causal policy can achieve sublinear queue length regret under the V_T-constrained adversary model. On the other hand, if $V_T = o(T)$, there *might* exist some causal policy that attains sublinear queue length regret, which we investigate in Section 4.2.

4.2 Algorithm Performance in V_T-Constrained Networks

In this section, we analyze the queue length regret achieved by two algorithms in V_T-constrained networks: the MaxWeight policy and the Tracking Algorithm. In particular, we show that both algorithms achieves sublinear regret if $V_T = o(T)$.

4.2.1 **MaxWeight**. The MaxWeight policy discussed in Section 3.2 can be directly applied in V_T-constrained networks. The following theorem gives the worst-case queue length regret achieved by the MaxWeight policy under the V_T-constrained adversary model.

THEOREM 4.3. *Under the V_T-constrained adversary model, the worst-case queue length regret achieved by the MaxWeight policy is* $O\left(T^{2/3}V_T^{1/3}\right)$.

PROOF. We consider a new system that is obtained by shedding a certain amount of traffic from the original system such that the new system is $(W, 0)$-constrained for some windows size W that is to be selected later. By the definition of V_T, there exists some (possibly non-causal) policy π^* such that

$$\max_{t\le T}\sum_i Q_i^{\pi^*}(t) = V_T. \tag{6}$$

Denote by $\tilde{\mathbf{a}}^{\pi^*}(t)$ and $\mathbf{a}^{\pi^*}(t)$ the arrival vector in the new system and in the original system in slot t under π^*, respectively. Also let X_T be the total amount of shed traffic within the time horizon, i.e.,

$$X_T = \sum_{t=0}^{T-1}\sum_i a_i^{\pi^*}(t) - \sum_{t=0}^{T-1}\sum_i \tilde{a}^{\pi^*}(t).$$

Now we divide the time horizon into frames of size W slots. Without loss of generality, assume that W divides T. Then the total number of frames is T/W. Denote by $t_r = (r-1)W$ the beginning of frame r. In order to make the new system $(W, 0)$-constrained, we can shed traffic in each frame r such that $\sum_i \widetilde{Q}_i^{\pi^*}(t_{r+1}) = 0$, where $\widetilde{\mathbf{Q}}^{\pi^*}(t)$ is the queue length vector in slot t in the new system under policy π^*. Note that $\sum_i Q_i^{\pi^*}(t_{r+1}) \le V_T$ by equation (6), and thus at most V_T arrivals need to be shed in frame r to ensure $\sum_i \widetilde{Q}_i^{\pi^*}(t_{r+1}) = 0$. Therefore, in order to make the new system $(W, 0)$-constrained, at most $V_T T/W$ arrivals need to be shed during the entire time horizon, i.e.,

$$X_T \le V_T T/W. \tag{7}$$

Let $\mathbf{Q}(t)$ and $\widetilde{\mathbf{Q}}(t)$ be queue length vector in slot t if the MaxWeight algorithm is applied to the original system and the new system, respectively. Then it is clear that

$$\sum_i Q_i(T) \le X_T + \sum_i \widetilde{Q}_i(T). \tag{8}$$

Since the new system is $(W, 0)$-constrained, by the proof of Theorem 3.3, we have

$$\sum_i \widetilde{Q}_i(T) \le c_1\sqrt{WT} \tag{9}$$

for some constant $c_1 > 0$. Combing (7), (8) and (9) we have

$$\sum_i Q_i(T) \le V_T T/W + c_1\sqrt{WT}.$$

Choosing $W = c_2 V_T^{2/3} T^{1/3} \le T$ for some constant $c_2 > 0$, we have

$$\sum_i Q_i(T) \le \left(\frac{1}{c_2} + c_1\sqrt{c_2}\right)T^{2/3}V_T^{1/3} = O(T^{2/3}V_T^{1/3}),$$

which implies that the worst-case queue length regret achieved by the MaxWeight policy under the V_T-constrained adversary model is $O(T^{2/3}V_T^{1/3})$. □

There are several observations about Theorem 4.3. First, the MaxWeight policy achieves sublinear queue length regret under the V_T-constrained adversary model whenever $V_T = o(T)$. Notice that sublinear regret cannot be achieved by any causal policy if $V_T = \Omega(T)$ (Theorem 4.2). We have the following corollary.

COROLLARY 4.4. *Under the V_T-constrained adversary model, sublinear worst-case queue length regret is achievable if and only if $V_T = o(T)$.*

Second, the MaxWeight policy does not attain the $\Omega(V_T)$ lower bound in Theorem 4.2, especially when V_T is significantly smaller than T.

4.2.2 **Tracking Algorithm**. The Tracking Algorithm introduced under the (W, ϵ)-constrained adversary model requires that the window constraints (2) be satisfied for some window size W. However, there might be no window structure under the V_T-constrained adversary model and thus the Tracking Algorithm cannot be directly applied in V_T-constrained networks. We slightly modify the Tracking Algorithm of Section 3.2 by setting $W = \left\lceil \sqrt{\frac{TV_T}{NB}} \right\rceil$. Note that $V_T \in [0, NTB]$, which guarantees that $W \leq T$. Moreover, step 6 of the original Tracking Algorithm is tweaked to find a sequence of control actions $\{\alpha_\tau^*\}_{\tau=t-W+1}^{t}$ that satisfies the following constraints:

$$\sum_{\tau=t-W+1}^{t} a_i(\omega_\tau, \alpha_\tau^*) \leq \sum_{\tau=t-W+1}^{t} b_i(\omega_\tau, \alpha_\tau^*) + V_T, \;\forall i \in \mathcal{N}. \tag{10}$$

Note that by the definition of V_T-constrained networks, there always exists a feasible solution satisfying the above constraints.

Under the above setting, the worst-case queue length regret achieved by the Tracking Algorithm under the V_T-constrained adversary model is given in the following theorem[2].

THEOREM 4.5. *Under the V_T-constrained adversary model, the worst-case queue length regret achieved by the Tracking Algorithm is $O(\sqrt{V_T T})$.*

PROOF. It can be easily verified that Lemma 3.6 still holds in V_T-constrained networks. Then following the similar line of argument as in the proof to Theorem 3.5, we have

$$\begin{aligned} Q_i(T) &\leq WB + \sum_{t=t_i}^{T-1} \Big(a_i(t) + a_i^*(t) - a_i^*(t) - b_i(t)\Big) \\ &\leq WB + \sum_{t=t_i}^{T-1} \Big(a_i(t) + b_i^*(t) - a_i^*(t) - b_i(t)\Big) + \frac{V_T T}{W} \\ &= WB + |\Omega| NBW + \frac{V_T T}{W}, \end{aligned}$$

where the second inequality is due to (10). Choosing $W = \left\lceil \sqrt{\frac{TV_T}{NB}} \right\rceil$, we have

$$Q_i(T) \leq (1 + B)\sqrt{\frac{TV_T}{NB}} + (|\Omega| + 1)\sqrt{TV_T NB},$$

which implies that the worst-case queue length regret achieved by the Tracking Algorithm under the V_T-constrained adversary model is $O(\sqrt{TV_T})$. □

There are several important observations about Theorem 4.5. First, similar to MaxWeight, the Tracking Algorithm also guarantees sublinear queue length regret whenever $V_T = o(T)$. Second, the Tracking Algorithm has a better regret bound than that under the MaxWeight policy when V_T is significantly smaller than T. Third, the regret bound of the Tracking

[2] As is discussed in Section 3.2.2, the set of possible network events should be finite in order for the Tracking Algorithm to work.

