Queuing the Trust: Secure Backpressure Algorithm against Insider Threats in Wireless Networks

Zhuo Lu
University of Memphis

Yalin E. Sagduyu, **Jason H. Li**
Intelligent Automation Inc.

Outline

- Backpressure
- Vulnerabilities of Backpressure
- How to build virtual trust queue?
  - Design
  - Benefits
- Results
- Conclusions

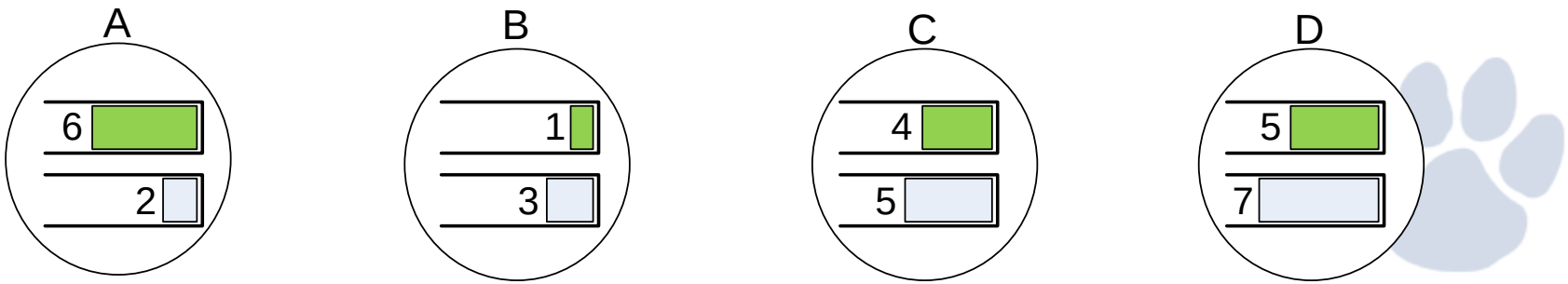
Backpressure Algorithm

- A widely-known routing and scheduling algorithm
  - Stabilizing all queues in the network and optimize the network throughput

- Idea in wireless networks
  - Maximize the sum of channel rates multiplying max queue backlog differences over all feasible link set.
    - A feasible link set is the set, in which all links do not collide with each other.

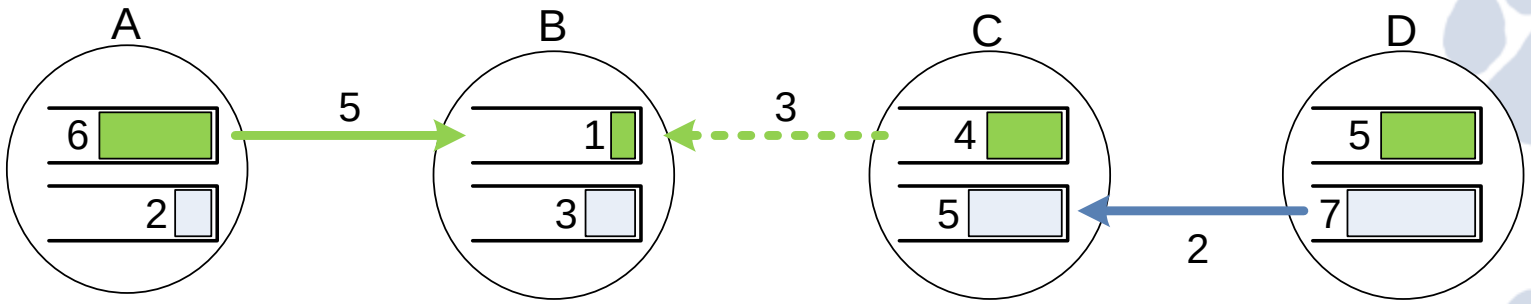
Example

- A wireless network on a line: nodes A, B, C, D
  - There are six links with the same link rate.
    - A→B, B→C, C→D, B→A, C→B, D→C
  - Feasible link sets: e.g., {A→B, D→C}
  - Two flows with different backlogs at each node.
    - green and gray colors



