Fundamental Limits of Volume-based Network DoS Attacks

XINZHE FU, LIDS, Massachusetts Institute of Technology, USA
EYTAN MODIANO, LIDS, Massachusetts Institute of Technology, USA

50

Volume-based network denial-of-service (DoS) attacks refer to a class of cyber attacks where an adversary seeks to block user traffic from service by sending adversarial traffic that reduces the available user capacity. In this paper, we explore the fundamental limits of volume-based network DoS attacks by studying the minimum required rate of adversarial traffic and investigating optimal attack strategies. We start our analysis with single-hop networks where user traffic is routed to servers following the Join-the-Shortest-Queue (JSQ) rule. Given the service rates of servers and arrival rates of user traffic, we first characterize the feasibility region of the attack and show that the attack is feasible if and only if the rate of the adversarial traffic lies in the region. We then design an attack strategy that is (i). *optimal:* it guarantees the success of the attack whenever the adversarial traffic rate lies in the feasibility region and (ii). *oblivious:* it does not rely on knowledge of service rates or user traffic rates. Finally, we extend our results on the feasibility region of the attack and the optimal attack strategy to multi-hop networks that employ Back-pressure (Max-Weight) routing. At a higher level, this paper addresses a class of dual problems of stochastic network stability, i.e., how to optimally de-stabilize a network.

Additional Key Words and Phrases: Denial-of-Service Attacks; Stochastic Network Scheduling; Network Queueing Theory

ACM Reference Format:
Xinzhe Fu and Eytan Modiano. 2019. Fundamental Limits of Volume-based Network DoS Attacks. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 3, Article 50 (December 2019), 36 pages. https://doi.org/10.1145/3366698

1 INTRODUCTION

1.1 Background and Motivation

Network denial-of-service (DoS) attacks, where an adversary seeks to make some network resource unavailable to its intended users, is one of the most serious security threats to the Internet. It often results in downtime of web services, cloud computing facilities, DNS services, etc., causing huge financial loss to institutions [1]. While some network DoS attacks exploit the vulnerabilities of protocols, the predominant type of attacks are volume-based, such as TCP SYN Flood, UDP Flood and DNS Flood [2]. They work by flooding the network with adversary traffic and blocking the service to normal users [2]. Such adversary traffic can be generated distributively from botnets and is difficult to distinguish from normal user traffic [4], which makes volume-based DoS attacks difficult to defend against. Due to the significance and prevalence of volume-based network DoS attacks, there have been a flurry of works focusing on their detection and mitigation [3, 5, 6]. However, a theoretical understanding of the limits of such attacks is still lacking, i.e., **how much resources does the adversary need for mounting a successful volume-based network DoS attack and what is the optimal attack strategy?**

Authors' addresses: Xinzhe Fu, LIDS, Massachusetts Institute of Technology, USA; Eytan Modiano, LIDS, Massachusetts Institute of Technology, USA.

2476-1249/2019/12-ART50 $15.00
https://doi.org/10.1145/3366698

Understanding the above questions is of great importance to the design and protection of networked systems. It would provide us with valuable insights regarding the robustness of the network, as the resource requirements for the adversary define the safety margin of the system. Furthermore, the structure of the optimal attack strategy sheds light on the design of practical detection and mitigation methods.

In this paper, we explore the fundamental limits of volume-based network DoS attacks. Taking a network flow and queueing perspective, we translate the scenario of network DoS attacks to one where the adversary injects traffic and seeks to de-stabilize the network by overflowing network queues. As we will show, such perspective closely mirrors volume-based DoS attacks in real life and enables us to conveniently inherit the modeling and analysis tools from the network flow and queueing literature. That being said, two features of our problem make it elude traditional frameworks in the existing literature. First, a network under attack often operates outside its stability region, requiring tools for analyzing networks in overload conditions. This renders the results from stochastic/adversarial queueing theory [7–9] and stochastic/adversarial network optimization [19, 20] inapplicable. Second, networked systems often employ dynamic load-balancing or scheduling mechanism (e.g., Join the Shortest Queue and Max-Weight) [11, 12, 18] which have a great impact on the optimal attack strategy. However, this aspect has been largely overlooked by previous network attack-defense frameworks such as network interdiction, which studies the problem of minimizing the max-flow of a capacitated network by removing network links [13, 15] or sending adversarial traffic flows [14]; and network security games, which adopt a game-theoretic perspective and focus on equilibrium analysis [16] and security mechanism design [17]. A notable exception is the paper by Paschos and Tassiulas [21] that also inspired our work, which studied the sustainability of networks with static routing under volume-based DoS attack.

1.2 Contribution

We start our analysis with a server farm which can be modeled as a single-hop network. The network has a general bipartite topology, where one side consists of traffic dispatchers (of user and adversary traffic) and the other side consists of servers with jobs queued at each server waiting for service. User traffic arrives at each user dispatcher and is sent to the servers following the Join-the-Shortest-Queue (JSQ) rule. Adversary traffic is sent by adversary dispatchers to the servers under some adversarial injection policy. The servers are not able to distinguish user and adversary traffic and employ the FCFS service discipline. The goal of the adversary is to block user traffic from being served, i.e., to cause user traffic in one or more queues to grow to infinity. We say that an attack is feasible if there exists an injection strategy for the adversary to achieve its goal (see Section 2 for rigorous definitions). Our model can represent web-servers where the queues correspond to service request buffers, the user dispatchers correspond to load-balancers and the adversary dispatchers correspond to initiators of the attack. Our model can thus be seen to capture many DoS attack scenarios such as TCP SYN Flood and DNS Flood [2]. Under this model, we obtain the following main results:

(1) We give a necessary and sufficient condition on the user traffic rates, adversary traffic rates and servers' service rates for the feasibility of network DoS attack, which can be interpreted as the resource requirement for mounting a successful attack.
(2) We design an optimal adversarial injection policy that achieves the goal of the network DoS attack whenever the feasibility condition is met. Our policy is oblivious, in the sense that it does not require knowledge of network statistics, including user traffic rates, adversary traffic rates and service rates.

(3) We generalize our results to multi-hop networks that employs the back-pressure routing policy [18]. We extend the feasibility condition and the optimal adversarial injection policy to the multi-hop scenario.

Our methodology also forms a general recipe for solving the problem of optimal de-stabilization of stochastic networks, which can be considered as a dual problem of stochastic network stability [10].

Our work is closely related to [21], where Paschos and Tassiulas defined the guaranteed user throughput region of networks under DoS attack with fixed adversary traffic rates. In contrast, we present a dual result, the minimum resource requirement for the attack to be feasible under fixed user traffic rates. They also considered the case where user traffic is routed following the JSQ rule and proposed a heuristic adversarial injection policy that they conjectured to be optimal, while the policy we propose is provably optimal (See Section 5 for a more detailed discussion).

The rest of the paper is organized as follows. In Section 2, we formally present our model and problem formulation. We then introduce the feasibility region in Section 3. We summarize several key properties of the Join-the-Shortest-Queue policy in Section 4, which will be instrumental in analyzing the optimal adversarial injection policy we propose. We introduce the policy in Section 5, and evaluate the injection policy by simulations in Section 6. Section 7 is devoted to generalization to multi-hop networks. We conclude the paper in Section 8.

2 MODEL AND PROBLEM FORMULATION

In this section, we formally present our system model for single-hop networks, which captures server farms as a major application. The model for multi-hop networks will be presented in Section 7. The notations that we use throughout the paper are summarized in Table 1.

2.1 Network Model

As our single-hop network model mainly mirrors server farms, we will use single-hop network and server farm interchangeably. Consider a single-hop network with a set of parallel servers (sinks) and a set of traffic dispatchers (sources). The dispatchers are divided into two disjoint subsets: user traffic dispatchers that route user traffic to servers, and adversary traffic dispatchers, controlled by the adversary, that send adversary traffic to servers to block the user traffic. We use $S = \{s_1, \ldots, s_N\}$ to denote the set of servers, $U = \{u_1, \ldots, u_L\}$ to denote the set of user traffic dispatchers and $V = \{v_1, \ldots, v_M\}$ to denote the set of adversary traffic dispatchers. A generic server, a generic user traffic dispatcher and a generic adversary traffic dispatcher are denoted by s_n or n, u_l or l, v_m or m, respectively. Let $S_{u_l} \subseteq S$ be the set of servers that user dispatcher u_l is connected to, and $S_{v_m} \subseteq S$ be the set of servers that adversary dispatcher v_m is connected to. Each dispatcher can only route jobs/packets to the servers to which it is connected. Finally, for consistency, we will refer to the "jobs" sent by dispatchers as packets and assume that all packets have the same length, which corresponds to jobs of equal size. Extension to varying packet lengths is straightforward.

2.2 Queueing Dynamics

We consider a discrete-time system with time t starting from 0. Each server has a infinite-size queue that buffers the packets, with $Q_n(t)$ representing the length of the queue of server s_n at time t. The offered service of server n at time t is denoted by $b_n(t)$. The servers do not distinguish user and adversary traffic and employ the First-Come-First-Serve (FCFS) service discipline[1]. In each time slot, $\lambda_l^u(t)$ packets arrive at user dispatcher u_l, which routes the packets to the servers

[1]Our results hold under all common service disciplines except priority based service with user traffic having the priority.

Table 1. Notations and Definitions

Notation	Definition
S, U, V	Sets of servers, user dispatchers and adversary dispatchers
N, L, M	Numbers of servers, user dispatchers and adversary dispatchers
$s_n, n; u_l, l; v_m, m$	generic server, user dispatcher and adversary dispatcher
S_{u_l}, S_{v_m}	The set of servers u_l (v_m) has connection to
$Q_n(t)$	Queue length at server n at time t
$b_n(t), \mu_n$	Offered service of server n at time t and its mean
$\lambda_l^u(t), \lambda_u^l$	User traffic arrival at u_l at time t and its mean
$\lambda_m^v(t), \lambda_v^m$	Adversary traffic arrival at v_m at time t and its mean
C	Upper bound of $\lvert b_n(t)\rvert$, $\lvert\lambda_n^u(t)\rvert$, $\lvert\lambda_n^v(t)\rvert$
$b_n^u(t), b_n^v(t)$	Offered service for user (adversary) traffic of server n at time t
$a_n^u(t), a_n^v(t)$	Total user (adversary) packets routed to server n at time t
$a_{ln}^u(t), a_{mn}^v(t)$	Amount of user (adversary) packets routed from $u_l(v_m)$ to n at t
$Q_n^u(t), Q_n^v(t)$	Amount of user (adversary) packets in Q_n at time t
$\boldsymbol{Q}(t)$	Queue length vector at time t
$\boldsymbol{\mu}, \boldsymbol{\lambda^u}, \boldsymbol{\lambda^v}$	Vectors of service rates, user traffic arrival rates and adversary budget
$U_{S'}$	Set of user dispatchers that only have connections to servers in S'

following the "Join-the-Shortest-Queue" (JSQ) policy, that is, at each time slot, each user dispatcher u_l routes all its incoming packets to the server s with the minimum queue length among the ones to which it is connected ($s \in \arg\min_{s_n \in S} Q_n(t)$); Similarly, $\lambda_m^v(t)$ packets arrive at adversary dispatcher v_m, which routes the packets to servers according to some adversarial injection policy. We assume that $b_n(t)$'s, $\lambda_l^u(t)$'s and $\lambda_m^v(t)$'s are independent sequences of i.i.d. random variables with $\mathbb{E}[b_n(t)] = \mu_n, \mathbb{E}[\lambda_l^u(t)] = \lambda_l^u, \mathbb{E}[\lambda_m^v(t)] = \lambda_m^v$. We assume that the random variables are bounded, i.e., there exists $C > 0$ such that $0 \le b_n(t), \lambda_l^u(t), \lambda_m^v(t) \le C$. We further define $Q_n^u(t)$ and $Q_n^v(t)$ as the number of user packets and adversary packets in Q_n at t, respectively. At each time slot t, we decompose the offered service $b_n(t)$ into that offered to user traffic $b_n^u(t)$ and that offered to adversary traffic $b_n^v(t)$ with $b_n^u(t) + b_n^v(t) = b_n(t)$. Under the FCFS service discipline, the breakdown between $b_n^u(t)$ and $b_n^v(t)$ only depends on the queue composition. We further define $a_n^u(t)$ as the sum of user traffic arrivals to server n and $a_n^v(t)$ as the counterpart of adversary traffic. we also write $a_{mn}^u(t)$ ($a_{ln}^v(t)$) as the amount traffic that user dispatcher u_l (adversary dispatcher v_m) sends to n at time t. We impose the following ordering on system dynamics for ease of presentation: in each time slot, first, user dispatchers route their incoming packets to the servers following JSQ; second, adversary dispatchers route adversary packets to the servers following some adversarial injection policy; finally, servers serve the packets in the queues. Based on the system dynamics, we summarize the queue length evolution as follows:

$$
\begin{aligned}
Q_n^u(t+1) &= [Q_n^u(t) + a_n^u(t) - b_n^u(t)]^+, \\
Q_n^v(t+1) &= [Q_n^v(t) + a_n^v(t) - b_n^v(t)]^+, \\
Q_n(t+1) &= Q_n^v(t+1) + Q_n^u(t+1),
\end{aligned}
$$

where $[a]^+ := \max\{a, 0\}$. We remind the reader that user traffic and adversary traffic are buffered in a single queue at each server, and Q_n^u, Q_n^v's represent the composition of user and adversary packets in the single queue rather than two separate queues. We give an illustration of our model in **Figure 1**.

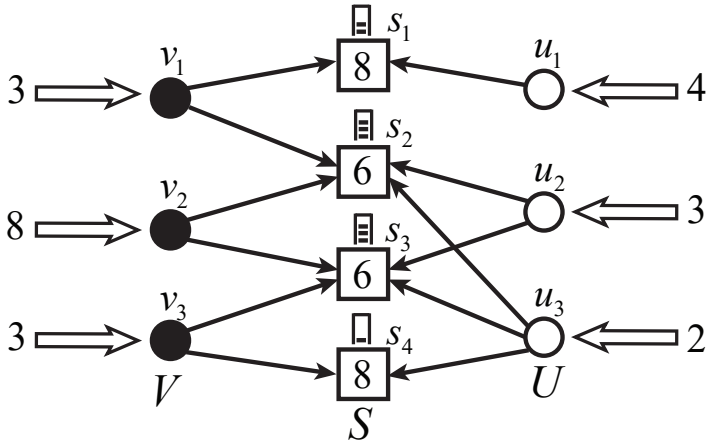

Fig. 1. Illustration of our single-hop network model. User dispatchers are represented by hollow circles. Adversary dispatchers are represented by solid circles. Servers are represented by rectangles. The numerical values in the graph represent the traffic arrival rates to user/adversary dispatchers and the service rates of servers.

2.3 Problem Formulation

The adversary dispatchers inject their packets to servers in an effort to prevent user packets from getting served. A network DoS attack is considered successful if the adversary manages to block a positive fraction of user traffic from service. Formally, the goal of the adversary is that

$$\text{For some } n \in \{1, \dots, N\}, \qquad \lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0, \tag{1}$$

which is equivalent to making user traffic in one of the queues mean rate-unstable [10].[2] Furthermore, by Little's law, (1) implies that the mean delay experienced by user traffic grow linearly with time. We say that the adversary *destabilizes* user traffic, if it achieves (1). Note that (1) implicitly relies on the existence of the limit. In this paper, we will assume that the limits $\lim_{t\to\infty} \frac{Q_n^u(t)}{t}$ and $\lim_{t\to\infty} \frac{Q_n^v(t)}{t}$ exists almost surely. We make this assumption to simplify the notation and avoid unnecessary complexity in derivation. This also implicitly restricts the space of adversarial injection policies to be stationary, i.e., time-invariant. When this assumption does not hold, one can replace (1) with lim inf and our results will still hold with some minor changes, which will be explained in Appendix E.1. Under the assumption, we have $\mathbb{E}[\lim_{t\to\infty} \frac{Q_n^u(t)}{t}] = \lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t}$, which suggests that (1) provides a unified metric for the growth rate of user traffic. (This will be formally justified in Appendix D.)

In an instance of network DoS attack, μ_n's and λ_l^u's can be seen as network statistics while λ_m^v's can be viewed as the adversary's resource budget since it dictates how many packets the adversary dispatchers can inject to servers. We summarize the statistics and budget into vector forms as the service vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_N)$, user traffic arrival vector $\boldsymbol{\lambda}^u = (\lambda_1^u, \dots, \lambda_L^u)$ and adversary budget vector $\boldsymbol{\lambda}^v = (\lambda_1^v, \dots, \lambda_M^v)$. We will also write the queue lengths in vector form $\boldsymbol{Q}(t) = (Q_1(t), \dots, Q_N(t))$.

[2]We adopt mean rate-instability instead of weaker criteria such as $\lim_{t\to\infty} \mathbb{E}[Q_n^u(t)] = \infty$ because of (i). mean rate-instability leads to cleaner analysis, and (ii). it captures the loss of user throughput more accurately since if a user queue satisfies $\lim_{t\to\infty} \mathbb{E}[Q_n^u(t)] = \infty$ but not $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$, then the users only lose a vanishing fraction of throughput. We further discuss this in Appendix E.2.

Based on the above preliminaries, we formally define the network DoS attack problem.