Algorithm does not attain the $\Omega(V_T)$ regret lower bound in Theorem 4.2. Thus, finding a causal policy that can close the gap remains an open problem.

4.3 Discussions

4.3.1 Sensitivity of Tracking Algorithm to Parameters. Note that the above Tracking Algorithm requires V_T as a parameter. Unfortunately, in practice, it is impossible to know the precise value of V_T for a given network in advance. To alleviate this issue, we can search for the correct value of V_T. Note that the range for V_T is $[0, NBT]$. Then one may perform binary search to find the correct value of V_T by running the Tracking algorithm with different values of V_T over multiple episodes within the time horizon (e.g., if the time horizon is $T = 10^5$ slots, then one episode could be 10^3 slots). Similar techniques can be applied if the Tracking Algorithm is used in (W, ϵ)-constrained networks where the values W and ϵ are required as input parameters.

4.3.2 Relationship Between Adversary Models. The V_T-constrained adversary model generalizes the (W, ϵ)-constrained adversary model: any sequence of network events that are (W, ϵ)-constrained must also be V_T-constrained with $V_T = O(W)$ due to the window structure. The analysis in the V_T-constrained adversary model also gives a more general condition for sublinear queue length regret.

4.3.3 Queue Length Regret vs. Queue Length. Note that under the (W, ϵ)-constrained model, the optimal queue length is bounded by $0 \le \sum_i Q_i^*(T) \le O(W)$; under the V_T-constrained model, the optimal queue length is bounded by $0 \le \sum_i Q_i^*(T) \le O(V_T)$. As a result, the queue length regret bounds we derived under the two constrained model are also legitimate bounds for total queue length. While queue length regret and queue length may be quantitatively similar under the two constrained models, it does not imply that queue length regret is equivalent to queue length in general. For example, if we replace queue length regret by queue length, the impossibility result in Theorem 2.2 would become meaningless since the adversary can always trivially overload the system to obtain $\Omega(T)$ queue length. By comparison, under the notion of queue length regret, Theorem 2.2 establishes that the gap between a causal policy and the optimal policy could be very large, and thus it impossible for any causal policy to stabilize the network even if the system is not overloaded. Such an impossibility result is the primary motivation for imposing constraints on the best achievable queue length performance as in Sections 3 and 4.

5 STABILITY REGION IN ADVERSARIAL NETWORKS

In this section, we characterize the stability region under adversarial network models. We first give the definition of rate stability.

Definition 5.1 (Rate Stability). A network is rate-stable under a control policy π if

$$\lim_{T \to \infty} \frac{\sum_i Q_i^\pi(T)}{T} = 0. \tag{11}$$

Intuitively, rate stability requires that the long-term arrival rate is less than or equal to the long-term service rate.

The following observation directly follows from the definition of queue length regret.

Observation 1. *In any V_T-constrained network with $V_T = o(T)$, rate stability is equivalent to sublinear queue length regret.*

Proof. See Appendix A.6. ☐

Combing the above observation with Theorems 4.3 and 4.5, we have the following corollary.

COROLLARY 5.2. *In any V_T-constrained network with $V_T = o(T)$, both the MaxWeight policy and the Tracking Algorithm achieve rate stability .*

The notion of stability region describes a necessary and sufficient condition such that rate stability could be achieved. In particular, in a single-hop stochastic network, it was shown in [11] that the stability region can be described by the existence of a causal policy π such that

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} a_i(t) \le \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} b_i^{\pi}(t),\ \forall i.$$

However, such conditions cannot be applied in adversarial settings since the above limits may not exist. Thus, we need a new characterization of the stability region for networks with arbitrary (possibly adversarial) dynamics. as is given in the following theorem.

THEOREM 5.3. *The network is rate-stable under some causal policy if and only if $V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = o(T)$ as $T \to \infty$, for a given sequence of network events $\omega_0, \omega_1, \cdots$.*

PROOF. See Appendix A.7. □

In other words, the stability region can be described as the set of sequences of network events $\{\omega_0, \omega_1, \cdots\}$ such that $v\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = o(T)$. We say that a control policy is *throughput-optimal* if it achieves rate stability whenever the sequence of network events $\omega_0, \omega_1, \cdots$ is inside the stability region. Combining Theorem 5.3 and Corollary 5.2, we conclude the following.

COROLLARY 5.4. *Both the MaxWeight policy and the Tracking Algorithm are throughput-optimal in networks with arbitrary (possibly adversarial) dynamics.*

6 SIMULATIONS

In this section, we empirically validate the theoretical bounds derived in this paper and compare the performance of the MaxWeight policy and the Tracking Algorithm in different scenarios.

In our simulations, we consider a one-hop wireless network with a single base station and N users. In each slot, the base station observes the current channel rate for each user and selects one of the users to serve. The packet arrival process and the evolution of channel rates are arbitrary and possibly adversarial. In the following, we investigate two specific scenarios.

6.1 Scenario I: Network with Adaptive Adversary

First, we consider a scenario where the channel rate vector in each slot is controlled by an adaptive adversary. There are $N = 2$ users in the network, and time is divided into frames of W slots. In the first $\lceil W/2 \rceil$ slots of each frame, the arrivals to each user are 2 packets/slot and the channel rate for each user is also 2 packets/slot. In the remaining slots of each frame, there are no arrivals to both users while the channel rate is zero for the user with a longer queue and 2 packets/slot for the other user. If the two users have the same queue length, ties are broken randomly. Such a scenario is similar to the one that we use to prove the regret lower bound under the (W, ϵ)-constrained adversary model (see the proof of Theorems 3.2), and it has been shown that **this is a $(W, 0)$-constrained adversary**.

Figure 1 shows the queue length trajectory of the MaxWeight policy and the Tracking Algorithm under the above adaptive $(W, 0)$-constrained adversary. It is observed that there is no steady state and the total queue length oscillates indefinitely with a period of $W = 50$ slots. In Figure 2, we plot the queue length regrets achieved by the MaxWeight policy and the Tracking Algorithm. The regret lower bound, as is shown in Theorem 3.2, is also plotted in the figures for comparison. The regret upper bounds (see Theorems 3.3 and 3.5), however, are much larger than the actual regrets achieved by the two algorithms, thus being omitted in most of the figures. Specifically, Figures 2(a)-2(c) illustrate the growth of queue length regret w.r.t. the time horizon T under different values of W. When $W = O(1)$, i.e., W does not grow with the time horizon T, the queue length regrets achieved by both algorithms remain at some constants when T is sufficiently large. When $W = O(\sqrt{T})$, the regrets achieved by the two algorithms increase with the time horizon T, yet the growth rate is sublinear. When $W = O(T)$, both algorithms have linearly-increasing queue length regret w.r.t. T. In fact, even the regret lower bound becomes linear in T, implying that no algorithms can have sublinear queue length regret in this case. In addition, the performances of the MaxWeight policy and the Tracking Algorithm are similar under this adaptive $(W, 0)$-constrained adversary, though the Tracking Algorithm could be slightly worse than MaxWeight occasionally. Figure 2(d) shows the growth of queue length regrets w.r.t. the increase of window size W, where we fix the time horizon $T = 10^5$ slots. It is observed that the queue length regret achieved by the Tracking Algorithm grows linearly with W, just as the upper bound (Theorem 3.5) predicts. The MaxWeight policy also empirically achieves a regret that is linear in W, which shows that the analysis in Theorem 3.3 is not tight in this scenario.

In conclusion, the above simulation suggests that the network does not have any steady state, and no algorithm can achieve sublinear queue length regret if $W = O(T)$. Both the MaxWeight policy and the Tracking Algorithm achieve sublinear regret when $W = o(T)$, and their performance bounds are validated.