How Backpressure Works

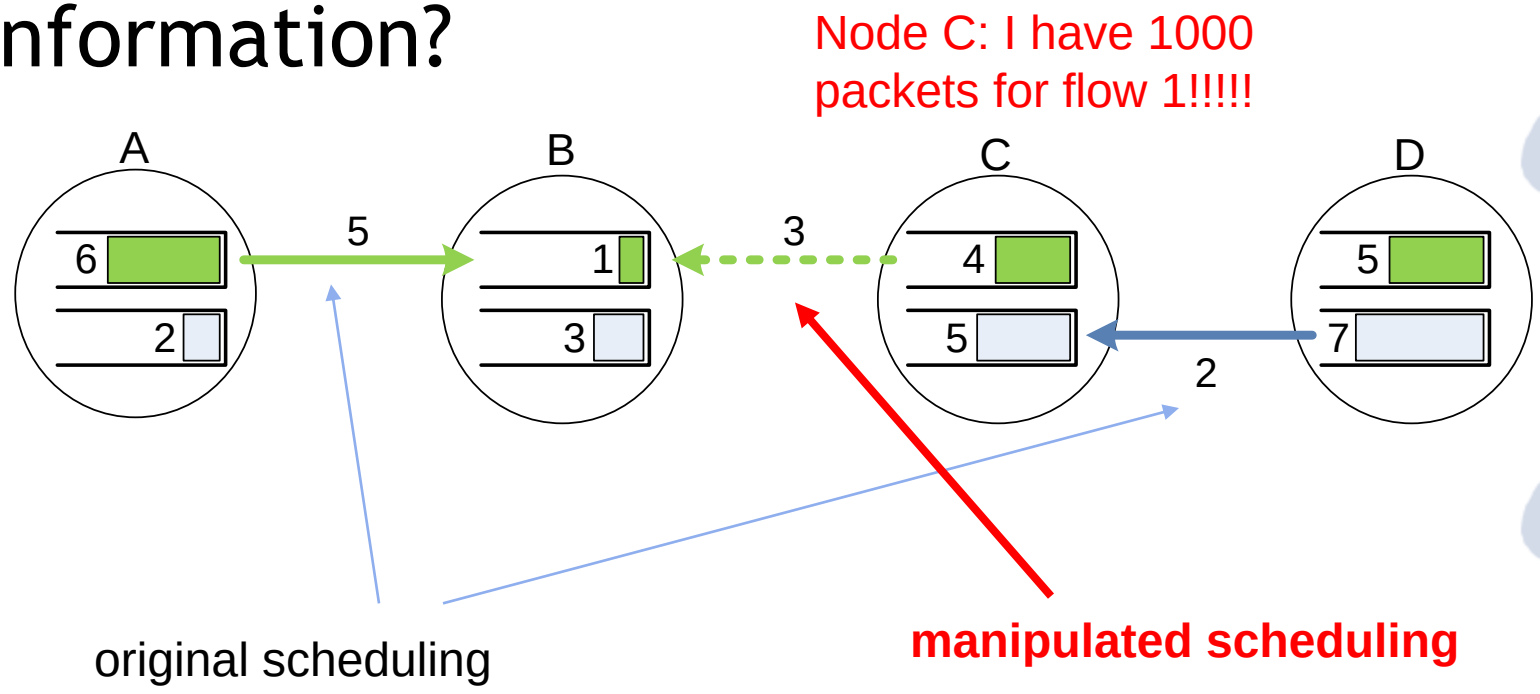
- Where to route packets? Steps:
  - Between two neighboring nodes:
    - Compute the backlog difference as a link weight for each flow. Choose the flow with maximum backlog difference



  - Maximize the sum of the weights of all chosen flows that do not collide with each other.
    - If we use {C→B}, the sum is 3.
    - If we use {A→B, D→C}, the sum is 5 + 2 = 7!

Vulnerability

- Backpressure needs
  - the backlog information inside a node!
- Issue: if node C wants to manipulate some information?



