Optimal Control for Networks with Unobservable Malicious Nodes

Bai Liu (MIT LIDS)
Eytan Modiano (MIT LIDS)

BACKGROUND

MODEL

ALGORITHM

ANALYSIS

SIMULATION

BACKGROUND

Motivation

Modern networks are increasingly complex

- Network dynamics can be **non-stationary and non-stochastic**
- Some nodes are **unobservable and uncontrollable**

Modern networks suffer from attacks

- Distributed Denial-of-Service (**DDoS**) attack: some nodes are hijacked and commanded to flood the network
- Structured Query Language (**SQL**) **injection** attack: malicious commands are injected into servers
- The hijacked nodes are also **unobservable and uncontrollable**, with the dynamics being **malicious**

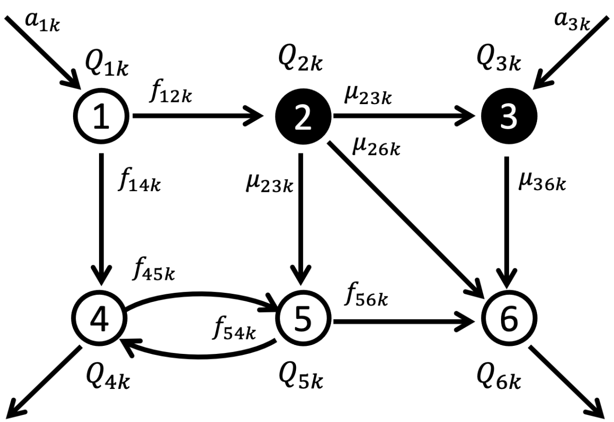
We aim to develop a control algorithm for networks that

- The external arrival process is **malicious**
- Some nodes execute **malicious** policies, and their states are **unobservable**
- **Malicious**: the adversary can **dynamically change the attack policy based on our actions** to maximize the damage

MODEL

Network Model

- Multi-hop network with N nodes (denoted by $\mathcal{N}$), K classes. $\mathcal{N}$ is partitioned into **accessible node set** $\mathcal{A}$ and **malicious node set** $\mathcal{M}$
- At the beginning of time slot t
  - A node i has $Q_{ik}(t)$ buffered packets of class k
  - Receives $a_{ik}(t)$ external packets (can be **malicious**)
- An accessible node $i \in \mathcal{A}$
  - The controller plans to transmit $f_{ijk}(t)$ packets to neighbor j
  - The **controller's policy** $\pi = \{f_{ijk}(t)\}_{0 \le t \le T-1}$ for $i \in \mathcal{A}$
- A malicious node $i \in \mathcal{M}$
  - The **adversary** plans to transmit $\mu_{ijk}(t)$ packets to neighbor j
  - We **cannot directly observe or control malicious nodes**
  - **Network event sequence** $\{\boldsymbol{a}(t), \boldsymbol{\mu}(t)\}_{0 \le t \le T-1}$: the actions taken by the adversary from time slot 0 to time horizon T



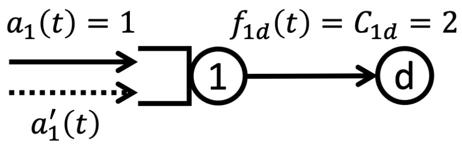
Maliciousness Metrics

A network event sequence $\{\boldsymbol{a}(t), \boldsymbol{\mu}(t)\}_{0 \le t \le T-1}$ is said to satisfy a constraint **if there exists a policy $\boldsymbol{\pi}$ such that the corresponding condition is satisfied** when the adversary implements the network event sequence.

Constraint	Condition
W **constraint** [Borodin, 1996]	$\sum_{i,k} Q_{ik}^{\pi}((n+1)W) \le \sum_{i,k} Q_{ik}^{\pi}(nW)$ for $n = 0, 1, \cdots$
V_T **constraint** [Liang, 2018]	$\max_{t \le T} \sum_{i,k} Q_{ik}^{\pi}(t) \le V_T$
Q_T **constraint** (This paper)	$\sum_{i,k} Q_{ik}^{\pi}(T) \le Q_T$

A network is said to have $W/V_T/Q_T$-constrained dynamics if all network event sequences generated by the adversary are $W/V_T/Q_T$-constrained, respectively.