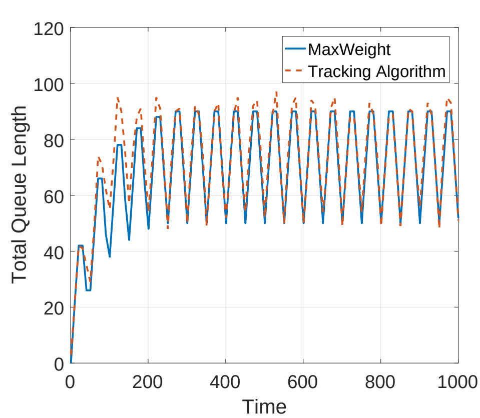
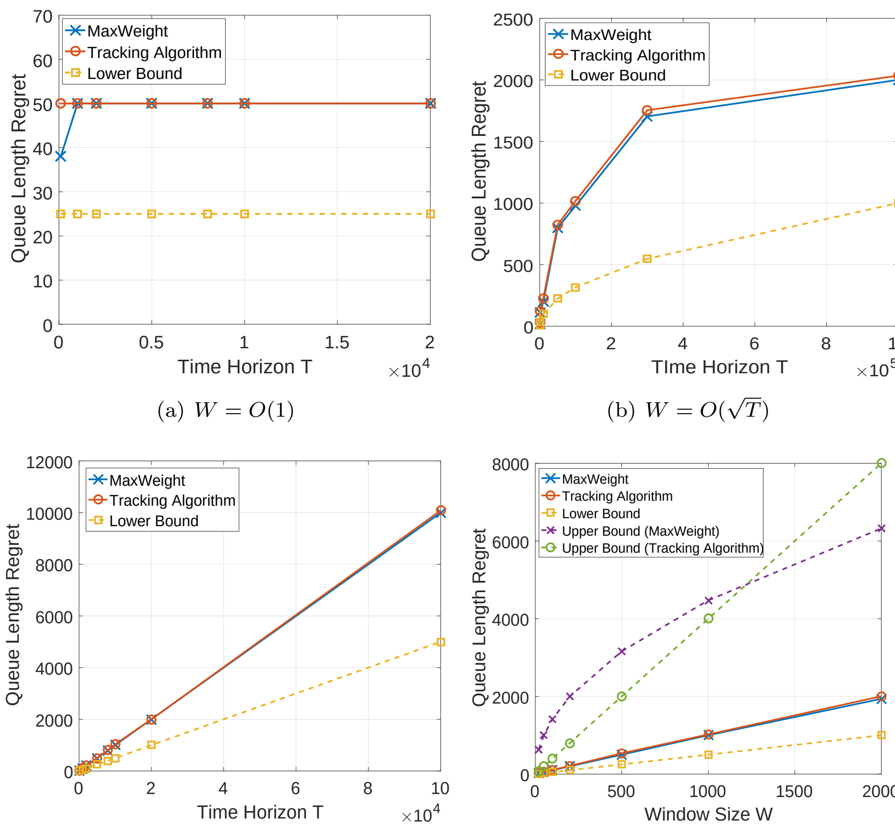


Fig. 1. Queue length trajectory of the MaxWeight policy and the Tracking Algorithm in Scenario I (adaptive $(W, 0)$-constrained adversary, $W = 50$).

6.2 Scenario II: Time-varying Network

Next, we consider a scenario where the channel rate for each user in each slot is a random variable whose mean is time-varying and periodic. In such a scenario, the network dynamics



(a) $W = O(1)$ (b) $W = O(\sqrt{T})$

(c) $W = O(T)$ (d) Regret vs. W

Fig. 2. Queue length regret of the MaxWeight policy and the Tracking Algorithm in Scenario I (adaptive $(W, 0)$-constrined adversary). (a)-(c): growth of queue length regret w.r.t. the time horizon T under different values of W; (d) growth of queue length regret w.r.t. the window size W where we fix the time horizon $T = 10^5$ slots.

is not as adversarial as in Scenario I, yet it sheds light on the regret performance of the MaxWeight policy and the Tracking Algorithm in a milder (less adversarial) setting. Similar simulation setup was also considered in [4]. There are $N = 3$ users, and the mean channel rate for each user in each slot is periodic within the set $[0, 30, 60]$ (packets/slot). Each mean rate is held for each user for X consecutive slots, and thus the period is NX. For example, if $X = 2$, the sequence of mean rate vectors is

$$(0, 30, 60) \to (0, 30, 60) \to (30, 60, 0) \to (30, 60, 0)$$
$$\to (60, 0, 30) \to (60, 0, 30) \to (0, 30, 60) \to \cdots$$

Let $r_i(t)$ and $R_i(t)$ be the mean rate and the actual rate for user i in slot t, respectively. If $r_i(t) = 0$, then $R_i(t) = 0$ with probability 1 otherwise $R_i(t)$ is a uniform random variable in the range of $[r_i(t) - Y, r_i(t) + Y]$ where Y is some constant controlling the variance of channel rates. Moreover, 20 packets arrive to each user in every slot such that the system is not trivially overloaded. Note that if $Y = 0$, the network is $(W, 0)$-constrained with $W = NX$, since the optimal schedule can always serve the user with rate 60 in every slot in order to maintain the window structure. On the other hand, if $Y > 0$, there exists no strict window structure, and the general V_T-constrained adversary model is adopted, where the value of V_T (i.e., the peak queue length during the optimal trajectory) is determined by X and Y. In the following, we fix $Y = 15$.

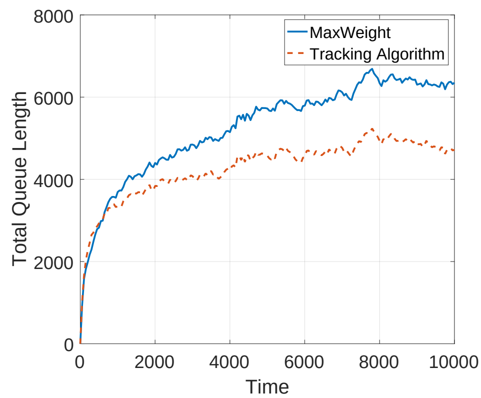
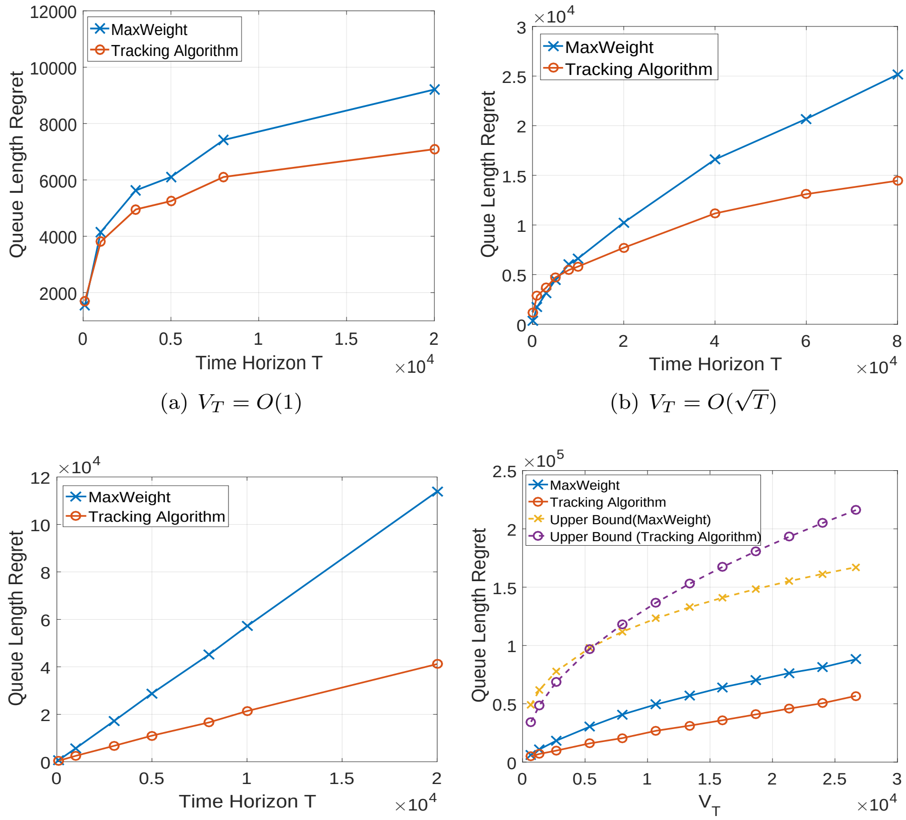