Examples of Insider Threats

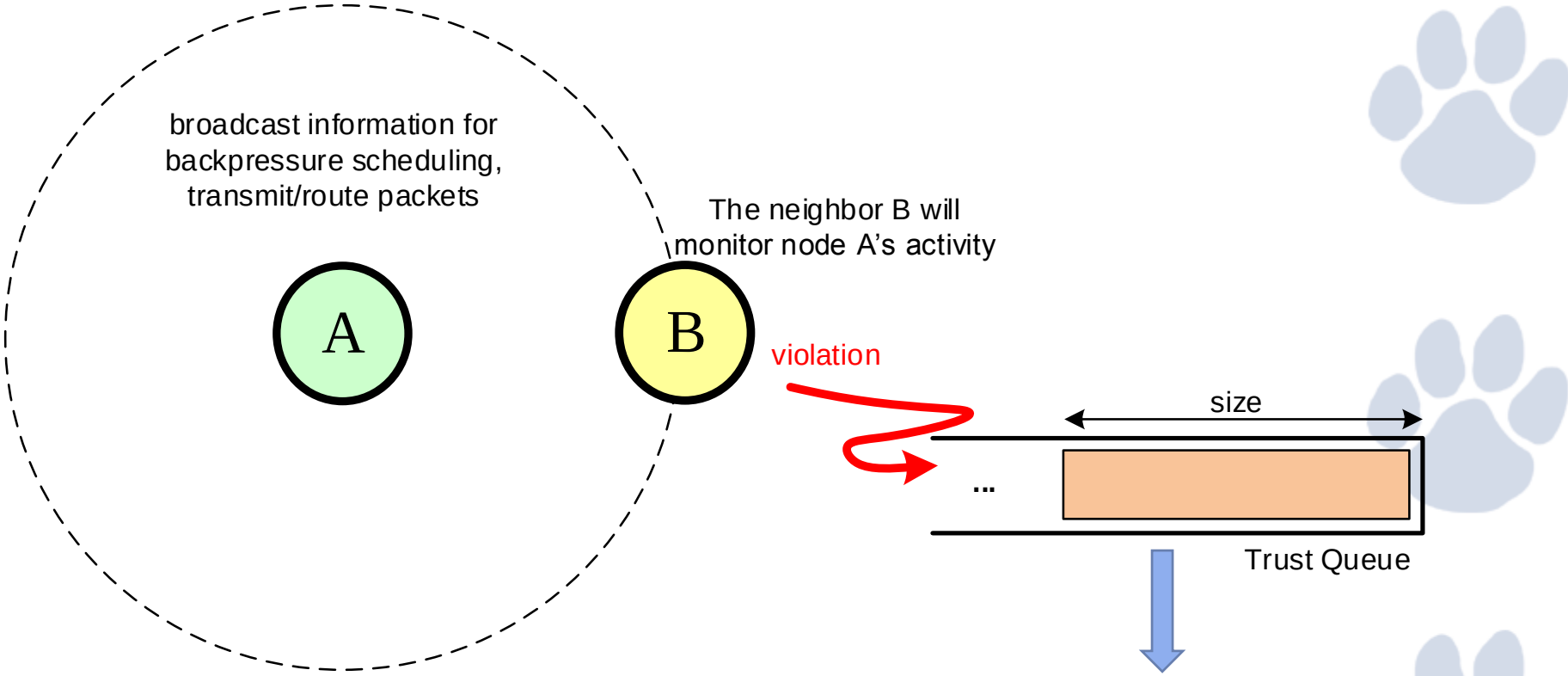
1. ***Blackhole attacks***, which always broadcast zero queue backlogs to attract packets to be routed to them, then drop all received packets.
2. ***Selective-forwarding attacks***, which keep relatively low profile compared with blackhole attacks. They do not falsify any information and obey the backpressure scheduling, but only drop packets routed to them for particular flows.
3. ***On-off attacks***, which act as blackholes or legitimate nodes during on and off periods.
4. ***Transmission-opportunity-wasting attacks***, which never falsify information, but simply abandon its scheduled transmission opportunity to degrade the network throughput.
5. ***Selfish nodes***, which always attempt to empty its queues by broadcasting high queue backlogs to capture the transmission opportunity.

… …

Our Objectives

- Objectives:
  - mitigate insider threats that aim to manipulate Backpressure as much as possible
  - while preserving optimal throughput objectives in Backpressure.
- Methodology:
  1) Neighborhood watch
  2) Define a virtual queue, queue the observation of my neighbor's activity.

The Main Idea



$$\mathbf{u}^*(t) = \underbrace{\arg\max_{\mathbf{u}(t)\in\mathcal{R}(t)} \sum_{u_{i,j}(t)\in\mathbf{u}(t)} (u_{i,j}(t)w_{i,j}(t)}_{\text{Original Backpressure Framework}} \underbrace{- X_{i,j}(t)D_{i,j}(t))}_{\text{what we add here}}$$

Theoretical Results

- The proposed optimization framework

$$\mathbf{u}^*(t) = \arg\max_{\mathbf{u}(t)\in\mathcal{R}(t)} \sum_{u_{i,j}(t)\in\mathbf{u}(t)} (u_{i,j}(t)w_{i,j}(t) - X_{i,j}(t)D_{i,j}(t))$$