Maliciousness Metrics



- A toy example where $a'_1(t)$ is malicious and
  - For $\frac{kT}{10} \leq t < \frac{kT}{10} + \frac{T}{20}$ with $k = 0, 1, \cdots, 9$, $a'_1(t) = 2$
  - For $\frac{kT}{10} + \frac{T}{20} \leq t < \frac{(k+1)T}{10}$ with $k = 0, 1, \cdots, 9$, $a'_1(t) = 0$
  - In other words, for each interval of length $\frac{T}{10}$, malicious arrival only exists during the first half interval
- For each interval $\frac{kT}{10} \leq t < \frac{(k+1)T}{10}$, the net increase of queue is zero, thus $W = \frac{T}{10}$
- The peak queue occurs at $t = \frac{kT}{10} + \frac{T}{20}$, which is $\frac{T}{20}$, thus $V_T = \frac{T}{20}$
- Since all packets are cleared at T, $Q_T = 0$

Maliciousness Metrics

Constraint	Requirement	Relationship
W constraint [Borodin, 1996]	Periodic patterns	$Q_T \leq V_T \leq c \cdot W$
V_T constraint [Liang, 2018]	Limited burstiness	
Q_T constraint (This paper)	None	

Since

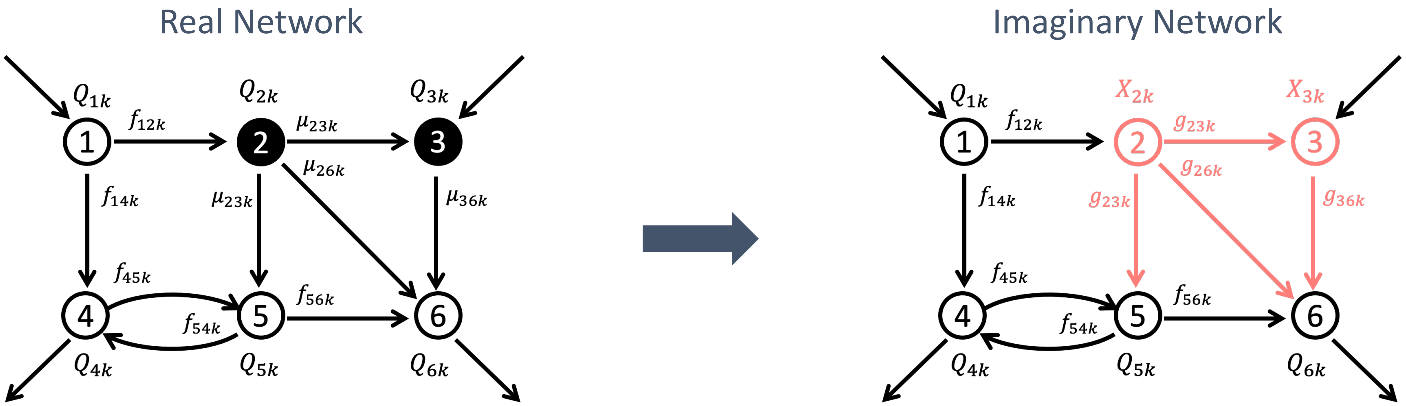
- Previous algorithms can stabilize the networks with $W = o(T)$ or $V_T = o(T)$
- Our algorithm can stabilize the networks with $Q_T = o(T)$ (proved later)
- $W = o(T)$ or $V_T = o(T)$ guarantees $Q_T = o(T)$, but not vice versa

We know that

- **Our algorithm can stabilize all stabilizable networks in previous works**
- **Some networks are not guaranteed to be stable under previous algorithms, but can be stabilized by our algorithm**

ALGORITHM

Overview



- Construct an **imaginary network** where every node is observable and controllable
- For a malicious node $i \in \mathcal{M}$ in the imaginary network, its queue and action may be different from the real network, and are denoted by X_{ik} and g_{ijk}, respectively
- The **imaginary network is easier to stabilize**. If we can also stabilize **the gap** $Y_{ik} \triangleq Q_{ik} - X_{ik}$ at the same time, the **real system is stabilized**

Overview

- Define a Lyapunov function

$$\Phi(t) = \underbrace{\sum_{i\in\mathcal{A},k} Q_{ik}^2(t)}_{} + \sum_{i\in\mathcal{M},k} X_{ik}^2(t) + \sum_{i\in\mathcal{M},k} Y_{ik}^2(t)$$