Definition 1 (Network DoS Attack Problem). *Given a single-hop network, the Network DoS Attack Problem seeks an adversarial injection policy that destabilizes user traffic, i.e., under the policy there exists some server n with $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$. The problem is **feasible** if such an injection policy exists.*

Example: Consider the network in Figure 1, the network DoS attack problem is feasible. An injection policy that destabilizes user traffic is as follows: v_1 injects all of its traffic to s_2. v_2 injects half of its traffic to s_2 and the other half to s_3. v_3 injects all its traffic to s_3. Note that if v_1 and v_2 both inject all of their traffic to s_2, this will cause the queue in s_2 to overflow, but will not destabilize user traffic. The (intuitive) explanation is that, as the user dispatchers are using JSQ, the user traffic from u_2 and u_3 will not be sent to s_2 (in equilibrium state). Since s_3 and s_4 have large enough capacities, there will not be any queue with $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$.

3 FEASIBILITY REGION

In this section, we develop a necessary and sufficient condition on the network statistics and the adversary's budget vector for the network DoS attack to be feasible. For a given network, the condition characterizes the feasibility region of the adversary. We begin by making some preliminary definitions.

For each subset of servers $S' \subseteq S$, we define $U_{S'}$ as the user dispatchers that only have connections to servers in S', i.e., $U_{S'} = \{u_l \mid S_{u_l} \subseteq S'\}$. We further define $\Delta(S')$ as

$$\Delta(S') = \sum_{s_n \in S'} \mu_n - \sum_{u_l \in U_{S'}} \lambda_l^u.$$

$\Delta(S')$ can be interpreted as the excess service rate of S' with respect to the user traffic generated by $U_{S'}$. Finally, for each $S' \subseteq S$, we define the following linear program $LP(S')$ whose optimal value is denoted as $val(S')$.

$$val(S') = \max \sum_{m\in V} \sum_{n\in S'} f_{mn} \tag{2}$$

$$\textbf{s.t.} \sum_{n\in S'} f_{mn} \le \lambda_m^v, \qquad \forall m \in V \tag{3}$$

$$\sum_{m\in V} f_{mn} \le \mu_n, \qquad \forall n \in S' \tag{4}$$

$$f_{mn} = 0, \qquad \text{if } n \notin S_{v_m}$$

$$f_{mn} \ge 0, \qquad \forall m \in V, n \in S'.$$

$val(S')$ can be interpreted as the maximum amount of traffic that the adversary dispatchers can send to S' without exceeding the budget constraints (Constraint (3)) or injecting to any server at a rate larger than its service (Constraint (4)). It will be clear soon that $val(S')$ represents the maximum meaningful capacity reduction that the adversary dispatchers can inflict on S', and it can be achieved by a stationary injection policy given by the solution to $LP(S')$. Relating $\Delta(S')$ and $val(S')$, we define the val-condition, which will play a key role in the characterization of the feasibility region.

Definition 2 (The val-condition). *A subset of servers $S' \subseteq S$ satisfies the val-condition if $U_{S'}$ is non-empty and $val(S') > \Delta(S')$.*

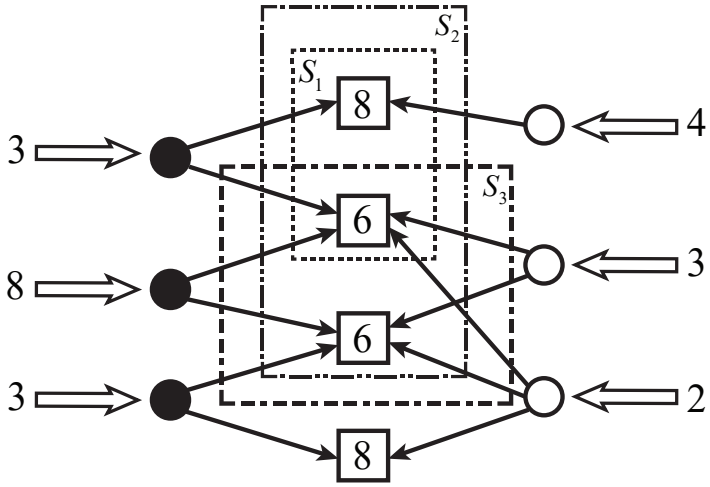

Fig. 2. Illustration of the *val*-condition. Consider three subsets of servers S_1, S_2 and S_3 enclosed in dashed rectangles. $val(S_1) = 9, val(S_2) = 14, val(S_3) = 12$ and $\Delta(S_1) = 10, \Delta(S_2) = 13, \Delta(S_3) = 9$. Thus, S_2 and S_3 satisfy the *val*-condition while S_1 does not.

We provide an illustration of the *val*-condition in **Figure 2** (with the same network as in Figure 1).

Intuitively, if S' satisfies the *val*-condition, then it is possible for the adversary to make the residual capacity of S' not sufficient to support the incoming user traffic of $U_{S'}$, thus successfully blocking user traffic since the user traffic from $U_{S'}$ can only go to S'. It is then natural to consider that the network DoS attack problem is feasible if and only if there exists a subset of user dispatchers that satisfies the *val*-condition. We formalize the intuition in the following theorem.

Theorem 1. *The network DoS problem is feasible if and only if there exists a subset of servers $S' \subseteq S$ that satisfies the val-condition.*

Proof. The proof is divided into two parts. In the first part, we prove the sufficiency of the *val*-condition by showing that, if there exists $S' \subseteq S$ that satisfies the condition, then the stationary injection policy induced by $LP(S')$ destabilizes user traffic. In the second part, we prove the necessity of the *val*-condition by starting from any given adversarial injection policy that destabilizes user traffic and taking time averages, which will lead to establishing that some S' satisfies the *val*-condition.

We begin the proof by noting that a user traffic dispatcher being connected to an overloaded server does not imply that the user traffic from that dispatcher is blocked, since the dispatcher may send the traffic to other servers that it is connected to. However, a positive fraction of user traffic will be blocked if there is a user traffic dispatcher that is only connected to overloaded servers. This observation follows directly from the FCFS service discipline and will be used multiple times throughout the paper. We formally state it as follows.

Observation 1. *Consider a user dispatcher u_l. If on a sample path ω of the system, $\lim_{t\to\infty} Q_n(t)/t > 0$ for all $n \in S_{u_l}$,[3] then there exists $n \in S_{u_l}$ such that $\lim_{t\to\infty} Q^u_n(t)/t > 0$.*

[3]In this paper, we often use the same symbol for both random variables and their realizations on sample paths for notational convenience, e.g. $Q_n(t)$ as $Q_n(t, \omega)$

Proof of Sufficiency: If there exists a $S' \subseteq S$ such that $U_{S'}$ is non-empty and $val(S') > \Delta(S')$, we denote the solution to $LP(S')$ as $\{f^*\}_{mn}$. Consider the following randomized injection policy for adversary: at every time slot, each adversary traffic dispatcher m that has connection to S' injects its traffic to server $n \in S'$ with probability $f^*_{mn} / \sum_{n'} f^*_{mn'}$; other adversary dispatchers inject the traffic arbitrarily. We proceed to show that such policy destabilizes user traffic, i.e., under the injection policy, there exists a server n such that $\lim_{t\to\infty} \frac{\mathbb{E}[Q^u_n(t)]}{t} > 0$.

First, examining the servers in S', we have

$$\begin{aligned}
&\frac{\sum_{n\in S'} Q_n(t)}{t} \geq \frac{\sum_{n\in S'} \sum_{i=0}^{t-1} a_n(i) - \sum_{n\in S'} \sum_{i=0}^{t-1} b_n(i)}{t} \\
=&\sum_{n\in S'} \left(\frac{\sum_{i=0}^{t-1} a^u_n(i)}{t} + \frac{\sum_{i=0}^{t-1} a^v_n(i)}{t} - \frac{\sum_{i=0}^{t-1} b_n(i)}{t} \right) \\
=&\sum_{i=0}^{t-1} \left(\frac{\sum_{n\in S'} a^u_n(i)}{t} + \frac{\sum_{n\in S'} a^v_n(i)}{t} - \frac{\sum_{n\in S'} b_n(i)}{t} \right)
\end{aligned}$$

Let t go to infinity in the above inequalities. By law of large numbers [10], the limits exist with probability one. Therefore, on each sample path (except a set of measure zero), we have

$$\begin{aligned}
&\lim_{t\to\infty} \frac{\sum_{n\in S'} Q_n(t)}{t} \\
\geq &\lim_{t\to\infty} \sum_{i=0}^{t-1} \left(\frac{\sum_{n\in S'} a^u_n(i)}{t} + \frac{\sum_{n\in S'} a^v_n(i)}{t} - \frac{\sum_{n\in S'} b_n(i)}{t} \right) \\
\geq &\sum_{l\in U_{S'}} \lambda^u_l + \sum_{m\in V} \sum_{n\in S'} f^*_{mn} - \sum_{n\in S'} \mu_n \\
= &val(S') - \Delta(S') > 0, \qquad (5)
\end{aligned}$$

where the last part of (5) follows from that S' satisfies the val-condition. From Inequality (5), we claim the following, which will lead to the first part of the proof.

Claim 1. *With probability 1, there exists $n \in S'$ such that $\lim_{t\to\infty} \frac{Q^u_n(t)}{t} > 0$.*

Proof of Claim 1 (Sketch): On each sample path, by Inequality (5), we show that we can find a user dispatcher u_l such that $\forall n \in S_{u_l}$, $\lim_{t\to\infty} \frac{Q_n(t)}{t} > 0$. The Claim then follows from Observation 1. See Appendix A.1 for details. □

Now, for each $n \in S'$, let p_n be the probability that $\lim_{t\to\infty} \frac{Q^u_n(t)}{t} > 0$. Claim 1 establishes that $\sum_{n\in S'} p_n \geq 1$. Therefore, it follows that there exists a $n \in S'$ such that, with probability at least $1/|S'|$, $\lim_{t\to\infty} \frac{Q^u_n(t)}{t} > 0$, where $|S'|$ denotes the cardinality of S'. For one such n, from a standard probability result that will be presented below as Lemma 1, we have $\lim_{t\to\infty} \frac{\mathbb{E}[Q^u_n(t)]}{t} > 0$.

Lemma 1. *A non-negative random variable X has zero expectation if and only if it equals zero almost surely. If $X > 0$ with positive probability, then $\mathbb{E}[X] > 0$.* [4]

Thus, we have shown that the randomized policy destabilizes user traffic if there exists an U' that satisfies the val-condition. Note that the sufficiency of the theorem does not rely on that the properties of JSQ, and thus holds for any routing policy of user traffic.

Proof of Necessity: Suppose that there exists an injection policy that destabilizes user traffic, then there exists a sample path on which $\lim_{t\to\infty} \frac{Q^u_n(t)}{t} > 0$ for some n. Under one such

[4] We provide a proof for this lemma in Appendix D for completeness.

sample path, let $S' \in S$ be the subset of servers that ended up getting overflowed, i.e., $\forall n \in S'$, $\lim_{t\to\infty} \frac{Q_n(t)}{t} > 0$ and $\forall n \notin S'$, $\lim_{t\to\infty} \frac{Q_n(t)}{t} = 0$. Since the queues in S' grow linearly with time while the queues in $S\backslash S'$ do not, eventually queues in S' will be longer than those in $S\backslash S'$. Therefore, for every user dispatcher u_l that has connection to servers that are not in S' the time average arrival rates from u_l to servers in S' are all zero, by the property of JSQ. Formally, we have

$$\lim_{t\to\infty} \sum_{n\in S'} \sum_{i=0}^{t-1} \frac{a_{ln}^u(i)}{t} = 0, \quad \forall u_l \; s.t. \; S_{u_l} \cap (S\backslash S') \neq \emptyset. \tag{6}$$

We now claim in the following that S' satisfies the *val*-condition. By establishing the claim, we prove the necessity part of the theorem.

Claim 2. *S' satisfies the val-condition.*

Proof of Claim 2 (Sketch): Recall that the set of user dispatchers that only have connections to servers in S' is denoted by $U_{S'}$. We first prove that $U_{S'}$ is not empty. For if not, suppose $U_{S'}$ is empty, then by Equation (6), the time average rates from all user dispatchers to S' are zero, i.e., the user dispatchers inject all their traffic to the non-overloaded servers $S\backslash S'$, and the adversary would not be able to destabilize user traffic. We next demonstrate that $val(S') > \Delta(S')$ by taking the time average traffic rates from adversary dispatchers as a feasible solution to the linear program $LP(S')$ and showing that the feasible solution manifest that $val(S') > \Delta(S')$. See Appendix A.2 for details. □

The necessity of *val*-condition follows directly from Claim 2. □

Based on Theorem 1, we have the following corollary. It states that to check the feasibility of the network DoS problem, we only need to check the subsets of servers induced by subsets of user dispatchers.

Corollary 1. *The network DoS problem is feasible if and only if there exists a non-empty subset of user dispatchers $U' \subseteq U$, such that $S_{U'} = \bigcup_{u_l\in U'} S_{u_l}$ satisfies the val-condition.*

Proof. From the definition of *val* function, we have that if there exists a subset of servers S' that satisfies the *val*-condition, then $U_{S'}$ satisfies the condition in Corollary 1. Conversely, if $U' \subseteq U$ satisfies the condition in the corollary, then $S_{U'} \subseteq S$ satisfies the *val*-condition by design. □

Remark: An adversarial injection policy is *optimal* if the adversary destabilizes user traffic under such policy whenever the network DoS attack problem is feasible, i.e., it achieves the feasibility region. Moreover, it is *oblivious* if the policy does not rely on knowledge of the network statistics $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}^u$ or budget vector $\boldsymbol{\lambda}^v$. The proof of Theorem 1 yields a randomized adversarial injection policy that is optimal. However, the policy is not oblivious since it involves solving *LP*'s which depend on network statistics. Since in practice the adversary often does not have such knowledge, we will develop an optimal oblivious policy in Section 5.

4 PROPERTIES OF JSQ

In this section, we introduce several properties of the JSQ routing policy, which lay the foundation of the adversarial injection policy that we propose in Section 5. We will also consider the JSQ policy where the queue non-negativity is relaxed, i.e., queues evolve under the dynamics $Q_n(t+1) = Q_n(t) - b_n(t) + a_n(t)$. Here, JSQ specifies that the arrivals go to the queue with the smallest value. We will refer to such evolution as the *relaxed dynamics* and the original one with non-negativity constraint as the *original dynamics*. We note that our results on the relaxed dynamics only serve as tools for analyzing the policy we proposed, and the analysis of the policy is carried out on the

system with the original dynamics. The results we establish in this section, including ones that extend results by Shah and Wischik in [22, 23] on overloaded Max-Weight switch networks, and a sample path-wise coupling bound of JSQ, may be of independent interests.

4.1 Queue Length Behavior Under JSQ

Consider a server farm with set of servers $S = \{s_1, \ldots, s_N\}$ and set of traffic dispatchers $U = \{u_1, \ldots, u_L\}$ (There is no adversary dispatcher for now). The traffic dispatchers route incoming traffic to servers following the JSQ rule, with ties broken arbitrarily. Similar to our model, the service rate of each server s_n at every time slot is a random variable, i.i.d across time, with mean μ_n, and the arrival at each dispatcher u_l is a random variable, i.i.d across time, with mean λ_l. The μ_n's determine the throughput region of the network, i.e., the set of λ_l's that are supportable under some routing policy. It is well known that JSQ is throughput-optimal in the sense that using JSQ, the server farm can support the incoming traffic as long as λ_l's lie in the throughput region [10]. It is also intuitively understood that when the arrival rates are outside the throughput region, JSQ achieves graceful degradation such that the queues grow with time in a balanced manner. Proposition 1 makes this intuition precise.

Consider the following optimization problem $\mathcal{P}$.

$$\min \sum_n r_n^2 \tag{7}$$

$$\textbf{s.t.} \sum_{n \in S_{u_l}} \lambda_{ln} = \lambda_l, \qquad \forall l \in U \tag{8}$$

$$r_n \geq \sum_{l: n \in S_{u_l}} \lambda_{ln} - \mu_n, \qquad \forall n \in S \tag{9}$$

$$\lambda_{ln} = 0, \qquad \forall n \notin S_{u_l} \tag{10}$$

$$r_n, \lambda_{ln} \geq 0, \qquad \forall l, n. \tag{11}$$

$\mathcal{P}$ is a convex optimization problem over both $\{r\}_n$ and $\{\lambda\}_{ln}$. Given a server farm, λ_{ln} can be interpreted as traffic rate from dispatcher u_l to server s_n, and r_n can be interpreted as queue growth rate at server n. The optimization problem $\mathcal{P}$ seeks a set of allocation of traffic rates $\{\lambda\}_{ln}$ that routes all the incoming traffic to servers while minimizing the sum of squares of queue growth rates $\{r\}_n$. We will mostly focus on the $\{r\}_n$ components. Hence, we will often use $(r_1, \ldots, r_n)$ to denote a solution to $\mathcal{P}$, and since $\mathcal{P}$ is strictly convex with respect to r_n's, it has a unique optimal solution (over the $\{r\}_n$ components).

Proposition 1. *Let $\boldsymbol{r^*} = (r_1^*, \ldots, r_n^*)$ be the optimal solution to the optimization problem $\mathcal{P}$ associated with the server farm. The queue lengths under JSQ satisfy that for any $\delta > 0$, for all n,*

$$\lim_{t \to \infty} \mathbb{P}\left\{ \left| \frac{Q_n(t)}{t} - r_n^* \right| < \delta \right\} = 1.$$

Proof. As JSQ is a special case of the Max-Weight algorithm on single-hop networks, the proposition follows from the results of two papers by Shah and Wischik [22, 23] and can be found in Appendix B.1. □

Proposition 1 establishes that under JSQ, the growth rate of queue lengths (over time) converges to the optimal solution to $\mathcal{P}$ in probability. It characterizes the queue length behavior under JSQ in both under-load and over-load regimes. In the former case, the solution is the zero vector, which implies that queues do not grow with time; in the latter case, the optimal solution is the "most balanced" overflow rate that is achievable, and the overflow rate under JSQ converges to that.