Fig. 3. Queue length trajectory of MaxWeight and Tracking Algorithm in Scenario II (V_T-constrained adversary with time-varying mean rates, $X = 50$, $Y = 15$).

Figure 3 illustrates the queue length trajectory under the MaxWeight policy and the Tracking Algorithm. Unlike in the previous scenario, the total queue lengths under both algorithm converge to some steady states. Figure 4 illustrates the regret performance of the two algorithms under different values of V_T. It is observed that if $V_T = o(T)$, the queue length regret achieved by both algorithms grows sublinearly with the time horizon T; if $V_T = O(T)$, then both algorithms suffer from linearly-increasing regret. Moreover, when T is sufficiently large, the regret performance of the Tracking Algorithm is significantly better than that of the MaxWeight policy, which implies that the Tracking Algorithm has a smaller regret growth rate or it has a better multiplicative factor. On theother hand, when T is relatively small, the MaxWeight policy outperforms the Tracking Algorithm by some additive constant factor. Finally, the analytical upper bounds we developed under the V_T-constrained adversary model (see Theorems 4.3 and 4.5) are validated in Figure 4(d).

Sensitivity of the Tracking Algorithm. The implementation of the Tracking Algorithm requires V_T as an input parameter. Unfortunately, in practice, it is impossible to know the precise value of V_T for a given network in advance. Figures 5 shows the sensitivity of the Tracking Algorithm to the value of V_T. An important observation is that using a larger value of V_T incurs a much smaller regret than using a smaller value of V_T, though it is still worse than using the correct estimation of V_T. In particular, using a smaller value of V_T leads to linear queue length regret while using a larger value of V_T achieves sublinear regret. As we discussed in Section 4.2, the regret upper bound we derived in Theorem 4.5 still holds true if the Tracking Algorithm uses a larger value of V_T and sublinear queue length regret can be achieved as long as the $V_T = o(T)$. However, the upper bound does not hold if we use a smaller value of V_T, and the Tracking Algorithm may suffer from linear regret even if the true value of V_T is $o(T)$.



(a) $V_T = O(1)$

(b) $V_T = O(\sqrt{T})$

(c) $V_T = O(T)$

(d) Regret vs. V_T

Fig. 4. Queue length regret of the MaxWeight policy and the Tracking Algorithm in Scenario II (time-varying network). (a)-(c): growth of queue length regret w.r.t. the time horizon T under different values of V_T; (d) growth of queue length regret w.r.t. the network variation budget V_T where we fix the time horizon $T = 10^4$ slots. Note that $Y = 15$ and X is varied to control the value of V_T.

7 CONCLUSIONS

In this paper, we focus on optimizing queue length regret under adversarial network models. We show that sublinear queue length regret cannot be achieved if the network dynamics are unconstrained, and investigate two constrained adversary models. We first consider the restrictive (W, ϵ)-constrained adversary model and then propose a more relaxed V_T-constrained adversary model. Lower bounds on queue length regret are derived under the two adversary models, and the regret performance of two control policies is analyzed, i.e., the MaxWeight policy and the Tracking Algorithm. It is shown that the Tracking Algorithm is nearly regret-optimal under the (W, ϵ)-constrained adversary model and that the Tracking Algorithm has a better regret bound than that of the MaxWeight policy. Finally, we establish the connection between the proposed adversarial framework and the traditional stochastic network optimization framework. In particular, we show the equivalence of sublinear queue length regret and rate stability, and characterize the stability region under adversary models.

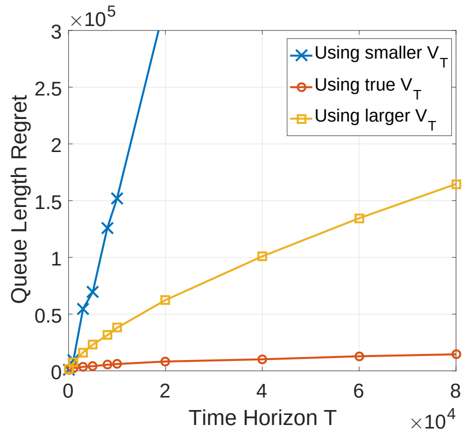


Fig. 5. Sensitivity of the Tracking Algorithm to the value of V_T. The variation in the actual sequence of network events is $O(\sqrt{T})$. The Tracking Algorithm is set to use the true value ($V_T = O(\sqrt{T})$), a smaller value ($V_T = O(1)$) and a larger value ($V_T = O(T^{3/4})$), respectively.

Both the MaxWeight policy and the Tracking Algorithm are shown to be throughput-optimal in an adversarial network.

REFERENCES

[1] Matthew Andrews, Baruch Awerbuch, Antonio Fernández, Tom Leighton, Zhiyong Liu, and Jon Kleinberg. 2001. Universal-stability results and performance bounds for greedy contention-resolution protocols. *Journal of the ACM (JACM)* 48, 1 (2001), 39–69.

[2] Matthew Andrews, Kyomin Jung, and Alexander Stolyar. 2007. Stability of the Max-weight Routing and Scheduling Protocol in Dynamic Networks and at Critical Loads. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing (STOC '07)*. ACM, 145–154.

[3] M. Andrews and L. Zhang. 2002. Scheduling over a time-varying user-dependent channel with applications to high speed wireless data. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* 293–302. https://doi.org/10.1109/SFCS.2002.1181952

[4] M. Andrews and L. Zhang. 2006. Scheduling Over Nonstationary Wireless Channels With Finite Rate Sets. *IEEE/ACM Transactions on Networking* 14, 5 (Oct 2006), 1067–1077. https://doi.org/10.1109/TNET.2006.882835

[5] Allan Borodin, Jon Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P Williamson. 2001. Adversarial queuing theory. *Journal of the ACM (JACM)* 48, 1 (2001), 13–38.

[6] Vicent Cholvi and Juan Echagüe. 2007. Stability of FIFO networks under adversarial models: State of the art. *Computer Networks* 51, 15 (2007), 4460–4474.

[7] Rene L Cruz. 1991. A calculus for network delay. I. Network elements in isolation. *IEEE Transactions on information theory* 37, 1 (1991), 114–131.

[8] Subhashini Krishnasamy, Rajat Sen, Ramesh Johari, and Sanjay Shakkottai. 2016. Regret of Queueing Bandits. In *Advances in Neural Information Processing Systems*. 1669–1677.