Accessible queues

Imaginary malicious queues

Gaps of the imaginary malicious queues

- We aim at minimizing the one-slot drift

$$\Delta\Phi(t) = \sum_{i\in\mathcal{A},k} Q_{ik}(t)\Delta Q_{ik}(t) + \sum_{i\in\mathcal{M},k} X_{ik}(t)\Delta X_{ik}(t) + \sum_{i\in\mathcal{M},k} Y_{ik}(t)\Delta Y_{ik}(t)$$

- However, $Y_{ik}(t)$ requires knowledge of $Q_{ik}(t)$ for $i \in \mathcal{M}$, which is unobservable
- For $i \in \mathcal{M}$, suppose we can estimate $Q_{ik}(t)$, but only inside a **sparse set of time slots** Γ_i
  - When $t \in \Gamma_i$, we obtain an estimate $\hat{Q}_{ik}(t)$ and estimate $Y_{ik}(t)$ as $\hat{Y}_{ik}(t) = \hat{Q}_{ik}(t) - X_{ik}(t)$. **We allow the estimates to be erroneous**
  - When $t \notin \Gamma_i$, we simply use the most recently updated $\hat{Y}_{ik}(t)$

MWUM (MaxWeight for Networks with Unobservable Malicious Nodes)

- At the beginning of time slot t, if $t \in \Gamma_i$, obtain an estimate $\hat{Q}_{ik}(t)$ and estimate $Y_{ik}(t)$ as

  $\hat{Y}_{ik}(t) = \hat{Q}_{ik}(t) - X_{ik}(t)$
- Solve

$$\boldsymbol{f}^{\boldsymbol{M}}(\boldsymbol{t}), \boldsymbol{g}^{\boldsymbol{M}}(\boldsymbol{t}) = \underset{0 \le f_{ijk}, g_{ijk} \le C_{ij}}{\operatorname{argmin}} \sum_{i \in \mathcal{A}, k} Q_{ik}(t) \left[\sum_{j \in \mathcal{A}} f_{jik} - \sum_{j \in \mathcal{N}} f_{ijk} \right] +$$
$$\sum_{i \in \mathcal{M}, k} X_{ik}(t) \left[\sum_{j \in \mathcal{A}} f_{jik} + \sum_{j \in \mathcal{M}} g_{jik} - \sum_{j \in \mathcal{N}} g_{ijk} \right] -$$
$$\sum_{i \in \mathcal{M}, k} \max\{\hat{Y}_{ik}(t), 0\} \cdot \left[\sum_{j \in \mathcal{M}} g_{jik} - \min\left\{ \sum_{j \in \mathcal{N}} g_{ijk}, X_{ik}(t) + a_{ik}(t) \right\} \right]$$

- Apply $\boldsymbol{f}^{\boldsymbol{M}}(\boldsymbol{t})$ to accessible nodes in the **real** network
- Apply both $\boldsymbol{f}^{\boldsymbol{M}}(\boldsymbol{t})$ and $\boldsymbol{g}^{\boldsymbol{M}}(\boldsymbol{t})$ to all nodes in the **imaginary** network

ANALYSIS

Stability

Theorem 1

Estimation error at t

If $Q_T = o(T)$, $\frac{\sum_{t=0}^{T-1} L(t)}{T} = o(T)$ and $|\epsilon_{ik}(t)| = o(t)$, we have $\lim_{T\to\infty} \frac{\sum_{i,k} Q_{ik}(T)}{T} = 0$, i.e., the network is rate stable, under MWUM.

Average delay in estimation

Corollary 1

The stability region of a given network is the set of network event sequences with $Q_T = o(T)$.

*If $Q_T = \Omega(T)$, there exists a network event sequence under which **no policy can stabilize** the network. If the adversary implements it, the network is not stabilizable. Meanwhile, when $Q_T = o(T)$, the network is stabilizable.*

Corollary 2

MUWM is throughput-optimal.

Robustness to Estimation Errors

Definition

A state-based algorithm determines control actions solely based on queue information.

MWUM , MaxWeight, BackPressure, reinforcement learning methods in network are all state-based.

Theorem 2

There exists a network with Q_T-constrained dynamics (where $Q_T = o(T)$) and $|\epsilon_{ik}(t)| = \Omega(t)$ such that **no state-based algorithm** can achieve rate stability.