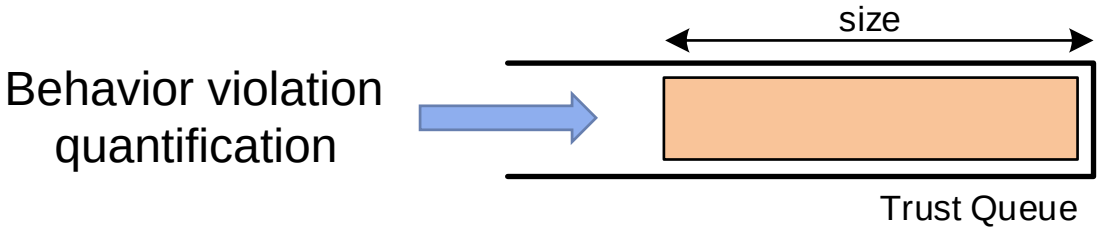
(details can be seen in the paper)

jointly stabilizes both data and virtual queues

- Zero penalty:
  - if there is no attack, there is no throughput degradation
- Impact Bounding:
  - if there are attacks, the virtual trust queue guarantees the attack's damage is always bounded from above.

How to Quantify the Violation

- Quantify according to attack category:
  - Information-falsification attacks
  - Protocol-violation attacks
- Enqueue each quantification into a virtual trust queue.

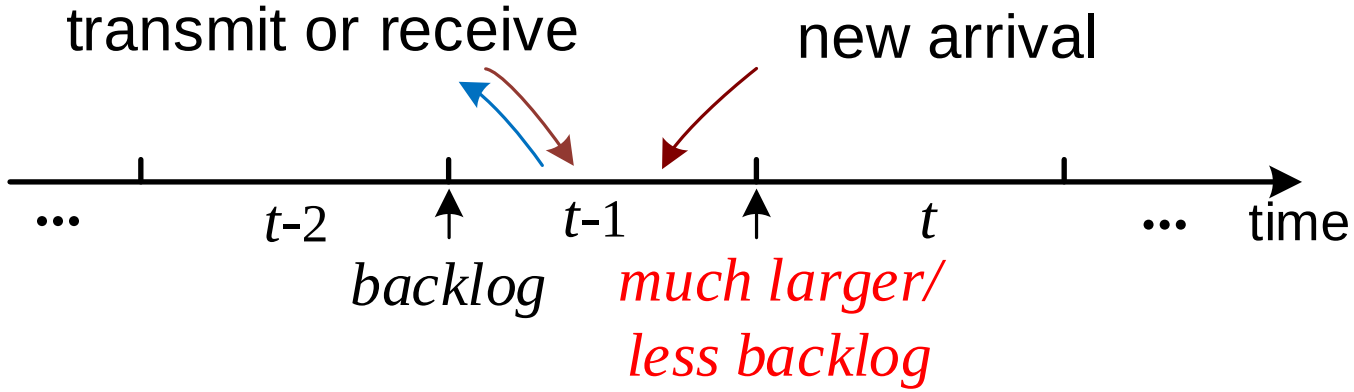


  - More precisely, it is an *"untrust queue"*.

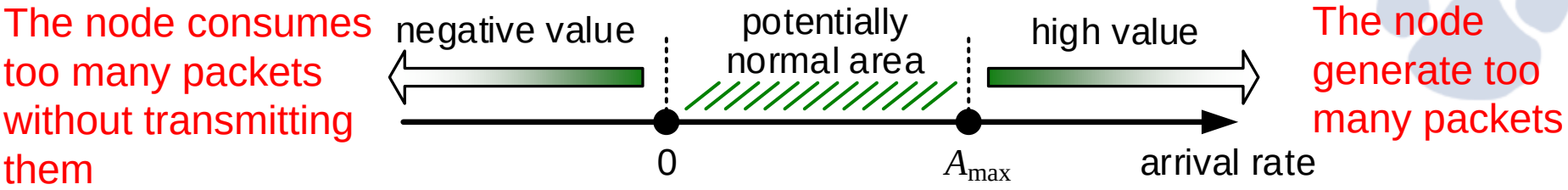
    (the larger the queue size, the less the trust)

Example

- Example: A node broadcasts inconsistent backlog information.

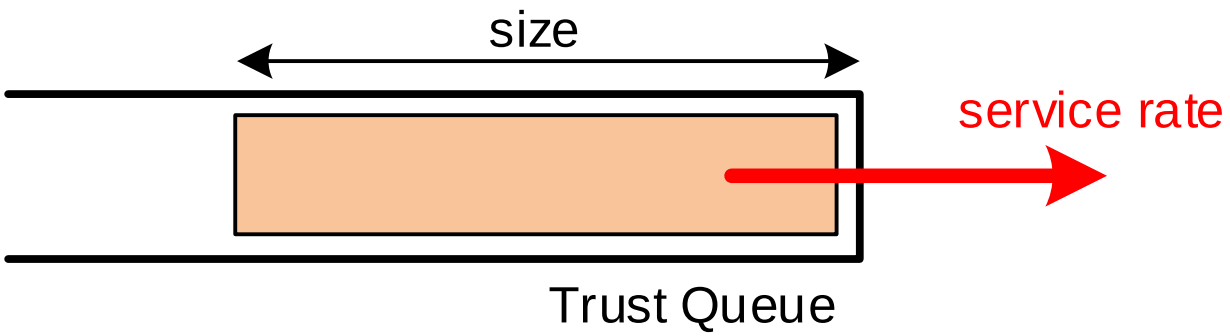


  - How to quantify? Estimate its new arrival rate.
  - An normal new arrival rate should be:



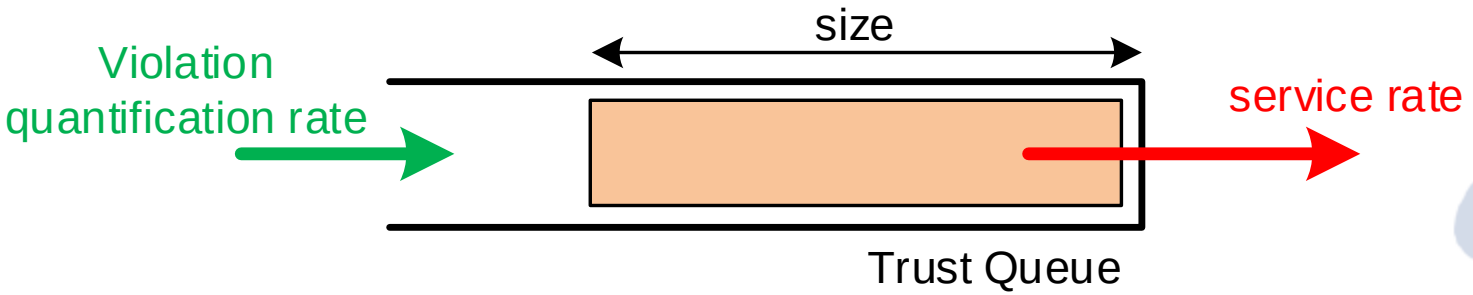
How to Define the Tolerance

- In wireless networks, observations from neighbor watch are not always perfect
  - We need to tolerate some errors
  - we set a small positive service rate.



Intuition behind Impact Bounding

- Average violation rate > service rate
  - Queue unstable: queue size → infinity