[9] Sungsu Lim, Kyomin Jung, and Matthew Andrews. 2014. Stability of the Max-weight Protocol in Adversarial Wireless Networks. *IEEE/ACM Trans. Netw.* 22, 6 (Dec. 2014), 1859–1872.

[10] Nick McKeown, Adisak Mekkittikul, Venkat Anantharam, and Jean Walrand. 1999. Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications* 47, 8 (1999), 1260–1267.

[11] Michael J Neely. 2010. Stability and capacity regions or discrete time queueing networks. *arXiv preprint arXiv:1003.3396* (2010).

[12] Michael J Neely. 2010. Universal scheduling for networks with arbitrary traffic, channels, and mobility. In *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 1822–1829.

[13] Michael J Neely and Hao Yu. 2017. Online Convex Optimization with Time-Varying Constraints. *arXiv preprint arXiv:1702.04783* (2017).

[14] George S Paschos and Leandros Tassiulas. 2016. Sustainability of Service Provisioning Systems under Stealth DoS Attacks. *IEEE Transactions on Control of Network Systems* (2016).

[15] Shai Shalev-Shwartz et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* 4, 2 (2012), 107–194.

[16] Leandros Tassiulas and Anthony Ephremides. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE transactions on automatic control* 37, 12 (1992), 1936–1948.

[17] Yulong Zou, Jia Zhu, Xianbin Wang, and Lajos Hanzo. 2016. A survey on wireless security: Technical challenges, recent advances, and future trends. *Proc. IEEE* 104, 9 (2016), 1727–1765.

A APPENDIX

A.1 Doubling Trick

In this appendix, we apply the standard doubling trick [15] to the example used for proving Theorem 2.2, such that it does not require any knowledge about the time horizon T in advance.

If the time horizon T is not known in advance, we can divide the time horizon into periods of exponentially increasing lengths: the first period has the length $T_1 = 2^0 = 1$, the second period has the length $T_2 = 2^1 = 2$,..., the m-th period has the length $T_m = 2^{m-1}$, etc. Then the total number of periods is at most $M = \lceil \log_2(T) \rceil$ (note that the length of the last period M may be less than 2^{M-1}). Since the length of each period is known, we can construct the same sequence of network events over each period as in the proof to Theorem 2.2. Let q_1 and q_2 be the queue length over link 1 and link 2 at the beginning of period $M-1$. Also let n_1 and n_2 be the number of packets cleared over link 1 and link 2 during the first half of period $M-1$. Following the similar line of argument as in the proof of Theorem 2.2, we have $n_1 + n_2 \leq T_{M-1} = 2^{M-2}$, which implies that $\min\{n_1, n_2\} \leq 2^{M-3}$. Define $i^* = \arg\min_{i=1,2} n_i$ (ties are broken arbitrarily). Then the queue length over link i^* after the first half of period $M-1$ is

$$q_{i^*} + 2^{M-2} - n_{i^*} \geq q_{i^*} + 2^{M-3} \geq 2^{M-3} \geq 2^{\log_2 T - 3} = T/8.$$

Since there is no capacity to clear any packets in the remaining half of period $M-1$, the total queue length at the end of period $M-1$ is

$$\sum_i Q_i^\pi(2^{M-1}) \geq Q_{i^*}^\pi(2^{M-1}) \geq T/8. \tag{12}$$

Note that during the last period M ($t = 2^{M-1}, \cdots, T$) the optimal policy always clears 2 packets per slot by our construction, which implies that

$$\sum_i Q_i^\pi(T) - \sum_i Q_i^\pi(2^{M-1}) \geq \sum_i Q_i^*(T) - \sum_i Q_i^*(2^{M-1})$$

Combing the above inequality with (12) and noticing that $\sum_i Q_i^*(2^{M-1}) = 0$, we have

$$\mathcal{R}_T^\pi = \sum_i Q_i^\pi(T) - \sum_i Q_i^*(T) \geq \sum_i Q_i^\pi(2^{M-1}) \geq T/8,$$

which completes the proof.

A.2 Proof to Theorem 3.2

We prove this theorem by constructing a sequence of network events $\{\omega_0, \cdots, \omega_{T-1}\} \in \mathcal{A}_T(W, \epsilon)$ such that the lower bound is attained. Consider the power control example mentioned in Section 2.2 with $N = 2$ links. The constraint on power allocation is $\alpha_t^{(1)} + \alpha_t^{(2)} \leq 1$ for each $t \in \mathcal{T}$, and the rate-power function is $b_i(t) = \alpha_t^{(i)} s_i(t)$. Without loss generality, we assume that $(1-\epsilon)W/2$ and $W/2$ are integers.

In slot $t = 0, \cdots, (1-\epsilon)W/2 - 1$, the exogenous arrivals and the channel capacities are

$$a_1(t) = a_2(t) = 2,\ s_1(t) = s_2(t) = 2.$$

In slot $t = (1-\epsilon)W/2, \cdots, W/2 - 1$, the arrivals and the channel capacities are

$$a_1(t) = a_2(t) = 0,\ s_1(t) = s_2(t) = 2.$$

Under the peak power constraints, the total number of packets that can be cleared in the first $W/2$ slots is at most W. For any causal policy π, let n_1 and n_2 be the number of packets cleared over link 1 and 2 during the first $W/2$ slots, respectively. Then it is clear that $n_1 + n_2 \leq W$, which implies that $\min\{n_1, n_2\} \leq W/2$. Define $i^* = \arg\min_{i=1,2} n_i$ (ties are broken arbitrarily). Then the queue length over link i^* after $W/2$ slots is

$$\begin{aligned} Q_{i^*}^{\pi}(W/2) &= (1-\epsilon)W - \min\{n_1, n_2\} \\ &\geq (1-\epsilon)W - W/2 \\ &= (1-2\epsilon)W/2. \end{aligned}$$

In the remaining slots, the adversary can set

$$a_{i^*}(t) = 0,\ s_{i^*}(t) = 0,\ t = W/2, \cdots, T-1.$$

For the other link (its index is denoted by i'), the adversary can set

$$a_{i'}(t) = 0,\ s_{i'}(t) = 2,\ t = W/2, \cdots, T-1.$$

Since there is no capacity to clear any packet over link i^* in the remaining $T - W/2$ slots, we have

$$Q_{i^*}^{\pi}(T) = Q_{i^*}^{\pi}(W/2) \geq (1-2\epsilon)W/2,$$

which implies that $\sum_i Q_i^{\pi}(T) \geq (1-2\epsilon)W/2$.

On the other hand, the optimal non-causal policy can choose the following sequence of power allocation vectors

$$\left(\alpha_t^{(i^*)}, \alpha_t^{(i')}\right) = \begin{cases} (1,0),\ t = 0, \cdots, W/2-1, \\ (0,1),\ t = W/2, \cdots, T-1. \end{cases}$$

such that constraints (2) are satisfied, which implies that the above sequence of network events $\{\omega_0, \cdots, \omega_{T-1}\}$ is (W, ϵ)-constrained. Moreover, we have $\sum_i Q_i^*(T) = 0$, and thus $\mathcal{R}_T^{\pi}\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \geq (1-2\epsilon)W/2$. Note that when $\epsilon > 1/2$, the lower bound becomes negative. Since queue length regret must be non-negative, we can finally conclude that $\mathcal{R}_T^{\pi}\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \geq \max\Big\{(1-2\epsilon)W/2,\ 0\Big\}$.