Intuition: since the external arrival in each time slot is bounded, the queue of any node grows at most linear in time.

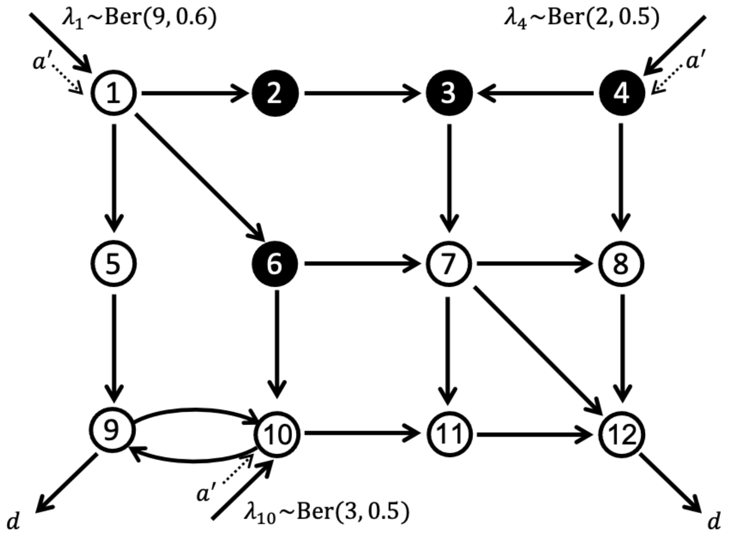
When $|\epsilon_{ik}(t)| = \Omega(t)$, the noise may completely mask the queue and thus hide the state information.

Since MWUM can stabilize the network when $|\epsilon_{ik}(t)| = o(t)$, MWUM is maximally robust.

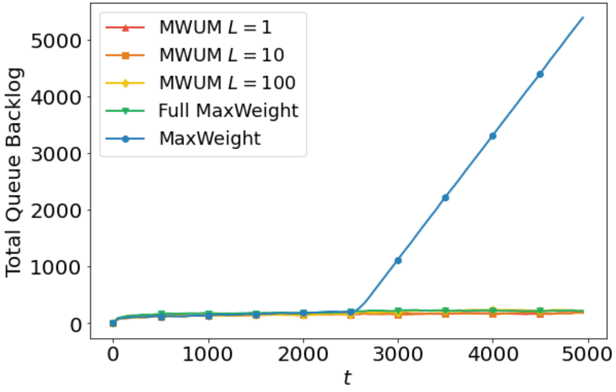
SIMULATION

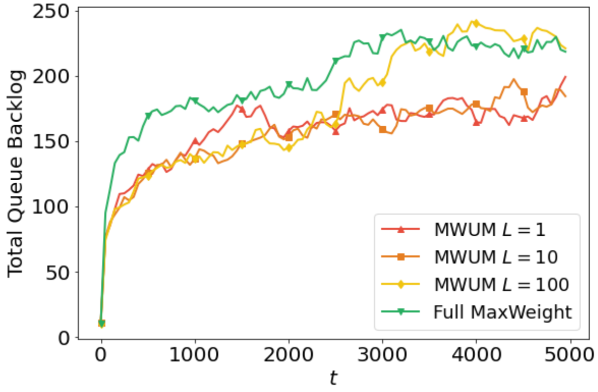
Model

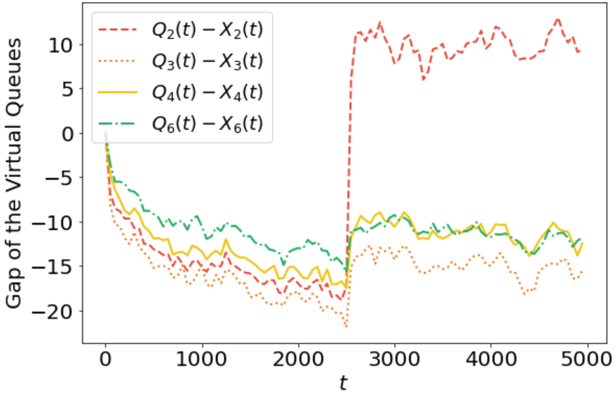
- All links have capacity $C = 5$
- The adversary tends to **allocate data to heavy loaded nodes**, since the nodes are closer to instability
- Malicious injection
  - $a' = 2$
  - Can inject into node 1, 4 or 10
  - Selects the node with the largest queue
- Malicious action
  - $\mu_{23}(t) = 5$ for $t \leq \frac{T}{2}$, and $\mu_{23}(t) = 1$ for $t > \frac{T}{2}$
  - $\mu_{37}(t) \equiv 5$
  - Node 4 and 6 apply the "join the longest queue" policy (in contrast to the JSQ policy)