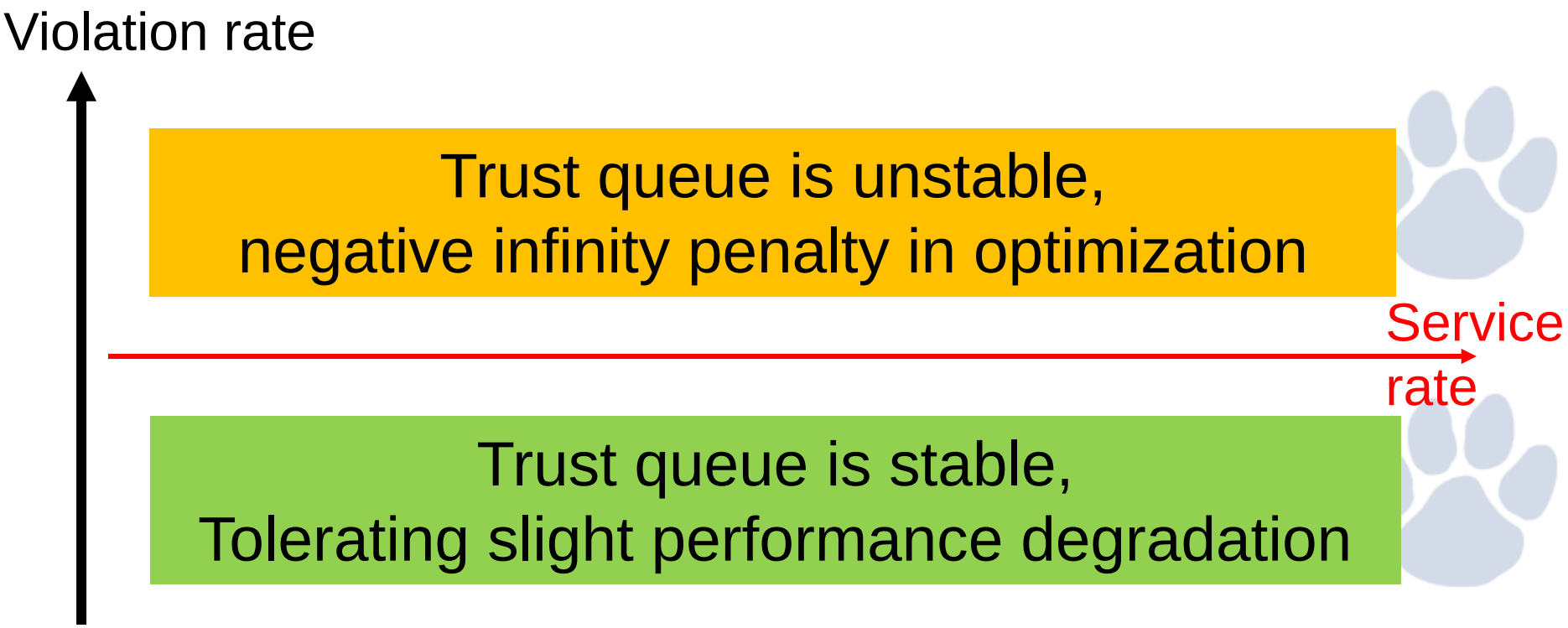


Queue size

$$\mathbf{u}^*(t) = \arg\max_{\mathbf{u}(t)\in\mathcal{R}(t)} \sum_{u_{i,j}(t)\in\mathbf{u}(t)} (u_{i,j}(t)w_{i,j}(t) - X_{i,j}(t)D_{i,j}(t))$$

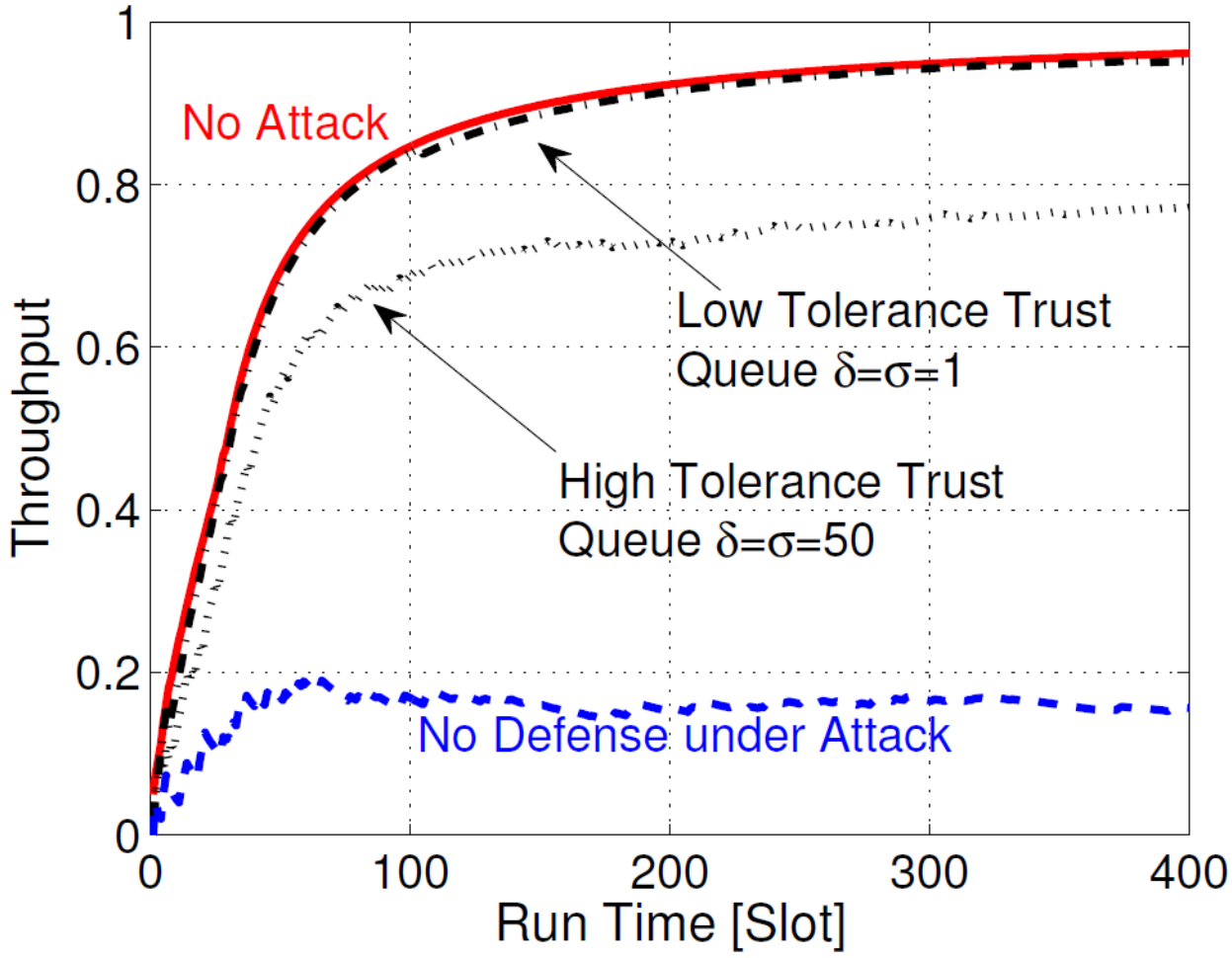
Impact Bounding (II)