Further, we consider the optimization problem $\mathcal{P}'$, which is a modified version of $\mathcal{P}$ that removes the non-negativity constraints on r_n and replace constraint (9) with equality:

$$\begin{aligned} \min \sum_n r_n^2 & \\ \textbf{s.t.} \sum_{n\in S_{u_l}} \lambda_{ln} &= \lambda_l, \qquad \forall l \in U \\ r_n &= \sum_{l:n\in S_{u_l}} \lambda_{ln} - \mu_n, \qquad \forall n \in S \\ \lambda_{ln} &= 0, \qquad \forall n \notin S_{u_l} \\ \lambda_{ln} &\geq 0, \qquad \forall l, n. \end{aligned}$$

$\mathcal{P}'$ is also convex and has unique optimal solution. Under the relaxed dynamics, the queue growth behavior under JSQ corresponds to the optimal solution to $\mathcal{P}'$.

Proposition 2. *Let $\tilde{\boldsymbol{r}}^* = (\tilde{r}_1^*, \ldots, \tilde{r}_n^*)$ be the optimal solution to the optimization problem $\mathcal{P}'$. The queue lengths under JSQ with the relaxed dynamics satisfy that for any $\delta > 0$, for all n,*

$$\lim_{t\to\infty} \mathbb{P}\left\{ \left| \frac{Q_n(t)}{t} - \tilde{r}_n^* \right| < \delta \right\} = 1. \tag{12}$$

Proof. The proof is the same as Proposition 1. □

By expanding the limit expression (12), we obtain the following corollary.

Corollary 2. *Under the relaxed dynamics, if $\min_n \tilde{r}_n^* > 0$, then for all $\epsilon > 0$, there exists a $T_\epsilon > 0$ such that for all $t \geq T_\epsilon$,*

$$\mathbb{P}\left\{ \forall n,\ Q_n(t) \geq \frac{\tilde{r}_n^* t}{2} \right\} \geq 1 - \epsilon.$$

The same holds for the original dynamics with $\boldsymbol{r}^*$.

Proof. Take $\delta = \frac{1}{2}\min_n \tilde{r}_n^* > 0$, by Proposition 2, for all n,

$$\lim_{t\to\infty} \mathbb{P}\left\{ \frac{Q_n(t)}{t} \geq \frac{\tilde{r}_n^*}{2} \right\} = 1$$

Hence, fix a $\epsilon > 0$, for each n, there exists a $T_{n,\epsilon}$ such that for all $t \geq T_{n,\epsilon}$,

$$\mathbb{P}\left\{ \frac{Q_n(t)}{t} \geq \frac{\tilde{r}_n^*}{2} \right\} \geq 1 - \frac{\epsilon}{N} \implies \mathbb{P}\left\{ \frac{Q_n(t)}{t} < \frac{\tilde{r}_n^*}{2} \right\} \leq \frac{\epsilon}{N},$$

where we recall that N is the number of servers. Let $T_\epsilon = \max_n T_{n,\epsilon}$. By union bound, we have for all $t \geq T_\epsilon$,

$$\mathbb{P}\left\{ \exists n, \frac{Q_n(t)}{t} < \frac{\tilde{r}_n^*}{2} \right\} \leq \epsilon.$$

Taking the complement, it follows that

$$\mathbb{P}\left\{ \forall n,\ \frac{Q_n(t)}{t} \geq \frac{\tilde{r}_n^*}{2} \right\} \geq 1 - \epsilon.$$

□

The two optimization problem $\mathcal{P}$ and $\mathcal{P}'$ differ only in the non-negativity constraints of r_n's. Our next result shows that their optimal solutions are identical under certain condition. For two vectors $\boldsymbol{r}$ and $\tilde{\boldsymbol{r}}$, we write $\boldsymbol{r} > \tilde{\boldsymbol{r}}$ if $\forall n,\ r_n > \tilde{r}_n$; $\boldsymbol{r} \geq \tilde{\boldsymbol{r}}$ if $\forall n,\ r_n \geq \tilde{r}_n$; $\boldsymbol{r} = \tilde{\boldsymbol{r}}$ if $\forall n,\ r_n = \tilde{r}_n$.

Proposition 3. *Let $\boldsymbol{r}^*$ be the optimal solution to $\mathcal{P}$ and $\tilde{\boldsymbol{r}}^*$ be the optimal solution to $\mathcal{P}'$. For any n, if $r_n^* > \mathbf{0}$, then $r_n^* = \tilde{r}_n^*$.*

Proof. The proof follows from the structure of $\mathcal{P}$ and $\mathcal{P}'$, in particular, that an optimal solution to $\mathcal{P}$ must be feasible to $\mathcal{P}'$ and that Constraints (9) must be binding for any optimal solution to $\mathcal{P}$. We defer the details to Appendix B.2. □

4.2 Monotonicity Property of JSQ

We present a sample path-wise bound regarding the queue length vector of JSQ server farm with the relaxed dynamics. Consider a server farm with servers $\{s_1, \ldots, s_N\}$ and dispatchers $\{u_1, \ldots, u_L\}$, where all dispatchers use JSQ routing. The arrivals and services at each time slot are upper bounded by C. We assume that the queues evolve under the relaxed dynamics (as in Proposition 2), which means that all the realized services equal the offered services and the queue lengths can become negative. Consider a sample path of such system. Observe that if we are given an initial queue length vector $\boldsymbol{Q}(0) = \{Q_1(0), \ldots, Q_n(0)\}$, sequence of arrivals at user dispatchers $(\boldsymbol{\lambda}^u(0), \boldsymbol{\lambda}^u(1), \ldots)$ where each $\boldsymbol{\lambda}^u(t) = (\lambda_1^u(t), \ldots, \lambda_L^u(t))$ specifies the arrivals at time t, sequence of offered services at servers $(\boldsymbol{b}(0), \boldsymbol{b}(1), \ldots)$ where each $\boldsymbol{b}(t) = (b_1(t), \ldots, b_N(t))$ specifies the offered services at time t, and certain tie-breaking rule, then the queue length vector at each future time slot can be fully determined. Therefore, consider two JSQ server farms whose traffic arrivals and services are identical random variables, one has initial queue length vector $\boldsymbol{Q}(0)$ and the other has $\tilde{\boldsymbol{Q}}(0)$. We can couple the (random) queue length vectors at time slot t of the two system in a sample path-wise manner. At each sample path with some common sequences of arrivals $(\boldsymbol{\lambda}^u(0), \boldsymbol{\lambda}^u(1), \ldots)$ and services $(\boldsymbol{b}(0), \boldsymbol{b}(1), \ldots)$, we let $\boldsymbol{Q}(t)$ and $\tilde{\boldsymbol{Q}}(t)$ be the (deterministic) queue length vectors at t starting from $\boldsymbol{Q}(0)$ and $\tilde{\boldsymbol{Q}}(\mathbf{0})$, respectively. Since the arrivals and services of the two systems are identical random variables, the above procedure forms a coupling. In the following proposition, we will establish a sample path-wise relation between the two random queue length vectors under the aforementioned coupling.

Proposition 4. *If $\boldsymbol{Q}(0) \geq \tilde{\boldsymbol{Q}}(0)$, then at each sample path, for all t and n,*

$$Q_n(t) \geq \tilde{Q}_n(t) - N_1 LC,$$

where $N_1 = N! + 1$. The result holds for arbitrary tie-breaking rules that the two systems use.

Proof. We defer rigorous proof to Appendix B.3 and gives some intuition here. Suppose that the system is performing JSQ in a packet-by-packet fashion, i.e., for each packet that arrives at some dispatcher, the dispatcher sends the packet to the server with the shortest queue (among the ones that it is connected to) and the queue lengths are updated immediately afterwards. Then, one can actually show that under certain tie-breaking rule, $Q_n(t) \geq \tilde{Q}_n(t)$ for all n, t. The argument is that such relation is invariant through any packet transmission, provided that the ties are broken appropriately. The additional $N_1 LC$ factor in Proposition 4 accounts for different tie-breaking rules, and that the JSQ in our model is performed in a slot-by-slot, rather than packet-by-packet fashion. □

5 OPTIMAL OBLIVIOUS ADVERSARY INJECTION POLICY

In this section, we design an optimal oblivious adversarial injection policy that destabilizes user traffic whenever it is feasible and does not require knowledge of network statistics. From the proof of Theorem 1, we observe that intuitively an optimal policy should be able to identify a subset of servers that satisfies the *val*-condition and appropriately allocate adversary dispatchers' traffic to that subset of servers without over-expending its budget on any single server (c.f. constraint (9)).

For an optimal policy to be oblivious, it needs to achieve the two aforementioned objectives based solely on queue-length information rather than network statistics. Before introducing such a policy, we first present an intermediate policy that is optimal but semi-oblivious, in the sense that it achieves the second objective without relying on network statistics, i.e., given a subset of servers that satisfies the *val*-condition, the policy that destabilizes user traffic decides the adversarial injection based on queue length information only. The policy, called "Target-JSQ policy", brings out a key idea and paves the way to the optimal oblivious policy.

5.1 Target-JSQ Policy

As its name suggests, the Target-JSQ policy works by identifying a subset of servers that satisfies the *val*-condition, and then making all the adversary dispatchers that have connection to that subset send packets to the servers following JSQ rule. Formally, let S' be a subset that satisfies the *val*-condition. For all the adversary dispatchers v_m such that $S_{v_m} \cap S' \neq \emptyset$, at each time slot, v_m send its packets to the servers in $S_{v_m} \cap S'$ with the shortest queue. Other adversary dispatchers send packets arbitrarily (or do not send packets at all). Theorem 2 establishes the optimality of the Target-JSQ policy.

Theorem 2. *Suppose $S' \subseteq S$ satisfies the val-condition and all the adversary dispatchers that have connections to S' inject traffic (only) to S' according to the JSQ rule, then we have*

$$\exists n \in S', \lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0.$$

Proof. Let $V_{S'}$ be the subset of adversary dispatchers that have connections to S', i.e., $V_{S'} = \{v_m \mid S_{v_m} \cap S' \neq \emptyset\}$, and recall that $U_{S'}$ is defined to be the subset of user dispatchers that only have connection to S', i.e., $U_{S'} = \{u_l \mid S_{u_l} \subseteq S'\}$. Since S' satisfies the *val*-condition, $U_{S'}$ is non-empty. If $V_{S'}$ is empty, then $\Delta(U') < 0$, which means that the total rate of incoming user traffic to S' is greater than the total service rate of S'. Hence, the theorem vacuously holds. Therefore, we can assume that $V_{S'}$ is not empty. For simplicity, we consider the case where the adversary dispatchers in $V_{S'}$ inject packets to S' according to the JSQ rule, and other adversary dispatchers send nothing to the servers. The argument applies to the case where other adversary dispatchers inject arbitrarily as well. Now, we study the system formed by user dispatchers U, adversary dispatchers $V_{S'}$ and servers S, with $V_{S'}$ only have connections to S'. Note that the system is a server farm where all the dispatchers ($U \cup V_{S'}$) employ JSQ routing. Therefore, the queue length growth rates follow Proposition 1. Let $\mathcal{P}$ be the optimization problem in Proposition 1, associated with this system, and $\boldsymbol{r}^*, \boldsymbol{\lambda}^*$ be an optimal solution to $\mathcal{P}$. We will use λ^*_{ln} and λ^*_{mn} to denote the component in $\boldsymbol{\lambda}^*$ that correspond to user dispatcher u_l and adversary dispatcher v_m, respectively.

For the subset of servers S', the total incoming rate of adversary traffic equals $\sum_{m\in V_{S'}} \lambda^v_m$, which by definition, is greater than or equal to $val(S')$. The total incoming rate of user traffic is at least $\sum_{u_l\in U_{S'}} \lambda^u_l$. Since S' satisfies the *val*-condition, we have $\sum_{m\in V_{S'}} \lambda^v_m + \sum_{u_l\in U_{S'}} \lambda^u_l > \sum_{s_n\in S'} \mu_n$. It follows by summing up constraints (9) of $\mathcal{P}$ over all $n \in S'$ that $\boldsymbol{r}^*$ must have at least one positive entry. Let $\tilde{S}'$ be the set of servers in S' whose corresponding entry is positive in $\boldsymbol{r}^*$, i.e., $\tilde{S}' = \{s_n \in S' \mid r^*_n > 0\}$. By the above reasoning, $\tilde{S}' \neq \emptyset$. We will show in the following claim that there must exist user dispatcher u_l such that $S_{u_l} \subseteq \tilde{S}'$, which means that the user traffic from u_l can only go to servers in $\tilde{S}'$.

Claim 3. *There exists user dispatcher u_l such that $S_{u_l} \subseteq \tilde{S}'$.*

Proof of Claim 3 (Sketch): We prove the claim by contradiction. If the claim does not hold, then every user dispatcher has connection to $S\backslash\tilde{S}'$. Combining this with the condition that S' satisfies the

val-condition, we obtain that $\sum_{n\in S\setminus\tilde{S}'} r_n^* > 0$, which contradicts the definition of $\tilde{S}'$. See Appendix A.3 for details. □.

Based on Claim 3, invoking Proposition 1 and Corollary 2, we have that for any $\epsilon > 0$, there exists a T_ϵ such that for all $t \geq T_\epsilon$,

$$\mathbb{P}\left\{ \forall n,\ \frac{Q_n(t)}{t} \geq \frac{r_n^*}{2} \right\} \geq 1-\epsilon.$$

Hence, with probability at least $1-\epsilon$, $\lim_{t\to\infty} \frac{Q_n(t)}{t} > 0$ for all $n \in \tilde{S}'$. Since there exists user dispatcher u_l with $S_{u_l} \in \tilde{S}'$, by Observation 1, we have with probability at least $1-\epsilon$, there exists n with $\lim_{t\to\infty} \frac{Q_n^u(t)}{t} > 0$. Since there are finitely many servers n and $\frac{Q_n^u(t)}{t}$'s are non-negative, by Lemma 1, we have that there exists $n \in \tilde{S}'$ such that $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$, which completes the proof of Theorem 2. □

The proof of Theorem 2 sets the stage for the definition of a *Vulnerable Set*, which will play an important role in the analysis of the optimal oblivious policy we propose. A subset $\tilde{S}$ of servers is a vulnerable set, if (i) $\tilde{S}$ is the (minimal) set of servers that some (non-empty) subset of user dispatchers are connected to, and (ii) if all the adversary dispatchers that are connected to $\tilde{S}$ injects traffic to $\tilde{S}$ following the JSQ rule, then all the queues in $\tilde{S}$ grow with time. We formally present the definition of vulnerable set as follows.

Definition 3 (Vulnerable Set). *A subset of servers $\tilde{S}$ is a vulnerable set, if (i) there exists a non-empty subset U' of user dispatchers such that $\tilde{S} = \bigcup_{u_l\in U'} S_{u_l}$, and (ii) if all the adversary dispatchers that have connection to $\tilde{S}$ send packets to $\tilde{S}$ following the JSQ rule, the system will become a JSQ server farm. Let $\boldsymbol{r}^*$ be the solution to the optimization problem $\mathcal{P}$ that corresponds to that server farm, then $r_n^* > 0$ for all $n \in \tilde{S}$. The collection of all vulnerable subsets is denoted by $\mathcal{S}_0$.*

The following corollary establishes the existence of vulnerable set for feasible network DoS attack problem.

Corollary 3. *If the network DoS problem is feasible, then there exists a vulnerable set.*

Proof. From Theorem 2, Claim 3 and their proofs, it is straightforward to show that S_{u_l} in Claim 3 is a vulnerable set. □

It is easy to see that all vulnerable subsets satisfy the *val*-condition, but the converse is not necessarily true. Let S' be a set of servers that satisfies the *val*-condition. S' itself may not be a vulnerable set, but it contains a vulnerable set as a subset. Recall that $U_{S'}$ denotes the user dispatchers that are only connected to servers in S'. S' satisfying the *val*-condition implies that user traffic from dispatchers in $U_{S'}$ will be blocked from service if the adversary injects to S' following the JSQ rule. On one hand, S' may not satisfy the conditions for vulnerable sets since S' may contain queues that do not grow with time or "redundant" servers in the sense that S' may be a strict super-set of $\bigcup_{u_l\in U_{S'}} S_{u_l}$. On the other hand, let $\tilde{U}_{S'} \subseteq U_{S'}$ be the set of user traffic dispatchers whose traffic will be blocked, then $\bigcup_{u_l\in\tilde{U}_{S'}} S_{u_l}$, can be verified to be a vulnerable set. We further illustrate this the following example.

Example: Consider **Figure 3**, the set S' satisfies the *val*-condition, but it is not a vulnerable set. First, it is a strict super-set of $\bigcup_{u_l\in U_{S'}} S_{u_l}$. Moreover, if the adversary injects to S' following the JSQ rule, not all user traffic from $U_{S'}$ will be blocked but only the one from $\tilde{U}_{S'}$, as the server at the bottom of the figure will not be overloaded. However, S' contains a vulnerable set as subset, which is $\bigcup_{u_l\in\tilde{U}_{S'}} S_{u_l}$ (the inner rectangle marked in red).