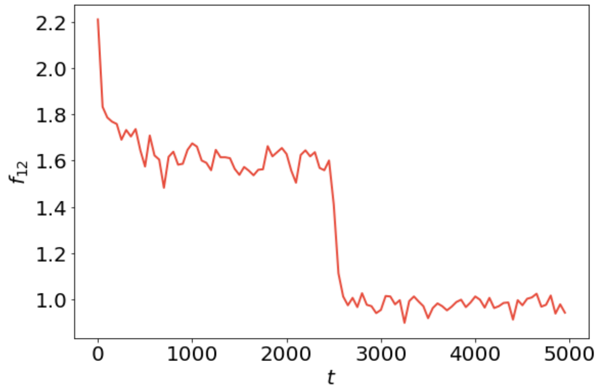


Numerical Results without Estimation Errors

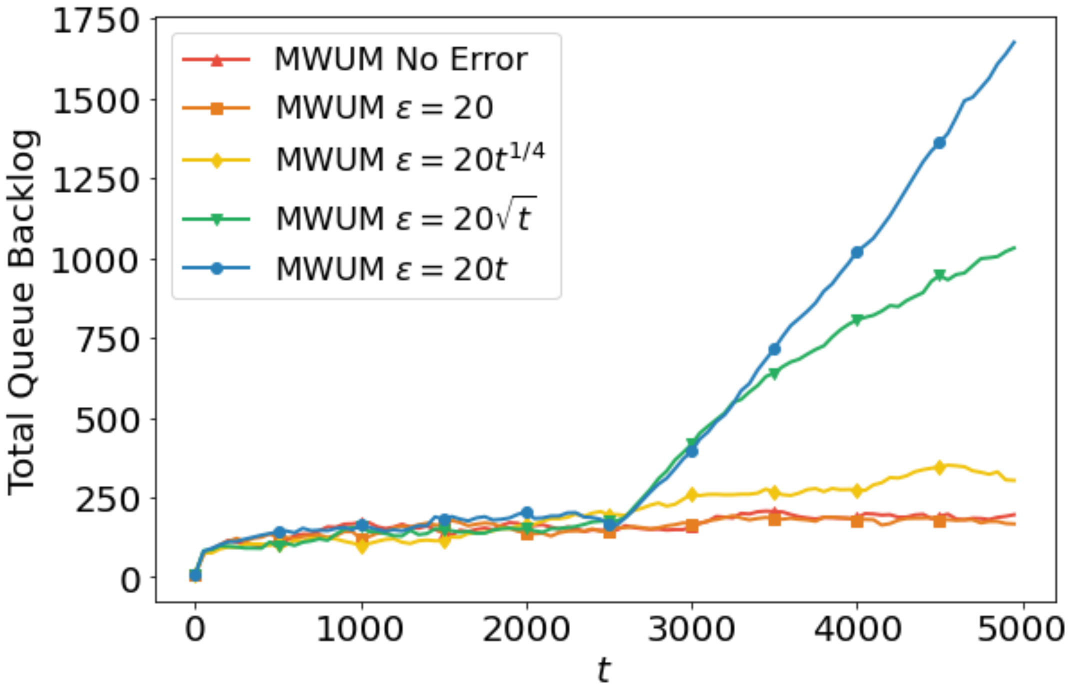








Numerical Results with Estimation Errors



Summary of Contributions

Modeling

- Propose a **new maliciousness metric** Q_T constraint
- Analyze the relationship between the Q_T constraint and the existing W constraint and V_T constraint
- Specify the stability region of networks with unobservable malicious nodes

Algorithm Design

- Existing relevant network control algorithms either require stochastic dynamics or full observability
- Develop the MWUM algorithm and rigorously show that MUWM is **throughput optimal**

Robustness Analysis

- Analyze the impact of estimation errors
- Show that MUWM is **maximally robust** to estimation errors