A.3 Proof to Theorem 3.3

Let $\mathbf{Q}(t)$, $\mathbf{a}(t)$ and $\mathbf{b}(t)$ be the queue length vector, the arrival vector and the service vector in slot t under the MaxWeight policy, respectively. Also define the potential function

$$\Phi(\mathbf{Q}(t)) = \frac{1}{2}\sum_{i \in \mathcal{N}} Q_i^2(t).$$

We first provide an upper bound on the W-slot drift $\Phi(\mathbf{Q}(t+W)) - \Phi(\mathbf{Q}(t))$.

Lemma A.1. *Let $\mathbf{a}^*(t)$ and $\mathbf{b}^*(t)$ be the arrival vector and the service vector in slot t generated by the best non-causal policy. Then the W-slot drift satisfies*

$$\begin{aligned} &\Phi(\mathbf{Q}(t+W)) - \Phi(\mathbf{Q}(t)) \\ \leq &cW^2 - \epsilon \sum_i \sum_{\tau=t}^{t+W-1} Q_i(\tau) a_i^*(\tau),\ \forall t \in \mathcal{T}, \end{aligned}$$

where $c \triangleq 7NB^2$.

Proof. For any $i \in \mathcal{N}$, if $Q_i(t) \geq \sum_{\tau=t}^{t+W-1} b_i(\tau)$, then

$$Q_i^2(t+W) = \Big[Q_i(t) + \sum_{\tau=t}^{t+W-1} a_i(\tau) - \sum_{\tau=t}^{t+W-1} b_i(\tau)\Big]^2.$$

If $Q_i(t) < \sum_{\tau=t}^{t+W-1} b_i(\tau)$, then

$$\begin{aligned} Q_i^2(t+W) &\leq \Big[Q_i(t) + \sum_{\tau=t}^{t+W-1} a_i(\tau)\Big]^2 \\ &< \Big[\sum_{\tau=t}^{t+W-1} b_i(\tau) + \sum_{\tau=t}^{t+W-1} a_i(\tau)\Big]^2. \end{aligned}$$

Thus, in any case, we have

$$\begin{aligned} Q_i^2(t+W) \leq & \Big[Q_i(t) + \sum_{\tau=t}^{t+W-1} a_i(\tau) - \sum_{\tau=t}^{t+W-1} b_i(\tau)\Big]^2 \\ &+ \Big[\sum_{\tau=t}^{t+W-1} b_i(\tau) + \sum_{\tau=t}^{t+W-1} a_i(\tau)\Big]^2. \end{aligned}$$

Then the W-slot drift is

$$\begin{aligned} &\Phi(\mathbf{Q}(t+W)) - \Phi(\mathbf{Q}(t)) \\ =&\frac{1}{2}\sum_i Q_i^2(t+W) - \frac{1}{2}\sum_i Q_i^2(t) \\ \leq&\frac{1}{2}\sum_i \Big[Q_i(t) + \sum_{\tau=t}^{t+W-1} a_i(\tau) - \sum_{\tau=t}^{t+W-1} b_i(\tau)\Big]^2 \\ &+ \frac{1}{2}\sum_i \Big[\sum_{\tau=t}^{t+W-1} b_i(\tau) + \sum_{\tau=t}^{t+W-1} a_i(\tau)\Big]^2 - \frac{1}{2}\sum_i Q_i^2(t) \\ \leq& \sum_{\tau=t}^{t+W-1} \sum_i Q_i(t)\Big(a_i(\tau) - b_i(\tau)\Big) + 2NB^2W^2. \end{aligned}$$

Note that for any $\tau \in [t, t+W-1]$ and any $i \in \mathcal{N}$ we have

$$Q_i(\tau) - WB \leq Q_i(t) \leq Q_i(\tau) + WB. \tag{13}$$

Then it follows that

$$
\begin{aligned}
&\sum_{\tau=t}^{t+W-1} \sum_{i} Q_i(t)\Big(a_i(\tau) - b_i(\tau)\Big) \\
\leq &\sum_{\tau=t}^{t+W-1} \sum_{i} \Big[\big(Q_i(\tau) + WB\big)a_i(\tau) - \big(Q_i(\tau) - WB\big)b_i(\tau)\Big] \\
= &\sum_{\tau=t}^{t+W-1} \sum_{i} Q_i(\tau)\Big(a_i(\tau) - b_i(\tau)\Big) + 2NB^2W^2 \\
\leq &\sum_{\tau=t}^{t+W-1} \sum_{i} Q_i(\tau)\Big(a_i^*(\tau) - b_i^*(\tau)\Big) + 2NB^2W^2 \\
\leq &\sum_{\tau=t}^{t+W-1} \sum_{i} \Big[(Q_i(t) + WB)a_i^*(\tau) - (Q_i(t) - WB)b_i^*(\tau)\Big] + 2NB^2W^2 \\
\leq &\sum_{i} Q_i(t) \sum_{\tau=t}^{t+W-1} \Big(a_i^*(\tau) - b_i^*(\tau)\Big) + 4NB^2W^2
\end{aligned}
$$

where the first and the third inequalities are due to (13), and the second inequality is due to the MaxWeight policy. As a result, the W-slot drift is

$$
\begin{aligned}
&\Phi(\mathbf{Q}(t+W)) - \Phi(\mathbf{Q}(t)) \\
\leq &\sum_{i} Q_i(t) \sum_{\tau=t}^{t+W-1} \Big(a_i^*(\tau) - b_i^*(\tau)\Big) + 6NB^2W^2 \\
\leq &6NB^2W^2 + \sum_{i} Q_i(t) \sum_{\tau=t}^{t+W-1} \Big(a_i^*(\tau) - \frac{1}{1-\epsilon}a_i^*(\tau)\Big) \\
\leq &6NB^2W^2 - \epsilon \sum_{i} Q_i(t) \sum_{\tau=t}^{t+W-1} a_i^*(\tau) \\
\leq &6NB^2W^2 - \epsilon \sum_{i} \sum_{\tau=t}^{t+W-1} \big(Q_i(\tau) - WB\big)a_i^*(\tau) \\
\leq &cW^2 - \epsilon \sum_{i} \sum_{\tau=t}^{t+W-1} Q_i(\tau)a_i^*(\tau),
\end{aligned}
$$

where the second inequality is due to the definition of (W, ϵ)-constrained adversary, the third inequality is due to (13) and $c \triangleq 7NB^2$. □

We then divide the time horizon into frames of size W slots. The total number of frames is $K = \lceil T/W \rceil$ and let $t_k = (k-1)W$ be the beginning of frame k (where $k = 1, \cdots, K$). Note that the last frame may contain fewer than W slots and thus $T - t_K \leq W$. By Lemma A.1, we have for any $\epsilon \geq 0$:

$$
\Phi(\mathbf{Q}(t_k)) - \Phi(\mathbf{Q}(t_{k-1})) \leq cW^2.
$$

Summing over $k = 2, \cdots, K$ and noticing that $t_1 = 0$ and $\mathbf{Q}(0) = \mathbf{0}$, we have

$$
\Phi(\mathbf{Q}(t_K)) \leq cW^2(K-1) \leq cWT.
$$

Thus, we have

$$\sum_i Q_i(t_K) \leq \sqrt{N}\sqrt{2\Phi(\mathbf{Q}(t_K))} \leq \sqrt{2NcWT}.$$

Since $T - t_K < W$, we have

$$\sum_i Q_i(T) \leq \sqrt{2NcTW} + NBW = O(\sqrt{TW}).$$

Note that the above inequality holds for any sequence of network events that is (W, ϵ)-constrained. Note also that $\sum_i Q_i^*(T) \geq 0$. Therefore, we conclude that the worst-case queue length regret achieved by MaxWeight under the (W, ϵ)-constrained adversary model is $O(\sqrt{TW})$ for any $\epsilon \geq 0$.