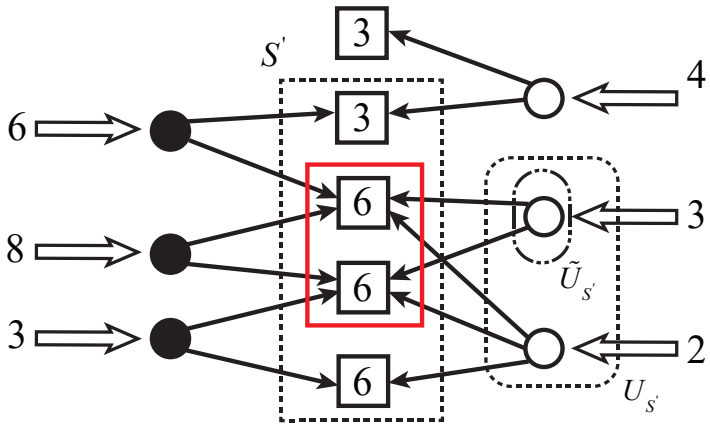

Fig. 3. Illustration of vulnerable set. The set S' satisfies the *val*-condition. It contains a vulnerable set (the inner rectangle marked in red) as subset.

Now, we have discovered an optimal adversarial injection policy that only requires the adversary dispatchers to perform JSQ routing to a vulnerable set of servers. Such JSQ-operation can automatically allocate the adversary's budget to servers in some vulnerable set without "wasting" adversary traffic. To design an optimal oblivious policy, the remaining task is to identify a vulnerable set without relying on network statistics, which is what we will do in the following section.

5.2 The Min-Zero Policy

In this section, we present the optimal oblivious adversarial injection policy – the Min-Zero policy. It uses the idea of the Target-JSQ policy and aims to identify a vulnerable set based (only) on queue-length information.

At each time slot t, the adversary maintains a target subset of user dispatchers and a corresponding target subset of servers, which are denoted by $U(t)$ and $S(t)$, with $U(t) \subseteq U$, $S(t) \subseteq S$ and $S(t) = \bigcup_{u_l \in U(t)} S_{u_l}$. All the adversary dispatchers that have connections to $S(t)$ send packets to $S(t)$ in a JSQ fashion, and other adversary dispatchers send packets arbitrarily. Then, after the servers finished their service during the current slot, the adversary checks if $\min_{n \in S(t)} Q_n(t) = 0$ (hence the name, Min-Zero). If so, then in the next slot, the adversary choose $U(t+1)$ uniformly at random from all non-empty subsets of user dispatchers and set $S(t+1)$ accordingly; otherwise, set $U(t+1) := U(t)$ and $S(t+1) := S(t)$. We formally present the Min-Zero policy in **Algorithm 1**.

Obviously, the Min-Zero policy is oblivious. We will next establish its optimality in Theorem 3. Before doing that, we present a key lemma. The lemma is adapted from Theorem 2.1.10 and Theorem 2.2.7 in [24], which state a sufficient condition for transience of countable-state Markov Chains.

Lemma 2. *Let $\mathcal{L}$ be an irreducible countable-state discrete-time Markov Chain with state space $\mathcal{A}$. Let $X(t)$ denote the state of the chain at time t. The chain $\mathcal{L}$ is transient, if there exist a non-negative function (Lyapunov Function) $f(\alpha), \alpha \in \mathcal{A}$, a positive integer k, and $\delta > 0, \gamma > 0$, such that, setting $A_\gamma = \{\alpha : f(\alpha) > \gamma\} \neq \emptyset$, the following conditions hold:*

1. *Let $D(X(t)) = f(X(t+k)) - f(X(t))$. There exists $b > 0$ such that $\mathbb{E}[D(X(t)) \cdot \mathbb{1}\{D(X(t)) < b\} \mid X(t) = \alpha_i] \geq \epsilon$, for all t, and $\alpha_i \in A_\gamma$, where $\mathbb{1}\{\cdot\}$ denotes the indicator function.*
2. *There exists $d > 0$ such that $f(X(t+1)) - f(X(t)) < -d$ almost surely, for all t.*

Algorithm 1 The Min-Zero Policy

Input: Server set $S = \{s_1, \ldots, s_n\}$, user dispatcher set $U = \{u_1, \ldots, u_l\}$, adversary dispatcher set $V = \{v_1, \ldots, v_m\}$

Initialize: $U(0) :=$ a random non-empty subset of U
$S(0) := \bigcup_{u_l \in U(0)} S_{u_l}$.
for $t = 0, 1, 2, \ldots$ **do**
  Each adversary dispatchers v_m such that $S_{v_m} \cap S(t) \neq \emptyset$ sends packets to $S_{v_m} \cap S(t)$ following the JSQ rule.
  All other adversary dispatchers send packets arbitrarily.
  After the service phase of current time slot t
  if $\min_{n \in S(t)} Q_n(t) = 0$ **then**
    $U(t+1) :=$ nonempty subset of U chosen uniformly at random.
    $S(t+1) := \bigcup_{u_l \in U(t+1)} S_{u_l}$
  else
    $U(t+1) := U(t), S(t+1) := S(t)$.

The lemma can be interpreted as a converse of the Foster-Lyapunov theorem. It states that, if we can find a Lyapunov function on the state space such that in conditioning on a subset of states, the Lyapunov function has positive k-slot (truncated) drift, then the Markov Chain is transient.

Now, we are ready to state and prove the optimality of the Min-Zero policy.

Theorem 3. *Under the Min-Zero policy, there exists a queue n with $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$ if the network DoS attack problem is feasible.*

Proof. First, by the execution of the Min-Zero Policy, it is straightforward to see that the evolution of the system follows an irreducible discrete-time countable-state Markov Chain with state $(\boldsymbol{Q}(t), S(t))$. Formally, let $\mathcal{S} = \{S' \mid S' = \bigcup_{u_l \in U'} S_{u_l} \text{ for some } U' \subseteq U\}$, i.e., the collection of all possible target subsets of servers that the Min-Zero might choose. The state space of the Markov Chain can be written as $\mathbb{N}^N \times \mathcal{S}$, the product set of N-dimensional vectors over natural numbers and $\mathcal{S}$. We will prove the theorem by invoking Lemma 2 with a suitably constructed Lyapunov function.

The Lyapunov function we define on the state space is

$$f(\mathbf{Q}(t), S(t)) = \mathbb{1}\{S(t) \in \mathcal{S}_0, \min_{n \in S(t)} \{Q_n(t)\} > TC_1\} \cdot \left(\min_{n \in S(t)} \{Q_n(t)\} - TC_1 \right),$$

with T being a large constant[5] that will be specified later and $C_1 = (M+L)C$ being an upper bound on the sum of arrivals/service that a queue can receive. Due to the boundedness of arrivals and services, we can first easily verify that the Markov Chain satisfies the second condition in Lemma 2. We proceed to show that it also satisfies the first condition, i.e., it has positive expected drift under certain set. Specifically, we will prove that for some $b > 0$

$$\mathbb{E}\left[D(\boldsymbol{Q}(t), S(t)) \mathbb{1} \cdot \{D(\boldsymbol{Q}(t), S(t)) < b\} \mid f(\boldsymbol{Q}(t), S(t)) > 0\right] \geq \delta,$$

for all t and suitable $\delta > 0$. Note that since f only takes integer value, the above construction corresponds to setting $k = T$, $\gamma = 1/2$ (γ can be any positive number less than 1) and A_γ accordingly in Lemma 2.

[5] Our choice of T will depend on the network size and network statistics, but the Min-Zero policy does not.

Before presenting the technical proof, we first give some intuition. Notice that the construction of the Lyapunov function f ensures that it takes positive value only in the states where the adversary is targeting some vulnerable subset (c.f. Definition 3), and will not change its target in the following T time slots. Conditioning on such states, the T-slot drift is roughly equal to the drift of the minimum queue length in the target subset $S(t)$. Since $S(t)$ is vulnerable, we can take T to be suitably large and invoke Proposition 1 to show that the queue lengths in $S(t)$ all grow at a positive rate over the T time slots. This suggests that the Markov Chain satisfies the first condition in Lemma 2.

To make the intuition concrete, we first lower bound the drift. Conditioning on any t such that $f(\boldsymbol{Q}(t), S(t)) > 0$, without loss of generality and for notational convenience, we assume $t = 0$. As $f(\boldsymbol{Q}(0), S(0)) > 0$, we have $\min_{n\in S(0)} Q_n(0) > TC_1$. It follows that the adversary's target will not change over the next T time slots since the queues in $S(0)$ will remain positive. Therefore, $S(T) = S(0) \in \mathcal{S}_0$. Moreover, the value of f can increase by at most TC_1 over the next T slots due to the boundedness of arrivals. Thus, taking $b = TC_1 + 1$, we have

$$\begin{aligned}&\mathbb{E}\left[D(\boldsymbol{Q}(0), S(0))\mathbb{1}\cdot\{D(\boldsymbol{Q}(0), S(0)) < b\} \mid f(\boldsymbol{Q}(0), S(0)) > 0\right]\\ =&\mathbb{E}\left[f((Q(T), S(T)) - f((Q(0), S(0)) \mid f(Q(0), S(0)) > 0\right].\end{aligned} \tag{13}$$

Hence, we can directly work with (13). Claim 4 shows that we can lower-bound (13) by (14), which will be justified in detail in Appendix A.4.

Claim 4.

$$\begin{aligned}&\mathbb{E}\left[f((Q(T), S(T)) - f((Q(0), S(0)) \mid f(Q(0), S(0)) > 0\right]\\ \geq&\mathbb{E}\left[\min_{n\in S(0)}\{Q_n(T)\} - \min_{n\in S(0)}\{Q_n(0)\} \mid f(Q(0), S(0)) > 0\right].\end{aligned} \tag{14}$$

We proceed to establish that the RHS of (14) is positive. Let $Q^*(0) = \min_{n\in S(0)} Q_n(0)$ and $\tilde{\boldsymbol{Q}}(0) = (Q^*(0), \ldots, Q^*(0))$. Based on this, we define random vector $\tilde{\boldsymbol{Q}}(t)$ as the resulting queue length vector at t under the Min-Zero policy starting from state $(\tilde{\boldsymbol{Q}}(0), S(0))$ at 0. We couple $\boldsymbol{Q}(t)$ and $\tilde{\boldsymbol{Q}}(t)$ in the same probability space by equaling their corresponding sequences of arrivals (to the dispatchers) and services on each sample path. We further define $\boldsymbol{Q}^o(t)$ as $\tilde{\boldsymbol{Q}}(t) - \tilde{\boldsymbol{Q}}(0)$. As $\tilde{\boldsymbol{Q}}(t)$ never goes negative in the interval $[0, T]$, it is clear that $\boldsymbol{Q}^o(t)$ has the same distribution as a random vector that is the result of performing the same sequence of routing actions (as in $\tilde{\boldsymbol{Q}}(t)$) starting from a all-zero queue length vector under the relaxed dynamics for $0 \leq t \leq T$ (i.e., all the realized services equal the offered services and queue lengths can be negative). By Proposition 4, we have that on each sample path, $\boldsymbol{Q}_n(t) \geq \tilde{\boldsymbol{Q}}_n(t) - N_1C_1$ for all n and $t \leq T$. It follows that $Q_n(t) \geq Q^o_n(t) + Q^*(0) - N_1C_1$ for all n and $t \leq T$.

Observe that for $0 \leq t \leq T$, $\boldsymbol{Q}^o(t)$ evolves as the queue length vector of a server farm employing the JSQ routing under the relaxed dynamics. Consider the sub-server farm formed by servers in $S(0)$, user dispatchers and adversary dispatchers that have connection to $S(0)$. Since $S(0) \in \mathcal{S}_0$, i.e., it is a vulnerable set, by Corollary 3, the optimal solution $\boldsymbol{r}^*$ of its corresponding optimization problem $\mathcal{P}$ satisfies $\boldsymbol{r}^* > \boldsymbol{0}$. Then, by Proposition 3, the optimal solution $\tilde{\boldsymbol{r}}^*$ to its corresponding optimization problem $\mathcal{P}'$ under the relaxed dynamics satisfies $\tilde{\boldsymbol{r}}^* = \boldsymbol{r}^* > \boldsymbol{0}$. Set $\epsilon = \epsilon_{S(0)}$ as $\frac{\min_{n\in S(0)} \tilde{r}^*_n}{2\min_{n\in S(0)} \tilde{r}^*_n + 4C_1} > 0$. We invoke Corollary 2, and obtain that if $T > T_{\epsilon_{S(0)}}$ (defined in Corollary 2), for all t with $T_{\epsilon_{S(0)}} \leq t \leq T$,

$$\mathbb{P}\left\{\forall n \in S(0), Q^o_n(t) \geq \frac{\tilde{r}^*_n t}{2}\right\} \geq 1 - \epsilon_{S(0)}.$$

Now, we set T as $\max_{S(0)\in\mathcal{S}_0} \max\{T_{\epsilon_{S(0)}}, \frac{8N_1C_1}{\min_{n\in S(0)} \tilde{r}_n^*}\}$ and obtain the following claim. The proof of Claim 5 is presented in Appendix A.5.

Claim 5.

$$\mathbb{E}\left[\min_{n\in S(0)}\{Q_n(T)\} - \min_{n\in S(0)}\{Q_n(0)\} \mid f(Q(0), S(0)) > 0\right] \geq N_1C_1,$$

and there exists n with $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$.

Claim 5 establishes that the Min-Zero policy destabilizes user traffic and concludes the proof of Theorem 3. □

Remark: (i) *Switching Threshold:* in **Algorithm 1** (line 5), we set the switching threshold as 0. It is clear from the proof that the policy remains optimal under any other positive constant switching threshold. Intuitively, the convergence time of the algorithm, i.e., the time it takes the adversary to identify a vulnerable set and not switch the target further, depends on the threshold. Too high a threshold would force the adversary to switch prematurely while too low a threshold would make the adversary switch too infrequently. (ii) *Distributed Implementation:* **Algorithm 1** describes the Min-Zero policy in a centralized fashion, but the policy can be easily implemented in a distributed way. In the distributed implementation, each adversary dispatcher maintains its own target subset of servers, sends traffic to the target set according to the JSQ rule, and switch if the minimum queue length in the target subset hits zero. The optimality of the distributed implementation can be shown using a similar method. This feature also makes the Min-Zero policy more attractive in practice. We will explore the aforementioned two aspects of the policy in the simulations.

Comparison with [21]: Paschos and Tassiulas proposed an alternative (oblivious) adversarial injection policy in [21], Join-the-Longest-Legitimate-Queue (JLLQ), where each adversary dispatcher sends packets to the servers whose queue contains the maximum number of user packets. When the user dispatchers use JSQ routing, they conjecture that JLLQ is optimal. We would like to point out that we have tried but not been able to prove the optimality of JLLQ. Further, our Min-Zero policy shares the simplicity and obliviousness of JLLQ, and has an additional advantage that Min-Zero does not require knowledge of the number user packets in the queues.

5.3 Practical Implications

We provide a discussion on the practical implications of our results from both the DoS attack and defense points of view.

The Min-Zero policy possesses several nice properties that render it a suitable DoS attack strategy in practice. Its simplicity, obliviousness, and amenity to distributed implementation makes it easy to deploy. Its optimality implies the cost-effectiveness of the attack, which is particularly important to resource-constrained adversaries. Finally, since adversary dispatchers also perform JSQ similarly as user dispatchers, the Min-Zero policy may enable the adversary to evade traditional statistical DoS attack detection [5].

The feasibility region and the optimal adversarial injection policy are based on the modeling assumptions that the user dispatchers perform JSQ routing and servers have infinite buffers. However, our results can still provide insights for practical networked systems that do not employ JSQ routing. As mentioned in the proof of Theorem 1, the sufficiency of the feasibility region does not rely on the properties of JSQ and holds for all user routing policy. This, combined with the necessity of the feasibility region when the network employs JSQ routing, suggests that JSQ is in a sense, the optimal defense strategy against DoS attack. Therefore, the Min-Zero policy and the JSQ routing can be intuitively interpreted as an equilibrium point of DoS attack and defense. Moreover, since

many load-balancing strategies proposed in the literature [25, 26] have similar characteristics as JSQ, the Min-Zero policy may be effective in a wide range of systems. Moreover, the assumption of infinite buffer is a standard one for mathematical tractability in queueing and stochastic network control literature [7, 10]. In practical systems with finite buffers, the criteria of queue instability will translate to buffer overflows. Although the analysis would break in finite-buffer systems, our results, as with other similar results in the literature, still have practical relevance as manifested in [27].

As we mentioned above, traditional statistical DoS detection method might not be effective against the Min-Zero policy due to the similarity of the behaviors of user and adversary dispatchers. It thus calls for more sophisticated detection methods that take into account the network topology or apply server-throttling approaches [28].

6 SIMULATIONS

In this section, we evaluate the Min-Zero policy through simulations. We focus on studying how the parameters of the network and the policy influence the performance of the policy. Specifically, on the network side, we investigate the effects that network size and network load have on the convergence time of the Min-Zero policy, respectively; on the policy side, we investigate the impact of switching threshold and distributed implementation on the convergence time. In the following, we will first introduce the simulation environment, and then present the simulation results.

6.1 Simulation Setting

Network Structure: We use two sets of network structures. In the first set, for each N (number of servers) in $\{100, 150, 200, \ldots, 500\}$, we generate 20 networks with N servers, $\lfloor N/4 \rfloor$ user dispatchers and $\lceil N/10 \rceil$ adversary dispatchers. Each user dispatcher is connected to $\lceil N/5 \rceil$ servers, selected uniformly at random. The service rate at each server is a binomial random variable $B(\mu, 1/2)$ with μ uniformly sampled from $\{20, \ldots, 50\}$. The arrival rate at each user dispatcher is a binomial random variable $B(\mu, 1/2)$ with μ uniformly sampled from $\{20, \ldots, 50\}$. Each adversary dispatcher has connections to $\lfloor N/5 \rfloor$ servers and injection rate as a binomial random variable. The connections and the mean injection rate (budget) are randomly assigned such that the overall network load ρ (total arrival/total service) of the network equals 0.75.[6] In the second set, for each network load ρ in $\{0.75, 0.80, 0.85, 0.90, 0.95\}$, we generate 20 networks with 200 servers, 50 user dispatchers and 20 adversary dispatchers. The service rates and user traffic arrival rates are generated similarly to the first set, while the connections and budgets of adversary dispatchers are randomly generated such that the network load equals ρ in expectation.