- Service rate
  - Draw a clear line to tolerate or eliminate



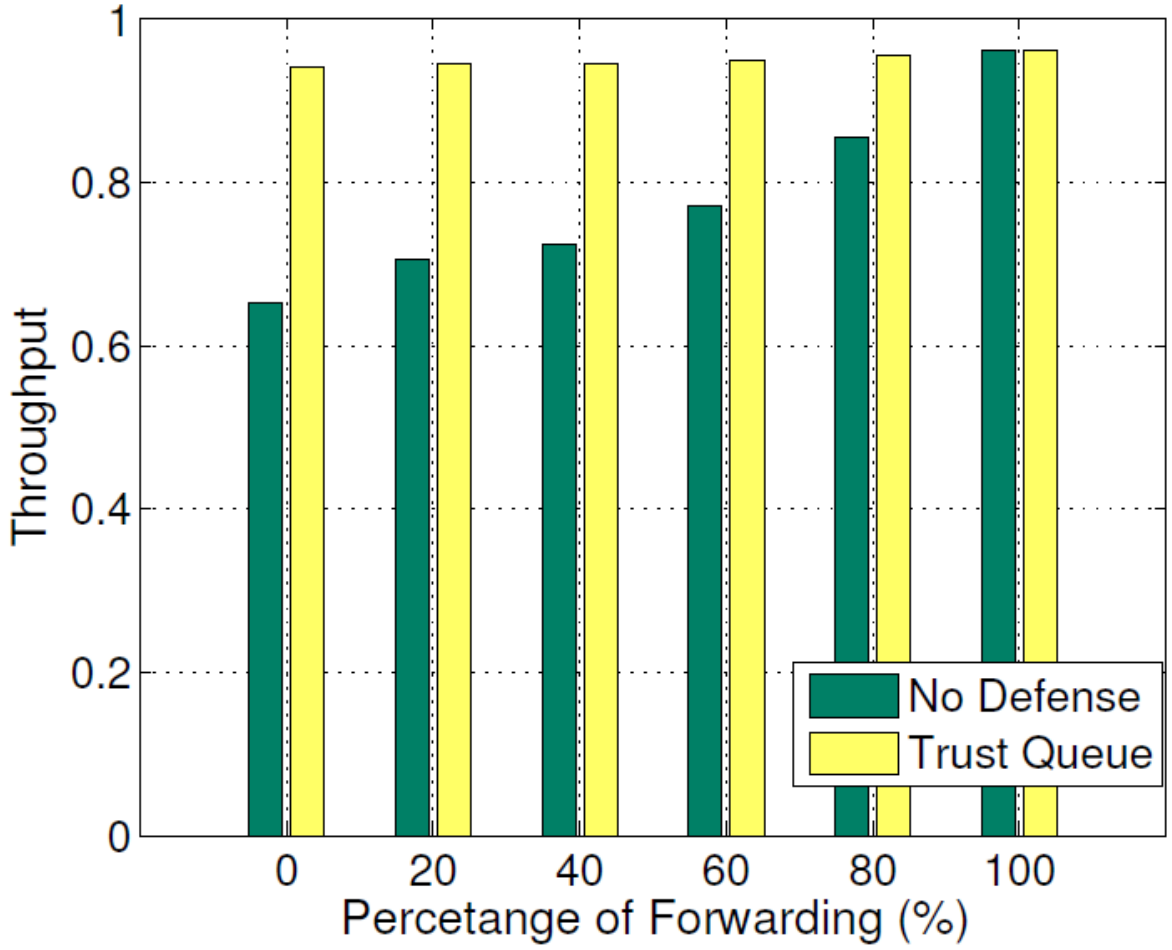
Result I

- Virtual trust queue under blackhole attacks



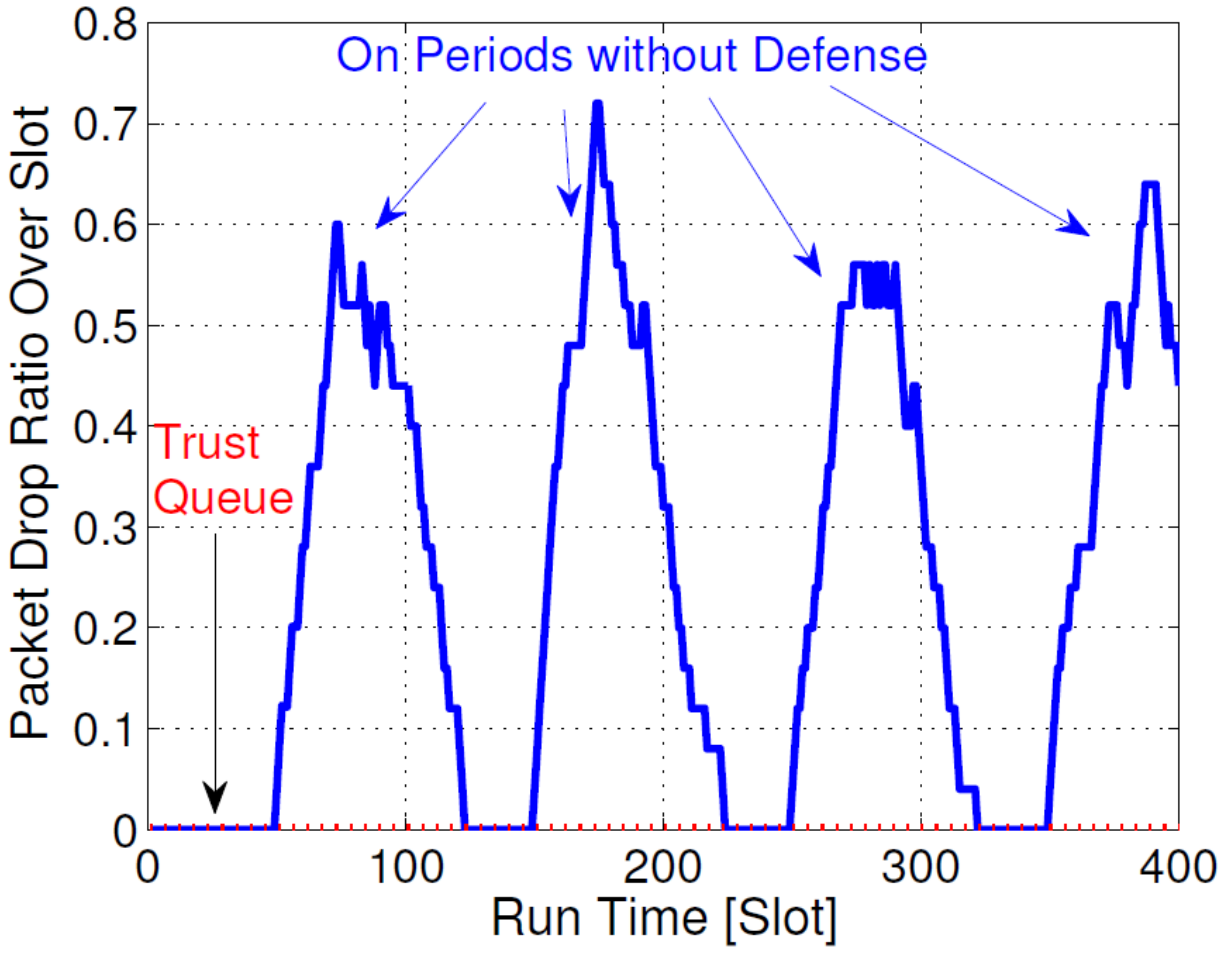
Result II

- Virtual trust queue under selective forwarding



Result III

- On-off attacks: on state – blackhole



Result IV

- Throughput performance under different attacks:
  1. Blackhole attacks
  2. Selective-forwarding attacks
  3. On-off attacks
  4. Transmission-opportunity-wasting attacks
  5. Selfish nodes

Attacks	4	5	4,5	3-5	2-5	1-5
Original Backpressure	0.23	0.85	0.21	0.15	0.09	0.0
Robust Backpressure	0.96	0.96	0.96	0.96	0.96	0.95

Conclusion

- We developed a generic secure backpressure framework based on virtual trust queuing.
  - neighborhood watch
  - quantifying any potential attack behavior
  - Jointly stabilizing data and trust queues.

- Virtual trust queue mechanism
  - can be easily integrated into an optimization framework with a formal way to build trust.

Thank you!

Q/A?