Special case: $\epsilon > 0$ and $a_{\min} > 0$.

In the special case where $\epsilon > 0$ and $a_{\min} > 0$, we prove that a better regret bound $O\left(\frac{W}{\epsilon^3 a_{\min}}\right)$ can be achieved by MaxWeight. By Lemma A.1 we have

$$\begin{aligned}
&\Phi(\mathbf{Q}(t_k)) - \Phi(\mathbf{Q}(t_{k-1})) \\
\leq& cW^2 - \epsilon \sum_i \sum_{\tau=t_{k-1}}^{t_k-1} Q_i(\tau) a_i^*(\tau).
\end{aligned}$$

Define

$$\mathcal{K} \triangleq \left\{k \in [1, K-1] \middle| \Phi(\mathbf{Q}(t_k)) \leq c'W^2/\epsilon^2\right\},$$

where c' is some constant to be determined later. Let k^* be the largest index in $\mathcal{K}$. Summing over $k = k^* + 1, \cdots, K$, we have

$$\begin{aligned}
&\Phi(\mathbf{Q}(t_K)) - \Phi(\mathbf{Q}(t_{k^*})) \\
\leq& cW^2(K - k^*) - \epsilon \sum_{\tau=t_{k^*}}^{t_K-1} \sum_i Q_i(\tau) a_i^*(\tau).
\end{aligned}$$

Then it follows that

$$\begin{aligned}
&\sum_{\tau=t_{k^*}}^{t_K-1} \sum_i Q_i(\tau) a_i^*(\tau) \\
\leq& \frac{cW^2(K - k^*)}{\epsilon} + \frac{\Phi(\mathbf{Q}(t_{k^*}))}{\epsilon} \\
\leq& \frac{cW^2(K - k^*)}{\epsilon} + \frac{c'W^2}{\epsilon^3},
\end{aligned}$$

where the last inequality is due to the definition of k^*. Since $a_i(t) \geq a^{\min} > 0$, we have

$$\sum_{t=t_{k^*}}^{t_K-1} \sum_{i=1}^{N} Q_i(t) \leq \frac{cW^2(K - k^*)}{\epsilon a^{\min}} + \frac{c'W^2}{\epsilon^3 a^{\min}}. \tag{14}$$

Note that by the definition of k^*, we have for any $k = k^* + 1, \cdots, K - 1$

$$\sum_{i=1}^{N} Q_i(t_k) \geq \sqrt{2\Phi(\mathbf{Q}(t_k))} \geq \sqrt{2c'W^2/\epsilon^2}. \tag{15}$$

Note also that the maximum increase or decrease in the total queue length in each slot is at most NB. Define

$$J \triangleq \min\left\{W, \frac{\sqrt{2c'W^2/\epsilon^2}}{NB}\right\},$$

where the value of c' will be chosen such that J is an integer. Then it follows that $(J-1)NB \leq \sqrt{2c'W^2/\epsilon^2}$. As a result, we have for any $k = k^* + 1, \cdots, K-2$

$$\begin{aligned}
&\sum_{t=t_k}^{t_{k+1}-1} \sum_{i=1}^{N} Q_i(t) \\
\geq &\sum_{j=0}^{J-1} \left(\sqrt{2c'W^2/\epsilon^2} - jNB\right) \\
= &J\sqrt{2c'W^2/\epsilon^2} - \frac{1}{2}(J-1)JNB \\
\geq &\frac{1}{2}\sqrt{2c'W^2/\epsilon^2}J \\
= &\min\left\{\frac{c'W^2}{\epsilon^2 NB}, \frac{W^2}{2\epsilon}\sqrt{2c'}\right\},
\end{aligned} \tag{16}$$

where the first inequality is due to (15) and the fact that maximum decrease in the total queue length in each slot is at most NB. Therefore, we have

$$\begin{aligned}
&\sum_{t=t_{K-1}}^{t_K-1} \sum_{i=1}^{N} Q_i(t) \\
\leq &\sum_{t=t_{k^*}}^{t_K-1} \sum_{i=1}^{N} Q_i(t) - \sum_{t=t_{k^*}}^{t_{K-1}-1} \sum_{i=1}^{N} Q_i(t) \\
\leq &\frac{cW^2(K-k^*)}{\epsilon a^{\min}} + \frac{c'W^2}{\epsilon^3 a^{\min}} - \frac{W}{2\epsilon}\sqrt{2c'}J(K-k^*-2),
\end{aligned}$$

where the last inequality is due to (14) and (16). We choose

$$c' \geq \max\left\{\frac{cNB}{a^{\min}}, \frac{2c^2}{(a^{\min})^2}\right\}$$

such that

$$\frac{W}{2\epsilon}\sqrt{2c'}J \geq \frac{cW^2}{\epsilon a^{\min}}.$$

Then it follows that

$$\sum_{t=t_{K-1}}^{t_K-1} \sum_{i=1}^{N} Q_i(t) \leq \frac{2cW^2}{\epsilon a^{\min}} + \frac{c'W^2}{\epsilon^3 a^{\min}}.$$

Denote by $X = \sum_i Q_i(t_K - 1)$, and similarly define $J' = \min\left\{W, \frac{X}{NB}\right\}$. Then

$$\begin{aligned}
&\sum_{t=t_{K-1}}^{t_K-1} \sum_{i=1}^{N} Q_i(t) \\
\geq &\sum_{j=0}^{J'-1} \Big(X - jNB\Big) \\
\geq &\frac{J'X}{2} = \min\left\{\frac{WX}{2}, \frac{X^2}{2NB}\right\}
\end{aligned}$$

As a result, we have

$$\min\left\{\frac{WX}{2}, \frac{X^2}{2NB}\right\} \leq \frac{2cW^2}{\epsilon a^{\min}} + \frac{c'W^2}{\epsilon^3 a^{\min}},$$

where we can solve

$$X \leq \max\left\{\frac{4cW}{\epsilon a^{\min}} + \frac{2c'W}{\epsilon^3 a^{\min}}, \sqrt{2NB\Big(\frac{2cW^2}{\epsilon a^{\min}} + \frac{c'W^2}{\epsilon^3 a^{\min}}\Big)}\right\}.$$

Therefore, we have $\sum_i Q_i(t_K - 1) = O\Big(\frac{W}{\epsilon^3 a_{\min}}\Big)$. Since $T - t_K \leq W$, we have

$$\sum_i Q_i(T) \leq \sum_i Q_i(t_K - 1) + NBW = O\Big(\frac{W}{\epsilon^3 a_{\min}}\Big).$$

Then we conclude that the worst-case queue length regret of the MaxWeight policy under the (W, ϵ)-constrained adversary model is $O\Big(\frac{W}{\epsilon^3 a_{\min}}\Big)$ if $\epsilon > 0$ and $a_{\min} > 0$.

A.4 Proof to Lemma 3.6

Time is divided into frames of size W slots. Note that by the Tracking algorithm the debt owned in each frame will be added in batch at the beginning of the next frame (or equivalently at the end of the current frame). Thus, there is no debt arrival *within* a frame, and the debt queue length in any slot within a frame is upper bounded by the debt queue length at the beginning of that frame. Therefore, it is sufficient to prove that at the beginning of each frame r, the total debt associated with event ω is at most NWB.

At the beginning of frame 1 (i.e., $t = 0, \cdots, W - 1$), the total debt associated with event ω is 0, because we initialize $q_{i,\omega}(0) = 0$ for any $i \in \mathcal{N}$.