Variants of Min-Zero policy: In the simulation, we run the Min-Zero policy with switching threshold (0, 5, 10, 20, 50, 100, 200) to evaluate its performance dependence on the threshold. For ease of presentation, we only show the results of thresholds 0, 10, 50 and 200 in the figures and summarize the complete results in **Tables 2 and 3** of Appendix F. We also run the distributed version of Min-Zero with threshold 0 (described in the final remark of Section 5) to compare the performance of centralized and distributed implementations.

Performance Metric: We use the convergence time of the policy as our performance metric. Theoretically, the convergence time of the policy is defined as the time when the adversary dispatchers identify a vulnerable subset and never switch after that. For ease of computation, in centralized Min-Zero policies, we calculate the convergence time as the time slot that the last "switch" happens, and in the distributed version, we calculate it as the first time slot such that the number of

[6] The procedure does not guarantee that the generated instances are feasible. But we found that all the instances we generated in the simulations were indeed feasible.

user packets in a queue exceeds 1000. By examining the queue length trajectories in our simulations, we have confirmed that the calculated convergence times in all the instances match the theoretical definition. Note that the results presented in each setting are averaged over 20 network instances.

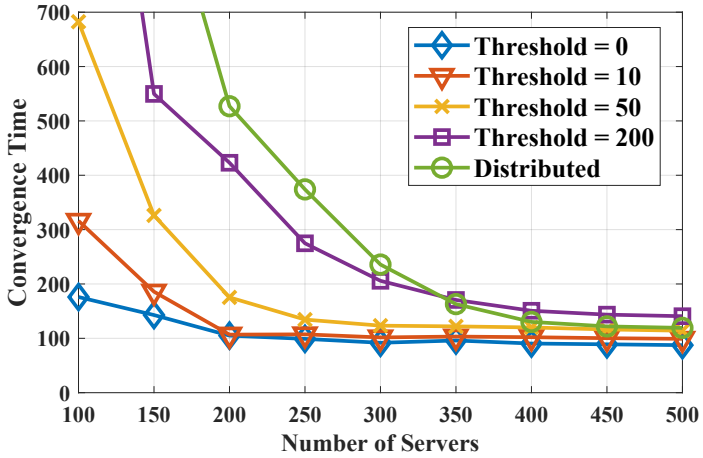

Fig. 4. Convergence times of variants of Min-Zero on networks with different sizes.

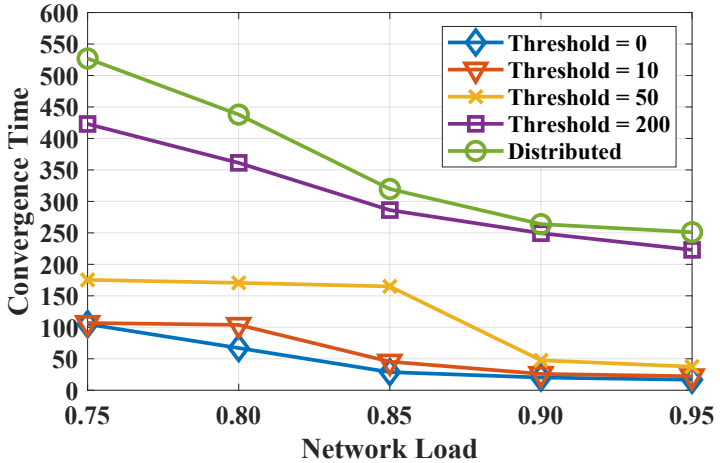

Fig. 5. Convergence times of variants of Min-Zero on networks with different loads.

6.2 Simulation Results

6.2.1 Network Size. We plot the results on the first set of data (varying network sizes) in **Figure 4**. A somewhat counter-intuitive observation is that the convergence times of all variants of Min-Zero decrease with the size of the network. One possible explanation is that among the networks we generated, the larger ones may have a larger portion of vulnerable subsets, making identifying a vulnerable subset easier and thus speeding up the convergence as the Min-Zero policy identifies a

vulnerable set through trial and error. Another interesting finding is that, the higher the threshold, the more sensitive the policy is to the network size. This can be attributed to that in larger networks, the variance of arrival to servers is larger. Such variance directly affect the switching frequency of Min-Zero variants with larger threshold, which translates to influence on the convergence time.

6.2.2 Network Load. We plot the results on the second set of data (varying network loads) in **Figure 5**. We observe that the policies converge faster on networks with larger load. This is not surprising as increasing the load generally increases the number of vulnerable subsets, which in turn speeds up the convergence.

6.2.3 Variants of Min-Zero Policy. From **Figures 4 and 5**, we see that among the centralized policies, the convergence speed decreases as the switching threshold increases. This suggest that simply setting the threshold to zero may be the most desirable in practice. Furthermore, the distributed version of Min-Zero converges slower than its centralized counterpart, but the gap becomes small in large networks.

7 GENERALIZATION TO MULTI-HOP NETWORKS

In this section, we extend our results to multi-hop networks. We first give the model and problem formulation in the multi-hop setting, and then present the counterparts of our previously obtained results in this setting. Since many of the proofs are similar to their single-hop counterparts and the notations are considerably heavier in the multi-hop case, we will only give proof sketches and focus on the differences from the proofs for single-hop results in Appendix C.

7.1 Model and Problem Formulation

Consider a network represented as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with $\mathcal{N}$ denoting the set of nodes and $\mathcal{E} \in \mathcal{N} \times \mathcal{N}$ denoting the set of links. For each node $n \in \mathcal{N}$, let $Out(n)$ be its outgoing neighbors, i.e., $Out(n) = \{n' \in \mathcal{N}, (n, n') \in \mathcal{E}\}$. We assume that the network users send traffic from a single source $s \in \mathcal{N}$ to a single destination $d \in \mathcal{N}$.[7] To avoid unnecessary complexity, we require that there exists a path in $\mathcal{G}$ from each n to d. Additionally, there is a set of adversarial source nodes $\{v_1, \ldots, v_M\}$ such that each node v_m in the set has connection to a subset $\mathcal{N}_{v_m} \subseteq \mathcal{N}$ of network nodes. Note that in our model, there are links from adversarial source nodes to network nodes, but no links in the opposite direction.

The system evolves in discrete time. Each network node $n \in \mathcal{N}$ is associated with a queue, with $Q_n(t)$ denoting the queue length at time t. For each link $e = (n, n') \in \mathcal{E}$, we use $b_e(t)$ or $b_{nn'}(t)$ to represent the offered transmission on the link at t, i.e., at most $b_e(t)$ packets can be sent through e. At each time slot, $\lambda_s(t)$ user packets arrive at source node s, and $\lambda_m^v(t)$ packets arrive at adversary source v_m, for $v_m \in \{v_1, \ldots, v_M\}$. Similar to the single-hop case, we assume that $\{b_e(t)\}$, $\{\lambda_s(t)\}$ and $\{\lambda_m^v(t)\}$ to be independent sequences of i.i.d. random variables with bounded support with $\mathbb{E}[b_e(t)] = c_e, \mathbb{E}[\lambda_s(t)] = \lambda_s, \mathbb{E}[\lambda_m^v(t)] = \lambda_m^v$. Here, we can consider c_e's as the capacity of the links, λ_s as the user traffic rate, and $\boldsymbol{\lambda}^v = (\lambda_1^v, \ldots, \lambda_M^v)$ as the adversary budget vector.

The network nodes (including s) employ a local implementation of the back-pressure routing policy [18, 29], which can also be considered as each node sending packets to its outgoing neighbors following the JSQ rule. Specifically, at time t, node n sends each of its packet to its outgoing neighbor n' that satisfies: n' has the minimum queue length in $Out(n)$, $Q_n(t) > Q_{n'}(t)$, and the offered transmission of the link $b_{nn'}(t)$ has not been depleted (n holds the packets in its queue if no such n' exists). The destination node d instantly absorbs all the packets it receives, and thus $Q_d(t) = 0$ for all t. Each adversarial source node injects traffic to the network nodes it has connection to

[7] Extension to multiple sources is straightforward by creating super-source nodes.

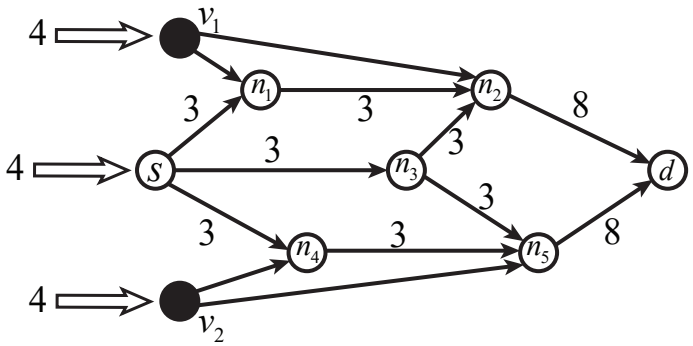

Fig. 6. Illustration of the multi-hop network model. v_1 and v_2 are the adversary sources. The capacities are labeled beside the links. In this example, the adversary can destabilize user traffic by v_1 injecting all of its traffic to n_1 and v_2 injecting all of its traffic to n_4.

following a certain adversarial policy. Without loss of generality, we assume that the capacities of links from adversarial source nodes to network nodes are unbounded so that the amount of traffic that adversarial source nodes can inject is only constrained by their budget. Once sent to network nodes, the adversarial traffic will be merged with user traffic and delivered to d through the network nodes. Define $a^v_{mn}(t)$ as the packets injected from adversarial source v_m to node n with $\sum_{n\in\mathcal{N}_{v_m}} a^v_{mn}(t) = \lambda^v_m(t)$, and $\tilde{b}_{nn'}(t)$ as the packets sent from n to n' at t. We can write the queue length evolution as:

$$Q_n(t+1) = [Q_n(t) + \sum_{m:n\in\mathcal{N}_{v_m}} a^v_{mn}(t) + \mathbb{1}_{n=s}\lambda_s(t) + \sum_{n':n\in Out(n')} b_{n'n}(t) - \sum_{n'\in Out(n)} b_{nn'}(t)]^+.$$

Similar to the single-hop setting, we break down each Q_n into user traffic component, Q^u_n, and adversary traffic component, Q^v_n. Here, we omit the evolution of Q^u_n and Q^v_m for ease of readability.

Based on the above preliminaries, we define the multihop version of network DoS attack problem.

Definition 4 (Multi-hop Network DoS Attack Problem). *The Multi-hop Network DoS Attack problem seeks an adversarial injection policy under which there exists some network node n with* $\lim_{t\to\infty} \frac{\mathbb{E}[Q^u_n(t)]}{t} > 0$.

We give an example of the multi-hop network DoS attack problem in **Figure 6**.

7.2 Feasibility Region

We proceed to present a necessary and sufficient condition for the Multi-hop Network DoS Attack problem to be feasible. It is based on a generalization of the *val*-condition. The key to generalizing the *val*-condition to multi-hop networks is to find the counterpart of "subset of servers" in the multi-hop setting and define the functions $\Delta(\cdot)$ and $val(\cdot)$ accordingly.

Let an *s-d* cut of the network be a partition $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ of network nodes such that $s \in \mathcal{S}$ and $d \in \mathcal{N}\backslash\mathcal{S}$, and the capacity of cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ be given by $Cap(\mathcal{S}) = \sum_{n\in\mathcal{S}, n'\in\mathcal{N}\backslash\mathcal{S}, (n,n')\in\mathcal{E}} c_{nn'}$. The notion of cut will serve as the multi-hop counterpart of "subset of servers". For each *s-d* cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$, we further define $\Delta(\mathcal{S}) := Cap(\mathcal{S}) - \lambda_s$. We next generalize the *val* function to *s-d* cuts. Intuitively, for an *s-d* cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$, we want $val(\mathcal{S})$ to capture the maximum amount

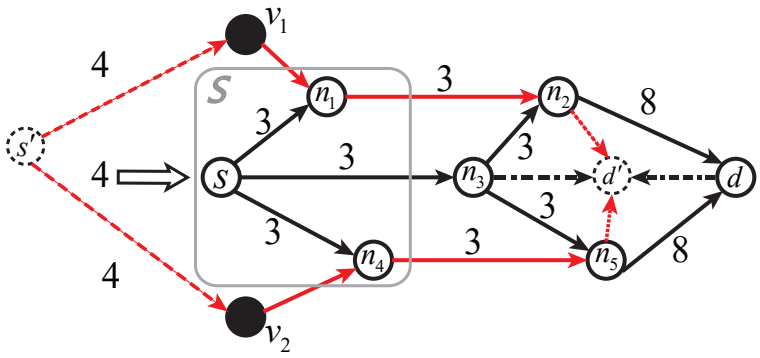

Fig. 7. Illustration of the extended val-condition: Consider the s-d cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ of the network in Figure 6 with $\mathcal{S} = \{s, n_1, n_4\}$. The constructed auxiliary flow network $\mathcal{G}_{\mathcal{S}}$ is shown in the figure. $\lambda_s = 4, Cap(\mathcal{S}) = 9$ and $val(\mathcal{S}) = 6$ (see the maximum s'-d' flow is marked in red.). Therefore, the cut satisfies the extended val-condition.

of traffic that the adversarial source nodes can send through $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$. Formally, for $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$, let $V_{\mathcal{S}} = \{v_m \mid \mathcal{N}_{v_m} \cap \mathcal{S} \neq \emptyset\}$ be the set of adversary source nodes that are connected to $\mathcal{S}$. To define $val(\mathcal{S})$, we construct an auxiliary flow network $\mathcal{G}_{\mathcal{S}}(\mathcal{N}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$ for each $\mathcal{S}$. The node set $\mathcal{N}_{\mathcal{S}} = \mathcal{N} \cup V_{\mathcal{S}} \cup \{s', d'\}$ where s' is a pseudo-source node and d' is a pseudo-destination node; The link set $\mathcal{E}_{\mathcal{S}}$ consists of the links in $\mathcal{E}$, the links from $V_{\mathcal{S}}$ to $\mathcal{S}$, one link from the pseudo-source s' to each node in $V_{\mathcal{S}}$ and one link from each node in $\mathcal{N}\backslash\mathcal{S}$ to the pseudo-destination d'. The capacities of links in $\mathcal{E}_{\mathcal{S}}$ are defined as follows. For link (n, n') that corresponds to an original link in $\mathcal{E}$, i.e. $(n, n') \in \mathcal{E}$, its capacity is equal to $c_{nn'}$. The capacities of links from the pseudo-source node to adversary source nodes are equal to the budgets of the adversary sources, i.e., the capacity of (s', v_m) is equal to λ_m^v. The capacities of other links from $\mathcal{N}\backslash\mathcal{S}$ to d' are infinity. Based on such capacitated flow network $\mathcal{G}_{\mathcal{S}}$, we define $val(\mathcal{S})$ as the value of the maximum flow from the pseudo-source to the pseudo-destination.

$$val(\mathcal{S}) := \text{the value of the maximum } s'\text{-}d' \text{ flow in } \mathcal{G}_{\mathcal{S}},$$

where the definition of the maximum s'-d' flow is standard [30]. An illustration of the extended val-condition is given in **Figure 7**.

Now, we are ready to define the extended val-condition, which leads to establishing the feasibility region of the multi-hop network DoS attack problem.

Definition 5 (Extended val-condition). *An s-d cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ satisfies the extended val-condition if $val(\mathcal{S}) > \Delta(\mathcal{S})$.*

Theorem 4. *The Multi-hop Network DoS Attack problem is feasible if and only if there exists an s-d cut that satisfies the extended val-condition.*

7.3 The Multi-hop Min-Zero Policy

In this section, we introduce the Multi-hop Min-Zero Policy, which is an optimal oblivious adversarial injection policy for multi-hop networks. Similar to its single-hop counterpart, the Multi-hop Min-Zero policy maintains a target cut at each time slot. Let $(\mathcal{S}(t), \mathcal{N}\backslash\mathcal{S}(t))$ be the target cut at t. All the adversary source nodes that are connected to nodes in $\mathcal{S}(t)$ send packets to those nodes following the JSQ rule. If the minimum queue length of nodes $\mathcal{S}(t)$ reaches 0 at t, then the adversary chooses the target at the next time slot uniformly at random from all s-d cuts, otherwise the

adversary targets the same cut at the next time slot. The Multi-hop Min-Zero policy is formally presented in Algorithm 2. We establish the optimality of the Multi-hop Min-Zero policy in Theorem 5.

THEOREM 5. *Under the Multi-hop Min-Zero policy, there exists a network node n with* $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$ *if the multi-hop network DoS attack problem is feasible.*

Algorithm 2 The Multi-hop Min-Zero Policy

Input: Multi-hop Network $\mathcal{G}(\mathcal{N}, \mathcal{E})$, adversary source nodes $V = \{v_1, \ldots, v_m\}$

Initialize: $(\mathcal{S}(0), \mathcal{N}\backslash\mathcal{S}(0)) :=$ a random s-d cut of $\mathcal{G}$.
for $t = 0, 1, 2, \ldots$ **do**
  All adversary source nodes v_m such that $\mathcal{N}_{v_m} \cap \mathcal{S}(t) \neq \emptyset$ send packets to $\mathcal{N}_{v_m} \cap \mathcal{S}(t)$ following the JSQ rule.
  All other adversary dispatchers send packets arbitrarily.
  After the transmission of current time slot t:
  if $\min_{n\in\mathcal{S}(t)} Q_n(t) = 0$ **then**
    $(\mathcal{S}(t+1), \mathcal{N}\backslash\mathcal{S}(t+1)) :=$ an s-d cut chosen uniformly at random.
  else
    $\mathcal{S}(t+1) := \mathcal{S}(t)$.

8 CONCLUSION

In this paper, we made a first attempt towards understanding the fundamental limits of volume-based network DoS attack. We characterized the feasibility region of the attack and proposed the Min-Zero attack policy. The Min-Zero policy is optimal, and oblivious to network statistics, which make it relevant to practical DoS attack in networked systems such as server farms and sensor networks.

From a theoretical point of view, this paper proposed a general recipe for a class of dual problems of stochastic network stability, i.e., how to optimally de-stabilize a network. The first step is to define an appropriate notion of "bottleneck" of the network. The second is to design an oblivious policy that identifies a bottleneck through trial and error, the key to which is to intelligently utilize queue lengths as an indicator of success. An important future work is to extend the recipe to multi-commodity networks, where the main challenge lies in coming up with a suitable notion of bottleneck, analogous to subset of servers in server farm or cut in single-commodity multi-hop networks.

ACKNOWLEDGMENTS

This work was supported by DTRA grants HDTRA1-13-1-0021 and HDTRA1-14-1-0058, and NSF grants AST-1547331, CNS-1617091, CNS-1524317, and CNS-1907905.

REFERENCES

[1] https://www.msspalert.com/cybersecurity-research/kaspersky-lab-study-average-cost-of-enterprise-ddos-attack-totals-2m/
[2] https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/
[3] S. T. Zargar, J. Joshi and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks", in *IEEE communications surveys & tutorials*, Vol. 15, No. 4, pp. 2046-2069, 2013
[4] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and other botnets", in *Computer*, Vol. 50, No. 7, pp. 80-84, 2017.

[5] R. Braga, E. de Souza Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow", in *IEEE LCN*, Vol. 10 pp. 408-415, 2010.
[6] A. Compagno, M. Conti, P. Gasti and G. Tsudik, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking", in *IEEE LCN*, pp. 630-638, 2013.
[7] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan and D. P. Williamson, "Adversarial queuing theory" in *Journal of the ACM*, Vol. 48, No. 1, pp. 13-38, 2001.
[8] D. Gamarnik, "Stability of adaptive and nonadaptive packet routing policies in adversarial queueing networks" in *SIAM Journal on Computing*, Vol. 32, No. 2, pp. 371-385, 2003.
[9] A. Goel, "Stability of networks and protocols in the adversarial queueing model for packet routing", in *Networks: An International Journal*, Vol. 37, No. 4, pp.219-224, 2001.
[10] M.J. Neely, "Stochastic network optimization with application to communication and queueing systems", in *Synthesis Lectures on Communication Networks*, Vol. 3, No. 1, pp. 1-211, 2010.
[11] V. Gupta, M. H. Balter, K. Sigman and W. Whitt, "Analysis of join-the-shortest-queue routing for web server farms", in *Performance Evaluation*, Vol. 64, No. 9-12, pp. 1062-1081, 2007.
[12] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus and A. Greenberg, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", in *Performance Evaluation*, Vol. 68, no. 11, pp. 1056-1071, 2011.
[13] R. K. Wood, "Deterministic network interdiction. Mathematical and Computer Modelling", Vol. 17, No. 2, pp. 1-18, 1993
[14] C. A. Phillips, "The network inhibition problem", in *Proc. of ACM STOC*, pp. 776-785, 1993.
[15] X. Fu and E. Modiano, "Network Interdiction Using Adversarial Traffic Flows", in *IEEE INFOCOM*, pp. 1765-1773, 2019.
[16] S. Wang and N. Shroff, "Security game with non-additive utilities and multiple attacker resources", in *Proc. of the ACM on Measurement and Analysis of Computing Systems*, Vol. 1, No. 1, pp.13, 2017
[17] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Başar and J-P Hubaux, "Game theory meets network security and privacy", in *ACM Computing Surveys*, Vol. 45, No. 3, pp. 25, 2013.
[18] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", in *IEEE Conference on Decision and Control*, pp. 2130-2132, 1990.
[19] Q. Liang and Modiano, "Network utility maximization in adversarial environments", in *IEEE INFOCOM*, pp. 594-602, 2018.
[20] Q. Liang and E. Modiano, "Minimizing Queue Length Regret Under Adversarial Network Models", in *Proc. of the ACM on Measurement and Analysis of Computing Systems*, Vol. 2, No. 1, pp.11, 2018.
[21] G. S. Paschos and L. Tassiulas, "Sustainability of Service Provisioning Systems Under Stealth DoS Attacks", in *IEEE Trans. on Control of Network Systems*, Vol. 4, No. 4, pp. 749-760, 2017.
[22] D. Shah and D. Wischik, "Fluid models of congestion collapse in overloaded switched networks," in *Queueing Systems*, vol. 69, no. 2, pp: 121, 2011.
[23] D. Shah and D. Wischik, "Switched networks with maximum weight policies: Fluid approximation and multiplicative state space collapse," in *The Annals of Applied Probability*, Vol. 22, No. 1, pp. 70-127, 2012.
[24] G. Fayolle, V. A. Malyshev and M. V. Men'shikov, "Topics in the constructive theory of countable Markov chains," Cambridge university press, 1995.
[25] N. Avrahami and Y. Azar, "Minimizing total flow time and total completion time with immediate dispatching," in *Algorithmica*, Vol. 47, No. 3, pp. 253-268, 2007.
[26] I. Grosof, Z. Scully and M. Harchol-Balter, "Load Balancing Guardrails: Keeping Your Heavy Traffic on the Road to Low Response Times," in *Proc. of the ACM on Measurement and Analysis of Computing Systems*, Vol. 3, No. 2, pp. 42, 2019.
[27] D. Berger, M. Karsten and J. Schmitt, "On the relevance of adversarial queueing theory in practice," in *ACM SIGMETRICS Performance Evaluation Review*, Vol. 42, No. 1, pp. 343-354, 2014.
[28] CW. Tan, DM. Chiu, J. CS. Lui and D. KY. Yau, "A distributed throttling approach for handling high bandwidth aggregates," in *IEEE Trans. on Parallel and Distributed Systems*, Vol. 18, No. 7, pp. 983-995, 2007.
[29] L. Georgiadis, L. Tassiulas, "Optimal overload response in sensor networks", in *IEEE Trans. on Information Theory*, Vol. 52, No. 6, pp. 2684-2696, 2006.
[30] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network flows", 1988.

A PROOF OF CLAIMS

A.1 Proof of Claim 1

We prove the claim by showing that we can find such n on each sample path (except a set of measure zero). Fixing an arbitrary sample path, by (5), there exists an $n_1 \in S'$ such that $\lim_{t\to\infty} \frac{Q_{n_1}(t)}{t} >$

0. If $\lim_{t\to\infty} \frac{Q^u_{n_1}(t)}{t} > 0$, then we are done. Otherwise, we have that $\lim_{t\to\infty} \frac{Q^v_{n_1}(t)}{t} > 0$ and $\lim_{t\to\infty} \frac{Q^u_{n_1}(t)}{t} = 0$. Since server n_1 is overloaded and its service discipline is FCFS, $\lim_{t\to\infty} \frac{Q^u_{n_1}(t)}{t} = 0$ implies that $\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a^u_{n_1}(i)}{t} = 0$. This means that the time-average traffic rate from $U_{S'}$ to n_1 is zero. Then, we show that there must exist another overloaded server, other than n_1. We consider the subset of servers $S'_1 = S'\backslash\{n_1\}$. Note that since $\sum_{m\in V} f^*_{mn_1} \leq \mu_{n_1}$, we have that $\sum_{m\in V}\sum_{n\in S'_1} f^*_{mn} \geq val(S') - \mu_{n_1}$. It follows that

$$
\begin{aligned}
&\lim_{t\to\infty} \frac{\sum_{n\in S'_1} Q_n(t)}{t} \\
\geq &\lim_{t\to\infty} \sum_{n\in S'} \left(\frac{\sum_{i=0}^{t-1} a^u_n(t)}{t} + \frac{\sum_{i=0}^{t-1} a^v_n(t)}{t} - \frac{\sum_{i=0}^{t-1} b_n(t)}{t} \right) \\
\geq &\sum_{u_l\in U_{S'}} \lambda^u_l + \sum_{m\in V}\sum_{n\in S'_1} f^*_{mn} - \sum_{n\in S'_1} \mu_n \\
\geq &[val(S') - \mu_n] - [\Delta(S') - \mu_n] \\
> &0,
\end{aligned}
$$

where the second inequality follows from $\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a^u_{n_1}(i)}{t} = 0$. Therefore, we can repeat the argument above, that there must exist another $n_2 \in S'_1$ such that $\lim_{t\to\infty} \frac{Q_{n_2}(t)}{t} > 0$, i.e., there exists an overloaded server in S'_1. And if $\lim_{t\to\infty} \frac{Q^u_{n_2}(t)}{t} = 0$, then again, it implies that the time-average traffic rate from $U_{S'}$ to n_2 is also zero. By repeating such argument, we will arrive at one of the two following situations: (i) we find a queue n_i such that $\lim_{t\to\infty} \frac{Q^u_{n_i}(t)}{t} > 0$, and the claim follows; (ii) we establish that for some u_l, $\lim_{t\to\infty} \frac{Q_n(t)}{t} > 0$ for all $n \in S_{u_l}$, and prove the claim by Observation 1. Therefore, with probability 1, there exists some n such that $\lim_{t\to\infty} Q^u_n(t)/t > 0$, which completes the proof of Claim 1. □

A.2 Proof of Claim 2

Having shown that $U_{S'}$ is non-empty, we now proceed to demonstrate that $val(S') > \Delta(S')$. Note that for every $n \in S'$, $\lim_{t\to\infty} Q_n(t)/t > 0$. It follows that the expected sum arrival rate at server n must be greater than its service rate. Therefore, for each $n \in S'$,[8]

$$
\begin{aligned}
&\lim_{t\to\infty} \left(\frac{\sum_{i=0}^{t-1} a^u_n(i)}{t} + \frac{\sum_{i=0}^{t-1} a^v_n(i)}{t} - \frac{\sum_{i=0}^{t-1} b_n(i)}{t} \right) \\
= &\lim_{t\to\infty} \left(\frac{\sum_{i=0}^{t-1} a^u_n(i)}{t} + \frac{\sum_{i=0}^{t-1} a^v_n(i)}{t} \right) - \mu_n > 0.
\end{aligned} \tag{15}
$$

It follows that for each $n \in S'$

$$
\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a^u_n(i)}{t} + \min\left(\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a^v_n(i)}{t} - \mu_n, 0 \right) \geq 0. \tag{16}
$$

Based on this, we define a set of $\{f\}_{mn}$ as:

$$
f_{mn} = \lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a^v_{mn}(i)}{t}, \quad \text{if } \lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a^v_n(i)}{t} \leq \mu_n,
$$

[8] To avoid unnecessary complexity, we assume that the time average of arrivals exist. Without the assumption, the argument would still hold by replacing lim with lim inf.

and otherwise,

$$f_{mn} = \left(\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a_{mn}^{v}(i)}{t} / \lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a_{n}^{v}(i)}{t} \right) \cdot \mu_n.$$

It is easy to verify that $\{f\}_{mn}$ satisfies the constraints of $LP(S')$. Furthermore, since $U_{S'}$ is non-empty, there exists $n \in S'$ such that $\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a_n^u(i)}{t} > 0$. Taking such an n, and combining with (15), we have

$$\lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a_n^u(i)}{t} + \sum_{m\in V} f_{mn} - \mu_n > 0.$$

It then follows from (16) that

$$\sum_{n\in S'} \lim_{t\to\infty} \frac{\sum_{i=0}^{t-1} a_n^u(i)}{t} + \sum_{n\in S'} \sum_{m\in V} f_{mn} - \sum_{n\in S'} \mu_n > 0.$$

Hence, we have

$$0 < \sum_{u_l \in U_{S'}} \lambda_l^u - \sum_{n\in S'} \mu_n + \sum_{n\in S'} \sum_{m\in V} f_{mn} \le val(S') - \Delta(S'),$$

which implies that S' follows the val-condition. □

A.3 Proof of Claim 3

The proof is done by contradiction. Assume that Claim 3 does not hold, i.e., there is no such user dispatcher u_l, that is, every user dispatcher has connection to $S\backslash\tilde{S}'$. First, we show that the optimal solution λ^* must satisfy that

$$\lambda_{ln}^* = 0, \lambda_{mn}^* = 0, \forall n \in \tilde{S}', \text{and } u_l, v_m \text{ that have connection to } S\backslash\tilde{S}'. \tag{17}$$

This is true since that if the claim does not hold, we can decrease some positive λ_{ln}^* (or λ_{mn}^*) by a small amount $\delta > 0$ and add to $\lambda_{ln'}^*$ (or $\lambda_{mn'}^*$) with $n' \in S\backslash\tilde{S}'$. This will result in that $\boldsymbol{r}_n^* > 0$ decreases by δ and $\boldsymbol{r}_{n'}^* = 0$ increases by δ, thereby obtaining a $\boldsymbol{r}$ with a smaller value of $\sum_n \boldsymbol{r}_n^2$, which contradicts the optimality of $\boldsymbol{r}^*$.

Let $\tilde{V}_{S'}$ be the subset of adversary dispatchers that only have connections in $\tilde{S}'$. Note that $\tilde{V}_{S'} \subseteq V_{S'}$. By (17), we have $\lambda_{mn}^* = 0$ for all $m \notin \tilde{V}_{S'}, n \in \tilde{S}'$. Therefore,

$$\begin{aligned} \sum_{m\in V_{S'}} \sum_{n\in S'\backslash\tilde{S}'} \lambda_{mn}^* &\ge \sum_{m\notin \tilde{V}_{S'}} \sum_{n\in S'\backslash\tilde{S}'} \lambda_{mn}^* \\ &= \sum_{m\notin \tilde{V}_{S'}} \sum_{n\in S'} \lambda_{mn}^* = \sum_{m\notin \tilde{V}_{S'}} \lambda_m^v. \end{aligned} \tag{18}$$

Now, recall the linear program $LP(S')$ that defines $val(S')$. Let $\{f^*\}_{mn}$ be a set of optimal solution to $LP(S')$. We have by the definition of $\tilde{V}_{S'}$,

$$\sum_{m\in \tilde{V}_{S'}} \sum_{n\in S'} f_{mn}^* = \sum_{m\in \tilde{V}_{S'}} \sum_{n\in \tilde{S}'} f_{mn}^* \le \sum_{n\in \tilde{S}'} \mu_n. \tag{19}$$

Furthermore,

$$\sum_{m\notin \tilde{V}_{S'}} \sum_{n\in S'} f_{mn}^* \le \sum_{m\notin \tilde{V}_{S'}} \lambda_m^v \le \sum_{m\in V_{S'}} \sum_{n\in S'\backslash\tilde{S}'} \lambda_{mn}^*, \tag{20}$$

where the first part follows from Constraint (3) of $LP(S')$ and the second part follows from (18). Hence, combining (19) and (20), we have,

$$\begin{aligned} val(S') - \sum_{n\in\tilde{S}'} \mu_n &= \sum_{m\in\tilde{V}_{S'}} \sum_{n\in S'} f^*_{mn} + \sum_{m\notin\tilde{V}_{S'}} \sum_{n\in S'} f^*_{mn} - \sum_{n\in\tilde{S}'} \mu_n \\ &\le \sum_{m\notin\tilde{V}_{S'}} \sum_{n\in S'} f^*_{mn} \le \sum_{m\in V_{S'}} \sum_{n\in S'\setminus\tilde{S}'} \lambda^*_{mn}. \end{aligned} \tag{21}$$

Again, by (17), we have

$$\begin{aligned} \sum_{l\in U} \sum_{n\in S'\setminus\tilde{S}'} \lambda^*_{ln} &\ge \sum_{l\in U_{S'}} \sum_{n\in S'\setminus\tilde{S}'} \lambda^*_{ln} \\ &= \sum_{l\in U_{S'}} \sum_{n\in S'} \lambda^*_{ln} = \sum_{l\in U_{S'}} \lambda^u_l. \end{aligned} \tag{22}$$

Then, summing up constraints (9) of $\mathcal{P}$ over $n \in S'\setminus\tilde{S}'$, we obtain

$$\sum_{n\in S'\setminus\tilde{S}'} r^*_n \ge \sum_{n\in S'\setminus\tilde{S}'} \sum_{l:n\in S_{u_l}} \lambda^*_{ln} + \sum_{n\in S'\setminus\tilde{S}'} \sum_{m:n\in S_{v_m}} \lambda^*_{mn} - \sum_{n\in S'\setminus\tilde{S}'} \mu_n \tag{23}$$

$$\ge \sum_{l\in U_{S'}} \sum_{n\in S'\setminus\tilde{S}'} \lambda^*_{ln} + \sum_{m\in V_{S'}} \sum_{n\in S'\setminus\tilde{S}'} \lambda^*_{mn} - \sum_{n\in S'\setminus\tilde{S}'} \mu_n \tag{24}$$

$$\ge \sum_{l\in U_{S'}} \lambda^u_l + val(S') - \sum_{n\in\tilde{S}'} \mu_n - \sum_{n\in S'\setminus\tilde{S}'} \mu_n \tag{25}$$

$$\ge \sum_{l\in U_{S'}} \lambda^u_l + val(S') - \sum_{n\in S'} \mu_n = val(S') - \Delta(S') > 0, \tag{26}$$

where Inequality (23) follows from rearrangement of the sums, Inequality (24) follows from (22) and Inequality (25) follows from Inequality (21). Observe that by definition of $\tilde{S}'$, $\sum_{n\in S'\setminus\tilde{S}'} r^*_n = 0$, which contradicts (26). Hence, there exists a u_l such that $S_{u_l} \in \tilde{S}'$. This concludes the proof of the lemma.