At the beginning of frame 2, the total debt associated with event ω is the new debt owned in frame 1 (since the debt owned in a frame will be added in batch at the beginning of the next frame). Suppose event ω occurs m_1 times in frame 1. Then the total debt associated with event ω owned in frame 1 is at most NBm_1, and thus the total debt associated with event ω at the beginning of frame 2 is also at most $Nm_1B \leq NWB$.

At the beginning of frame 3, the total debt associated with event ω is the sum of the remaining debt from frame 1 and the new debt generated in frame 2. Suppose event ω occurs m_2 times in frame 2. We discuss two scenarios.

- If $m_2 \geq m_1$, then the debt owned in frame 1 will be cleared during the first m_1 occurrences of event ω in frame 2, since the Tracking algorithm clears as much total debt for event ω as possible every time event ω occurs. As a result, the remaining debt from frame 1 is zero at the end of frame 2. At the same time, the new debt owned in frame 2 is at most Nm_2B. As a result, at the beginning of frame 3, the total debt associated with event ω is at most Nm_2B.

• If $m_2 < m_1$, then the debt owned in the first m_2 occurrences of even ω in frame 1 will be cleared by the similar argument as in the first scenario. As a result, the remaining debt from frame 1 at the end of frame 2 is at most $N(m_2 - m_1)B$. At the same time, the new debt owned in frame 2 is at most Nm_2B. Therefore, the total debt associated with event ω at the beginning of frame 3 is at most $N(m_1 - m_2)B + Nm_2B = Nm_1B$.

Therefore, in both of the above scenarios, we have that at the beginning of frame 3 the total debt associated with event ω is at most

$$\max\{m_1, m_2\}NB \leq NWB.$$

Similar argument applies to any of the subsequent frame $r \geq 3$: at the beginning of frame r, the total debt associated with event ω is at most

$$NB \max_{j=1,\cdots,r-1} m_j \leq NWB.$$

This concludes our proof.

A.5 Proof to Theorem 4.2

For any causal policy π, we construct a sequence of network events $\{\omega_0, \cdots, \omega_{T-1}\}$ such that $V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \leq V_T$, but $\mathcal{R}_T^{\pi}\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \geq cV_T$. Define $Z \triangleq \frac{V_T}{NB}$. Since $NB \geq 1$ and $V_T \leq NTB$, we have $Z \leq V_T$ and have $Z \leq T$. Without loss of generality, assume that $Z/2$ is an integer.

Consider the power control example mentioned in Section 2.2 with $N = 2$ links. The constraint on power allocation is $\alpha_t^{(1)} + \alpha_t^{(2)} \leq 1$ for each $t \in \mathcal{T}$, and the rate-power function is $b_i(t) = \alpha_t^{(i)} s_i(t)$. The exogenous arrivals and channel capacities in the first $Z/2$ slots is

$$a_1(t) = a_2(t) = 2,\ s_1(t) = s_2(t) = 2,\ \forall t = 0, \cdots, Z/2 - 1.$$

Under the peak power constraint, the total number of packets that can be cleared in the first $Z/2$ slots is at most Z. For any causal policy π, let n_1 and n_2 be the number of packets cleared over link 1 and 2 during the first $Z/2$ slots, respectively. Then it is clear that $n_1 + n_2 \leq Z$, which implies that $\min\{n_1, n_2\} \leq Z/2$. Define $i^* = \arg\min_{i=1,2} n_i$ (ties are broken arbitrarily). Then the queue length over link i^* after $Z/2$ slots is

$$Q_{i^*}^{\pi}(Z/2) = Z - n_{i^*} = Z - \min\{n_1, n_2\} \geq Z/2.$$

In the next $T - Z/2$ slots, the adversary can set

$$a_{i^*}(t) = 0,\ s_{i^*}(t) = 0,\ t = Z/2, \cdots, T-1.$$

For the other link (its index is denoted by i'), the adversary can set

$$a_{i'}(t) = 0,\ s_{i'}(t) = 2,\ t = Z/2, \cdots, T-1.$$

Since there is no capacity to clear any packet over link i^* in the last $T - Z/2$ slots, we have

$$Q_{i^*}^{\pi}(T) = Q_{i^*}^{\pi}(Z/2) \geq Z/2,$$

which implies that $\sum_i Q_i^{\pi}(T) \geq Z/2$. On the other hand, the optimal non-causal policy π^* can choose the following sequence of power allocation vectors

$$\left(\alpha_t^{(i^*)}, \alpha_t^{(i')}\right) = \begin{cases} (1, 0),\ t = 0, \cdots, Z/2 - 1, \\ (0, 1),\ t = Z/2, \cdots, T-1, \end{cases}$$

such that $\sum_i Q_i^*(T) = 0$, which implies that

$$\mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \geq Z/2 = \frac{V_T}{2NB} \triangleq cV_T.$$

Now we evaluate the value of $V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big)$ for the above network dynamics. Clearly, the total queue length under any (possibly non-causal) policy after the first $Z/2$ slots is at least Z, i.e., $V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \geq Z$. At the same time, note that under the optimal non-causal policy π^*, we have $V^{\pi^*}\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = Z$, which implies that $V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \leq Z$. Therefore, we conclude that $V\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = Z \leq V_T$.

A.6 Proof to Observation 1

Suppose that the sequence of network events is $\omega_0, \omega_1, \cdots$. If the network is rate-stable, then (11) implies that $\sum_i Q_i^\pi(T) = o(T)$ as $T \to \infty$. In any V_T-constrained network with $V_T = o(T)$, we also have $\sum_i Q_i^*(T) = o(T)$. By the definition of queue length regret, we have

$$\mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = \sum_i Q_i^\pi(T) - \sum_i Q_i^*(T) = o(T).$$

Conversely, if the queue length regret is sublinear, then we have

$$\sum_i Q_i^\pi(T) = \sum_i Q_i^*(T) + \mathcal{R}_T^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) = o(T),$$

which implies that equation (11) holds.

A.7 Proof to Theorem 5.3

We only show the necessity since the sufficiency has been given in Corollary 5.2. If the network is rate-stable under a certain policy π, then we have

$$\lim_{T\to\infty} \frac{\sum_i Q_i^\pi(T)}{T} = 0,$$

i.e., for any $\epsilon > 0$, there exists some $T_0 \geq 0$ such that for any $T \geq T_0$ we have $\sum_i Q_i^\pi(T) \leq \epsilon T$.

For any $T \geq T_0$, we also have

$$\begin{aligned}\max_{t\leq T} \sum_i Q_i^\pi(t) &= \max\Big\{\max_{t<T_0} \sum_i Q_i^\pi(t), \max_{T_0\leq t\leq T} \sum_i Q_i^\pi(t)\Big\} \\ &\leq \max\{T_0 NB, \epsilon T\}.\end{aligned}$$

Then it follows that for any $\epsilon > 0$, there exists some $T_1 \geq \frac{T_0 NB}{\epsilon}$ such that for any $T \geq T_1$

$$\frac{\max_{t\leq T} \sum_i Q_i^\pi(t)}{T} \leq \max\Big\{\frac{T_0 NB}{T}, \epsilon\Big\} = \epsilon,$$

which implies that

$$\lim_{T\to\infty} \frac{\max_{t\leq T} \sum_i Q_i^\pi(t)}{T} = 0.$$

As a result, we can conclude that as $T \to \infty$

$$\begin{aligned}v\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) &\leq v^\pi\Big(\{\omega_0, \cdots, \omega_{T-1}\}\Big) \\ &= \max_{t\leq T} \sum_i Q_i^\pi(t) = o(T).\end{aligned}$$

This completes the proof.