A.4 Proof of Claim 4

By the reasoning above, we have

$$\begin{aligned} &f((\boldsymbol{Q}(T), S(T)) - f((\boldsymbol{Q}(0), S(0)) \\ =&\mathbb{1}\{S(T) \in \mathcal{S}_0, \min_{n\in S(T)}\{Q_n(T)\} > TC_1\} \cdot \left(\min_{n\in S(T)}\{Q_n(T)\} - TC_1\right) \\ &- \mathbb{1}\{S(0) \in \mathcal{S}_0, \min_{n\in S(0)}\{Q_n(0)\} > TC_1\} \cdot \left(\min_{n\in S(0)}\{Q_n(0)\} - TC_1\right) \\ =&\mathbb{1}\{\min_{n\in S(T)}\{Q_n(T)\} > TC_1\} \cdot \left(\min_{n\in S(T)}\{Q_n(T)\} - TC_1\right) \\ &- \left(\min_{n\in S(0)}\{Q_n(0)\} - TC_1\right) \\ \ge&\left(\min_{n\in S(0)}\{Q_n(T)\} - TC_1\right) - \left(\min_{n\in S(0)}\{Q_n(0)\} - TC_1\right) \end{aligned}$$

$$= \min_{n\in S(0)}\{Q_n(T)\} - \min_{n\in S(0)}\{Q_n(0)\}$$

Hence, we have

$$\begin{aligned}&\mathbb{E}\left[f((Q(T),S(T)) - f((Q(0),S(0)) \mid f(Q(0),S(0)) > 0\right]\\ \geq&\mathbb{E}\left[\min_{n\in S(0)}\{Q_n(T)\} - \min_{n\in S(0)}\{Q_n(0)\} \mid f(Q(0),S(0)) > 0\right].\end{aligned}$$

A.5 Proof of Claim 5

For all $t \geq T_{\epsilon_{S(0)}}$, we have

$$\begin{aligned}&\mathbb{P}\left\{\forall n \in S(0), \frac{Q_n^o(t)}{t} \geq \frac{\tilde{r}_n^*}{2}\right\} \geq 1-\epsilon_{S(0)}\\ \Longrightarrow&\mathbb{P}\left\{\forall n \in S(0), Q_n^o(t) \geq \frac{\tilde{r}_n^* t}{2}\right\} \geq 1-\epsilon_{S(0)}\\ \Longrightarrow&\mathbb{P}\left\{\forall n \in S(0), \tilde{Q}_n(t) \geq \frac{\tilde{r}_n^* t}{2} + Q^*(0)\right\} \geq 1-\epsilon_{S(0)}\\ \Longrightarrow&\mathbb{P}\left\{\forall n \in S(0), Q_n(t) \geq \frac{\tilde{r}_n^* t}{2} + Q^*(0) - N_1C_1\right\} \geq 1-\epsilon_{S(0)}\\ \Longrightarrow&\mathbb{P}\left\{\min_{n\in S(0)} Q_n(t) \geq \frac{\tilde{r}_n^* t}{2} + Q^*(0) - N_1C_1\right\} \geq 1-\epsilon_{S(0)}.\end{aligned}$$

On the other hand, we also have $\min_{n\in S(0)} Q_n(t) \geq Q^*(0) - C_1 t$ with probability 1 because the queue lengths can decrease by at most C in each time slot. Now, we set the previously mentioned T as $\max_{S(0)\in\mathcal{S}_0} \max\{T_{\epsilon_{S(0)}}, \frac{8N_1C_1}{\min_{n\in S(0)}\tilde{r}_n^*}\}$, we have that

$$\begin{aligned}&\mathbb{E}\left[\min_{n\in S(0)}\{Q_n(T) - \min_{n\in S(0)}\{Q_n(0)\} \mid f(Q(0),S(0)) > 0\right]\\ \geq&(1-\epsilon_{S(0)})\cdot\frac{\min_{n\in S(0)}\tilde{r}_n^* T}{2} - (1-\epsilon_{S(0)})N_1C_1 - \epsilon_{S(0)}C_1T\\ \geq&\frac{\min_{n\in S(0)}\tilde{r}_n^* T}{4} - N_1C_1 \geq N_1C_1 > 0.\end{aligned}$$

By Lemma 2, this establishes that the Markov Chain is transient. Since the chain is irreducible, its transience implies that starting from any initial state, each state is visited finitely many times with probability 1. By the definition of f, the set $\{(\boldsymbol{Q},S) \mid f(\boldsymbol{Q},S) = 0\}$ is finite. Therefore, we have, starting from any initial state, $f(\boldsymbol{Q}(t),S(t)) = 0$ happens finitely often with probability 1. It follows that, on each sample path (except a set with measure zero), the adversary identifies a vulnerable subset and never changes the target after a finite time. Since the number of vulnerable sets $|\mathcal{S}_0|$ is finite, there exists a vulnerable set $S' \in \mathcal{S}_0$ such that with probability at least $\frac{1}{|\mathcal{S}_0|}$, after a finite time, the adversary keeps injecting traffic to S' following the JSQ rule. Invoking Proposition 1, we have that with positive probability,

$$\forall n \in S', \lim_{t\to\infty}\frac{Q_n(t)}{t} > 0,$$

It follows from Observation 1 that, $\exists n \in S'$, $\lim_{t\to\infty} \frac{Q_n^u(t)}{t} > 0$ with positive probability. Therefore, by Lemma 1, we have,

$$\exists n, \lim_{t\to\infty} \frac{\mathbb{E}\left[Q_n^u(t)\right]}{t} > 0. \tag{27}$$

Hence the Min-Zero policy destabilizes user traffic.

B PROOF OF PROPOSITIONS

B.1 Proof of Propositions 1, 2 and Their Multi-hop Generalizations

We prove Propositions 1 and 2 by applying the results in [22, 23]. As stating the results involve heavy additional notations and modeling in [22, 23], we do not present them here rigorously, but simply describe them at an intuitive level and show how to apply them to prove our propositions.

The relevant results are Theorem 1 in [22] and Corollary 4.4 in [23]. Theorem 1 in [22] considers the fluid model of switch networks that employ Max-Weight scheduling, which subsume the JSQ server farm considered in this paper as a special case. It establishes that the fluid model solution divided by time converges to the optimal solution of certain optimization problem. Projecting this onto our case, the optimization problem corresponds to $\mathcal{P}$ in Proposition 1 and $\mathcal{P}'$ in Proposition 2 under the alternative dynamics. Corollary 4.4 in [23] shows the convergence of scaled queue length vector of network (single-hop or multi-hop) under Max-Weight (Back-pressure) routing to the corresponding fluid model solution. Combining these two results, setting the parameter t (different with t in our model) in Corollary 4.4 of [23] as 1, it follows that in our model $\boldsymbol{Q}(t)/t$ converges to the solution to $\mathcal{P}$ (or $\mathcal{P}'$ under the relaxed dynamics) in probability, which concludes the proof of Propositions 1 and 2.

One can straightforwardly extend Theorem 1 in [22] to the multi-hop network in our model. Combining again with Corollary 4.4 in [23], it leads to Proposition 5 that we use in extending our results to multi-hop networks. Consider a directed network $\mathcal{G}(\mathcal{N}, \mathcal{E})$ evolving in discrete time. The network $\mathcal{G}$ has a single destination node d and multiple sources. The external traffic arrivals and offered transmission of network links are independent random variables and are i.i.d. across time. The time average external traffic arrival rate at node n is denoted as λ_n ($\lambda_n = 0$ if n is not a source), and the time average offered transmission rate at link $(n, n') \in \mathcal{E}$ is denoted as $c_{nn'}$. Note that under our assumptions, the time average rates are equal to the mean of the corresponding random variables. The network operates under the (local) back-pressure policy introduced in Section 7. Consider the following optimization problem $\mathcal{P}_m$:

$$\begin{aligned} \min \sum_n r_n^2 \\ \sum_{n':(n',n)\in\mathcal{E}} f_{n',n} + r_n + \lambda_n \geq \sum_{n':(n,n')\in\mathcal{E}} f_{nn'}, \qquad \forall n \in \mathcal{N}\backslash\{d\} \\ 0 \leq f_{nn'} \leq c_{nn'}, \qquad \forall (n,n') \in \mathcal{E} \\ r_n \geq 0, \qquad \forall n \in \mathcal{N}. \end{aligned}$$

It is easy to verify that $\mathcal{P}_m$ is a convex optimization problem and thus have a unique optimal solution. Now, we are ready to present the proposition in the multi-hop setting.

Proposition 5. *Let $\boldsymbol{r}^* = (r_1^*, \ldots, r_n^*)$ be the optimal solution to the optimization problem $\mathcal{P}_m$ associated with the multi-hop network. The queue lengths satisfy that for any $\delta > 0$*

$$\lim_{t\to\infty} \mathbb{P}\left\{\left|\frac{Q_n(t)}{t} - r_n^*\right| < \delta\right\} = 1.$$

Note that Proposition 5 is the counterpart of Proposition 1 in the multi-hop setting. One can easily extend Proposition 2 to the multi-hop setting in almost identical fashion, which we omit here.

B.2 Proof of Proposition 3

We first show a simplified version of Proposition 3: if $\boldsymbol{r}^* > \mathbf{0}$, then $\boldsymbol{r}^* = \tilde{\boldsymbol{r}}^*$, and then extend the proof to the original proposition. We prove the simplified version by establishing two claims: (i). Under the condition that $\boldsymbol{r}^* > \mathbf{0}$, if $\tilde{\boldsymbol{r}}^* \geq \mathbf{0}$, then $\boldsymbol{r}^* = \tilde{\boldsymbol{r}}^*$ and (ii). If $\boldsymbol{r}^* > \mathbf{0}$, then $\tilde{\boldsymbol{r}}^* \geq \mathbf{0}$. These two claims combined yield the simplified version.

We start with the first claim. Observe that if $\boldsymbol{r}_n^* > 0$ in the optimal solution to $\mathcal{P}$, then the corresponding constraint (9) must be satisfied with equality, since otherwise we can decrease $\boldsymbol{r}_n^*$ and obtain a better solution. Therefore, $\boldsymbol{r}^*$ must be feasible to $\mathcal{P}'$. Conversely, if some $\tilde{\boldsymbol{r}} \geq \mathbf{0}$ is feasible to $\mathcal{P}'$, then it must also be feasible to $\mathcal{P}$. Combining these, we have under the condition that $\boldsymbol{r}^* > \mathbf{0}$, $\tilde{\boldsymbol{r}}^*$ is feasible to $\mathcal{P}$, and if $\tilde{\boldsymbol{r}}^* \geq \mathbf{0}$, then $\boldsymbol{r}^*$ is feasible to $\mathcal{P}'$. Since both $\mathcal{P}$ and $\mathcal{P}'$ have unique optimal solution and they have the same objective function, we have $\boldsymbol{r}^* = \tilde{\boldsymbol{r}}^*$. We now proceed to establish the second claim. If there is an n such that $\tilde{r}_n^* < 0$, we consider a dispatcher l such that $n \in S_{u_l}$. We claim that for all $n' \in S_{u_l}$, $\tilde{r}_{n'}^* < 0$. Since otherwise if there is a $n' \in S_{u_l}$ with $\tilde{r}_{n'}^* \geq 0$, we can increase λ_{ln} and decrease $\lambda_{ln'}$ and form a vector $\tilde{\boldsymbol{r}}'$ with $||\tilde{\boldsymbol{r}}'||^2 < ||\tilde{\boldsymbol{r}}^*||^2$, which contradicts the optimality of $\tilde{\boldsymbol{r}}^*$. We repeat this argument for other user dispatchers that have connections to S_{u_l}, until we arrive at a subset of servers S' that satisfies: (i). $\tilde{r}_n^* < 0$ for all $n \in S'$ and (ii). there does not exist a user dispatcher that has connection to both servers in S' and servers in $S \backslash S'$. This implies that in the optimization problem $\mathcal{P}$, it is feasible to have $r_n = 0$ for all $n \in S'$ since the constraints for $n \in S'$ and $n \in S \backslash S'$ do not interfere with one another. Hence, by setting the entries in $\boldsymbol{r}^*$ that correspond to all $n \in S'$ to zero, we obtain a better solution to $\mathcal{P}$, which contradicts the optimality of $\boldsymbol{r}^*$. Hence, we prove the second claim, and conclude the proof of: if $\boldsymbol{r}^* > \mathbf{0}$, then $\boldsymbol{r}^* = \tilde{\boldsymbol{r}}^*$.

Now, we extend the proof to Proposition 3. For an arbitrary n such that $r_n^* > 0$, again, consider some dispatcher l such that $n \in S_{u_l}$. We have $r_{n'}^* > 0$ for all $n' \in S_{u_l}$, since otherwise one can decrease λ_{ln} and increase $\lambda_{ln'}$ for some n' with $r_{n'}^* \leq 0$ and obtain a better $\boldsymbol{r}$ than $\boldsymbol{r}^*$. Repeating this argument, by a similar reasoning as above, we will arrive at a subset of servers S' that satisfies $r_n^* > 0$ for all $n \in S'$ and there is no user dispatcher that has connection to both S' and $S \backslash S'$. Therefore, we can decompose $\mathcal{P}$ into two parts that correspond to S' and $S \backslash S'$ respectively. Applying the previously proved simplified version to the part of S', we have $r_{n'}^* = \tilde{r}_{n'}^*$ for all $n' \in S'$, which include the n we start with. Hence, we conclude the proof of the proposition.

B.3 Proof of Proposition 4

Define $d_n(t)$ as $\tilde{Q}_n(t) - Q_n(t)$. We have $d_n(0) \leq 0$ for all n, and if we can show that $d_n(t) \leq N_1 LC$, then we prove the proposition. We first give three observations regarding $d_n(t)$:

(1) $\sum_n d_n(t) = \sum_n d_n(0) \leq 0$ for all t.
(2) Service at server n does not change $d_n(t)$.
(3) If $d_n(t+1) > d_n(t) > 0$, then there exists an n' such that $d_{n'}(t) \geq d_n(t)$.

The three observations are justified as follows. Since in the relaxed dynamics, $Q_n(t+1) = Q_n(t) + a_n(t) - b_n(t) = Q_n(t) + \sum_l a_{ln}(t) - b_n(t)$, we have

$$\sum_{n=1}^{N} Q_n(t) = \sum_{n=1}^{N} Q_n(0) + \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{i=0}^{t-1} a_{ln}(i) - \sum_{n=1}^{N} \sum_{i=0}^{t-1} b_n(i)$$

$$= \sum_{n=1}^{N} Q_n(0) + \sum_{l=1}^{L}\sum_{i=0}^{t-1} a_l(i) - \sum_{n=1}^{N}\sum_{i=0}^{t-1} b_n(i) \tag{28}$$

Since $\boldsymbol{Q}$ and $\tilde{\boldsymbol{Q}}$ evolve under the same sequence of $\{\boldsymbol{a}(i)\}$ and $\{\boldsymbol{b}(i)\}$, (28) validates the first observation. The second observation can be seen from the fact that after some service $b_n(t)$ for server n at time t, $Q_n(t)$ and $\tilde{Q}_n(t)$ both decrease by an amount of $b_n(t)$. Hence, $d_n(t)$ cannot be changed by services. The third observation can be established as follows: if at t, $d_n(t) > 0$ and $d_n(t)$ increases at the next time slot, then there must exist a dispatcher u_l (with non-zero arrival at t) such that following the JSQ rule, u_l sends packets to n under queue length vector is $\tilde{Q}_n(t)$ but does not send packets to n under $Q_n(t)$. It follows that there exists n' such that $\tilde{Q}_{n'}(t) \geq \tilde{Q}_n(t)$ while $Q_{n'}(t) \leq Q_n(t)$. This implies that $d_{n'}(t) = \tilde{Q}_{n'}(t) - Q_{n'}(t) \geq \tilde{Q}_n(t) - Q_n(t) = d_n(t)$.

By Observation (2), we do not need to consider services. Hence, we focus on arrivals to queues from dispatchers and proceed to prove the proposition. Suppose for the sake of contradiction that there exists a time t and server n_1 such that $d_{n_1}(t) > N_1 LC$. We take t_1 to be the smallest t that satisfies the above condition. Since at each time slot, the total arrival to a server is at most LC, $t_1 > N_1$. Due to the same reason, we have $d_n(t_1 - 1) > (N_1 - 1)LC$. By definition of t_1, we have $d_n(t) \leq N_1 LC$ for $0 \leq t \leq t_1$, and in particular, $d_{n_1}(t_1 - 1) < d_{n_1}(t_1)$. Hence, by observation (3), there must exist another server n_2 such that $d_{n_2}(t_1 - 1) \geq d_{n_1}(t_1 - 1) > (N_1 - 1)LC$. Next, consider the function $d_{n_1}(t) + d_{n_2}(t)$, let t_2 be the largest time slot such that $0 \leq t_2 < t_1 - 1$ and $d_{n_1}(t_2 + 1) + d_{n_2}(t_2 + 1) > d_{n_1}(t_2) + d_{n_2}(t_2)$. Such t_2 must exist as $d_{n_1}(0) + d_{n_2}(0) \leq 0$ while $d_{n_1}(t_1 - 1) + d_{n_2}(t_1 - 1) > 2(N_1 - 1)LC$. Note that since $d_{n_1}(t) + d_{n_2}(t)$ does not increase from $t_2 + 1$ to $t_1 - 1$, we have for $t_2 + 1 \leq t < t_1 - 1$

$$d_{n_1}(t) + d_{n_2}(t) > d_{n_1}(t_1 - 1) + d_{n_2}(t_1 - 1) > 2(N_1 - 1)LC.$$

Combining this with $d_n(t) \leq N_1 LC$ for $n = n_1, n_2, t \leq t_1$, we have $d_n(t_2 + 1) > (N_1 - 2)LC$ and $d_n(t_2) > (N_1 - 3)LC$ for $n = n_1, n_2$. Since $d_{n_1} + d_{n_2}$ increases at t_2, there must exist a dispatcher such that it sends packets to some $n \in \{n_1, n_2\}$ (w.l.o.g. $n = n_1$) under $\tilde{\boldsymbol{Q}}$ but does not send packets to either n_1 or n_2 under $\boldsymbol{Q}$. Following similar reasoning as in establishing observation (3), we have that there exists n_3 such that $d_{n_3}(t_2) \geq d_{n_1}(t_2) > (N_1 - 3)LC$. Applying the same argument with the largest time slot t_3 ($0 \leq t_3 \leq t_2 - 1$) at which the function $d_{n_1}(t) + d_{n_2}(t) + d_{n_3}(t)$ increases, we have for $t_3 + 1 \leq t < t_2 - 1$

$$\begin{aligned} d_{n_1}(t) + d_{n_2}(t) + d_{n_3}(t) &> d_{n_1}(t_2 - 1) + d_{n_2}(t_2 - 1) + d_{n_3}(t_2 - 1) \\ &> 3(N_1 - 3)LC. \end{aligned}$$

And we have $d_n(t_3 + 1) > (3(N_1 - 3) - 2N_1)LC = (N_1 - 9)LC$ and $d_n(t_3) > (N_1 - 10)LC$ for $n = n_1, n_2, n_3$. It follows that there exists another n_4 with $d_{n_4} > (N_1 - 10)LC$. Repeating such argument, we will find a time t_N such that $d_n(t_N) > LC$ for all n, which implies that $\sum_{n=1}^{N} d_n(t_N) > NLC$. This contradicts observation (1). Thus, we conclude the proof.

C PROOF OF THEOREMS FOR MULTI-HOP NETWORKS

C.1 Proof of Theorem 4

(Sketch) Note that under any adversarial injection policy, the network $\mathcal{G}$ can be considered as a network that employs Back-pressure routing with multiple sources and single destination. The proof follows similar line as that of Theorem 1, and additionally relies on the following observation regarding single-commodity back-pressure network, which can be obtained from results in [29].

Observation 2. *For each sample path, if* $\lim_{t\to\infty} Q_n(t)/t > 0$ *for some* $n \in \mathcal{N}$, *then for all* $n' \in Out(n)$, *the time average traffic rate from* n *to* n' *equals* $c_{nn'}$.

We first show the sufficiency of the condition. If some cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ satisfies the extended val-condition, we consider the randomized adversarial injection policy given by the max-flow that defines $val(\mathcal{S})$. Then on an arbitrary sample path (except a set of measure zero), there must exist node n with $\lim_{t\to\infty} Q_n(t)/t > 0$. Let $\mathcal{S}'$ be the set of nodes n with $\lim_{t\to\infty} Q_n(t)/t > 0$. We claim that $s \in \mathcal{S}'$, and that the adversary destabilizes user traffic directly follows. To justify the claim, suppose for the sake of contradiction $s \notin \mathcal{S}'$, then the time average traffic rate from s to $\mathcal{S}'$ must be zero, and the overflow of $\mathcal{S}'$ is caused solely by adversary traffic. Let $\tilde{\mathcal{S}} = \mathcal{S}' \cap \mathcal{S}$. We have that the adversary traffic that goes through the cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ does not pass $\tilde{\mathcal{S}}$. Let $\tilde{\lambda}$ be the time average rate of adversarial traffic that goes from $\mathcal{S}\backslash\tilde{\mathcal{S}}$ to $\mathcal{N}\backslash\mathcal{S}$. By the definition of the val function, we have that

$$\sum_{n\in\mathcal{S}\backslash\tilde{\mathcal{S}}, n'\in\mathcal{N}\backslash\mathcal{S}} c_{nn'} - \tilde{\lambda} - \lambda \le Cap(\mathcal{S}) - val(\mathcal{S}) - \lambda < 0, \tag{29}$$

which leads to a contradiction since (29) implies that there exists node not in $\mathcal{S}'$ whose queue length also grows linearly with time at some positive rate.

We now proceed to the necessity part. If there exists a policy that destabilizes user traffic, there exists a sample path at which the user part of some queue grows at positive rate with time. Pick such a sample path, let $\mathcal{S}$ be the set of nodes whose queue length grow with time at positive rate on the sample path. It is easy to see that $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ is an s-d cut. We proceed to show that it satisfies the extended val-condition. Let λ^v be the total adversarial traffic injection rate under the policy. Denote r^u, r^v respectively as the sum of user and adversary parts of queue length growth rates of nodes in $\mathcal{S}$, and p^u, p^v respectively as the total rates of user and adversary traffic that goes through the cut $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$. Based on the definitions, we have $\lambda^v = p^v + r^v$ and $\lambda_s = p^u + r^u$. Further by Observation 2, we have $p^v + p^u = Cap(\mathcal{S})$. It follows that

$$\begin{aligned} val(\mathcal{S}) + \lambda_s &= p^u + r^u + val(\mathcal{S}) \\ &= p^u + val(\mathcal{S}) - Cap(\mathcal{S}) - r^u + Cap(\mathcal{S}) \\ &\ge p^u + p^v - Cap(\mathcal{S}) + r^u + Cap(\mathcal{S}) \\ &= r^u + Cap(\mathcal{S}) > Cap(\mathcal{S}). \end{aligned}$$

Hence, $(\mathcal{S}, \mathcal{N}\backslash\mathcal{S})$ satisfies the extended val-condition and we conclude the proof.

C.2 Proof of Theorem 5

(Sketch) The proof follows the same road-map of that of Theorem 3. First, we establish multi-hop counterparts of Propositions 1 and 2 in Proposition 5 which will be given in the Appendix B.1. Following similar ideas and using Proposition 5, we can generalize Theorem 2 and Corollary 3 to multi-hop networks, that is, if the problem is feasible, then there exists an s-d cut that satisfies the extended val-condition and by each adversary source node v_m injecting to $\mathcal{N}_{v_m} \cap \mathcal{S}$ following the JSQ rule, all the queues in $\mathcal{S}$ grow with time, i.e., $\forall n \in \mathcal{S}$, $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n(t)]}{t} > 0$. Define the set of all such s-d cuts as vulnerable cuts $\mathcal{S}_0$. Next, we again interpret the network dynamics under the Multi-hop Min-Zero policy as a Markov Chain with state $(\boldsymbol{Q}(t), \mathcal{S}(t))$. We construct the same Lyapunov function as in the proof of Theorem 3 by choosing an appropriate T:

$$\begin{aligned} f(\mathbf{Q}(t), S(t)) =& \mathbb{1}\{S(t) \in \mathcal{S}_0, \min_{n\in S(t)}\{Q_n(t)\} > TC_1\} \cdot \\ & \left(\min_{n\in S(t)}\{Q_n(t)\} - TC_1\right), \end{aligned}$$

where $C_1 = (|\mathcal{N}| + M)C$ is an upper bound on the change of queue length at ann node in one time slot. By a similar coupling bound as Proposition 4, together with Proposition 5, we establish the positive expected T-slot drift of f conditioned on $f((\boldsymbol{Q}(t), \mathcal{S}(t)) > 0$. Invoking Lemma 2, we demonstrate that the Markov chain is transient. Finally, going through the same reasoning as in Theorem 3, we show that there exists n such that $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^u(t)]}{t} > 0$, which concludes the proof.

D SUPPLEMENTARY PROOFS

In this section, we provide supplementary proofs to several probabilistic arguments in the paper. The proofs are mostly standard, and are provided here for completeness.

LEMMA 3. $\mathbb{E}[\lim_{t\to\infty} \frac{Q_n(t)}{t}] = \lim_{t\to\infty} \frac{\mathbb{E}[Q_n(t)]}{t}$. *The same holds for* Q_n^u *and* Q_n^v.

PROOF. The existence of the left-hand-side is guaranteed by our assumption. Since at each time slot, the arrival to a queue is bounded, we can find a constant C_0 such that $Q_n(t)/t \le C_0$ for all t. Hence, by dominated convergence theorem, we have

$$\mathbb{E}\left[\lim_{t\to\infty} \frac{Q_n(t)}{t}\right] = \lim_{t\to\infty} \mathbb{E}\left[\frac{Q_n(t)}{t}\right] = \lim_{t\to\infty} \frac{\mathbb{E}[Q_n(t)]}{t}.$$

□

LEMMA 4. *(Same as Lemma 1) A non-negative random variable* X *has zero expectation if and only if* $X = 0$ *with probability 1. Hence, if* $X > 0$ *with positive probability, then* $\mathbb{E}[X] > 0$.

PROOF. That $X = 0$ with probability 1 implies $\mathbb{E}[X] = 0$ is straightforward. We now prove the only if part. For each $N \in \mathbb{N}^+$, let E_N denote the event $\{X \ge 1/N\}$. Note that $\{E_N\}$ is a sequence of increasing events. Since X is non-negative, we have $\mathbb{E}[X] \ge \frac{1}{N}\mathbb{P}(E_N)$. As $\mathbb{E}[X] = 0$, it follows that $\forall N$, $\mathbb{P}(E_N) = 0$. Hence, by continuity of probability, we have

$$\mathbb{P}\{X > 0\} = \mathbb{P}\left(\bigcup_{N=1}^{\infty} E_N\right) = \lim_{N\to\infty} \mathbb{P}(E_N) = 0.$$

Therefore, $\mathbb{P}\{X = 0\} = 1$ and we conclude the proof. □

E NON-STATIONARY ADVERSARIES AND INSTABILITY CRITERION

In this section, we first remove the assumptions that $\lim_{t\to\infty} \frac{Q_n^u(t)}{t}$ exists almost surely, which will allow us to take into account non-stationary adversaries. We then discuss other criteria for instability and their effect on our results.

E.1 Non-stationary Adversaries

Without the assumption that $\lim_{t\to\infty} \frac{Q_n^u(t)}{t}$ exists almost surely, we need to modify the definition of the goal of the adversary by replacing the limit with lim inf or lim sup. More specifically, the first alternative is that the adversary destabilizes user traffic if

$$\text{For some } n \in \{1, \dots, N\}, \qquad \mathbb{E}\left[\liminf_{t\to\infty} \frac{Q_n^u(t)}{t}\right] > 0. \tag{30}$$

The second alternative is that the adversary destabilizes user traffic if

$$\text{For some } n \in \{1, \dots, N\}, \qquad \mathbb{E}\left[\limsup_{t\to\infty} \frac{Q_n^u(t)}{t}\right] > 0. \tag{31}$$

Under definitions (30) and (31), we allow the adversary to use an arbitrary, even non-stationary policy, as long as it satisfies the budget constraints.

Under definition (30), all of our results still hold. Theorem 1, the feasibility region of the attack, can be proved in a similar way with some minor changes. Specifically, the sufficiency part relies on constructing a stationary randomized policy that achieves the goal defined in (1), which implies (30). The necessity part follows by replacing the $\lim_{t\to\infty} \frac{Q_n^u(t)}{t}$ with $\liminf_{t\to\infty} \frac{Q_n^u(t)}{t}$. Since allowing for non-stationary policies does not enlarge the feasibility region, the Min-Zero policy is actually optimal over all policies (stationary and non-stationary).

Under definition (31), which is a weaker criterion than (30) since it is easier to achieve as $\limsup_{t\to\infty} \frac{Q(t)}{t} \geq \liminf_{t\to\infty} \frac{Q(t)}{t}$, our results would break if we consider non-stationary adversaries. As mentioned in [21], under this weak criterion, the feasibility region of non-stationary adversarial injection policies can be much larger than that of stationary policies. Indeed, an adversarial dispatcher can compromise all the servers it has connections to with service rates less than its budget by injecting packets following an exponentially increasing sequence. We provide an example in **Figure 8**, where a policy that injects traffic to servers following an exponentially increasing sequence trivially achieves the goal of (31). Such policy would not achieve the goal under definitions (30) or (1), which suggests that (31) be a less meaningful definition. Nonetheless, under (31), our Min-Zero policy is optimal over all stationary policies.

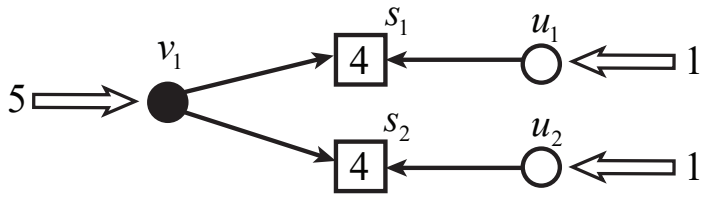

Fig. 8. Consider the above server farm with deterministic arrival and service rates. Define a sequence of time intervals as $\tau_1 = \{1\}, \tau_2 = \{2,3\}, \tau_3 = \{4,5,6,7\}, \cdots$ The adversary uses a non-stationary injection policy that sends all traffic to server S_1 during intervals τ_{2k} and sends all traffic to server S_2 during intervals τ_{2k+1} for $k \in \mathbb{N}$. For sufficiently large k, at the end of τ_{2k}, the adversarial traffic backlog in S_1 is at least 2^{2k-1}, which takes 2^{2k-3} time slots to drain. In the meantime, the user traffic in the queue of S_1 builds up to 2^{2k-3}. It follows that user traffic in Q_1 is de-stabilized by the definition (31). Similar argument holds for Q_2^u. This example can be generalized to one with arbitrary number of parallel servers.

E.2 Choice of Instability Criterion

Definitions (1), (30) and (31) all correspond to the mean rate-instability in queueing theory literature [10]. An alternative, and weaker criterion is to require $\mathbb{E}[\lim_{t\to\infty} Q_n^u(t)] = \infty$ ($\mathbb{E}[\liminf_{t\to\infty} Q_n^u(t)] = \infty$ or $\mathbb{E}[\limsup_{t\to\infty} Q_n^u(t)] = \infty$ when the limit does not exist).

If we use $\mathbb{E}[\lim_{t\to\infty} Q_n^u(t)] = \infty$ or $\mathbb{E}[\liminf_{t\to\infty} Q_n^u(t)] = \infty$ as the instability criterion, by the proof of Theorem 1, we can see that the feasibility region of the attack under the alternative criterion only differs with the original one at the boundary (this holds even for non-stationary attack policy). Therefore, the original feasibility region is the interior of the new feasibility region. Depending on the distributions of arrivals and services, the boundary points of the original feasibility region may or may not be included in the new feasibility region, determining which requires complicated analysis. It follows that the Min-Zero policy is near-optimal as it achieves the interior of the new feasibility region. Whether it can destabilize user traffic when the budget vector lies on the boundary depends on the distributions of arrivals and services, and requires more complicated analysis to determine.

Table 2. Convergence times of variants of Min-Zero on networks with different loads (rounded to the nearest integer).

Threshold \ Network Size	100	150	200	250	300	350	400	450	500
0	176	143	105	99	92	96	90	89	87
5	226	151	100	99	102	100	93	91	90
10	316	186	107	107	102	103	102	100	100
20	418	199	167	114	117	108	106	102	110
50	682	326	175	134	123	122	120	116	114
100	811	400	223	170	143	140	130	127	124
200	1493	550	423	275	206	171	151	144	141

Table 3. Convergence times of variants of Min-Zero on networks with different loads (rounded to the nearest integer).

Threshold \ Network Load	0.75	0.80	0.85	0.90	0.95
0	105	67	30	20	17
5	100	74	42	21	22
10	107	104	45	26	22
20	167	141	116	32	24
50	175	171	165	48	37
100	224	197	174	131	102
200	423	361	286	250	223

If we use $\mathbb{E}[\limsup_{t\to\infty} Q_n^u(t)] = \infty$ as the definition of instability, then similar phenomenon as in **Figure 8** would occur when we allow for non-stationary attack policies. Under this definition, our Min-Zero policy is near optimal over all stationary adversarial injection policies.

Therefore, we conclude that the two choices of instability criterion do not lead to significantly different results. Furthermore, the criterion we use in this paper, i.e., (1), has the implication that by achieving the goal, the adversary causes the users to lose a non-zero fraction of throughput. On the other hand, $\mathbb{E}[\lim_{t\to\infty} Q_n^u(t)] = \infty$ does not guarantee this. For example, when $Q^u(t) = \log t$, $\mathbb{E}[\lim_{t\to\infty} Q_n^u(t)] = \infty$ while the users only lose a vanishing fraction of (time-average) throughput as $\lim_{t\to\infty} \frac{\log t}{t} = 0$.

F SIMULATION RESULTS

In this section, we present simulation results on the convergence times of the Min-Zero policy with thresholds $\{0, 5, 10, 20, 50, 100, 200\}$ in **Tables 2 and 3**. From the results, we can see that the convergence time generally increases with the threshold.

Received August 2019; revised September 2019; accepted October 2019