Structural Attacks on Local Routing in Payment Channel Networks

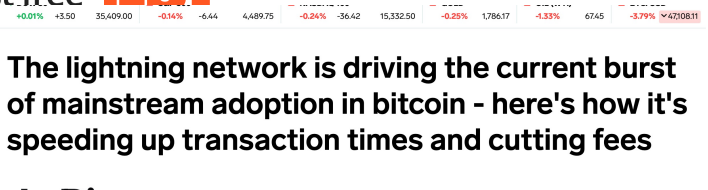
Ben Weintraub
Northeastern University

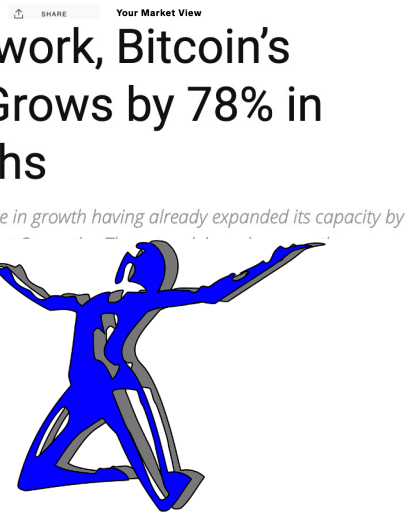
Cristina Nita-Rotaru
Northeastern University

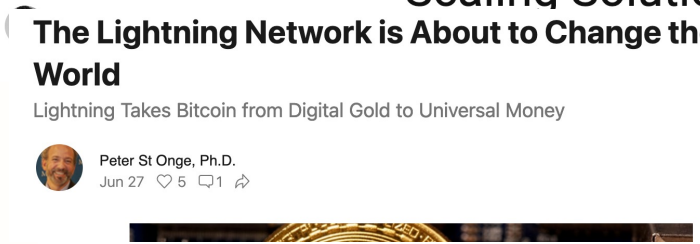
Stefanie Roos
TU Delft

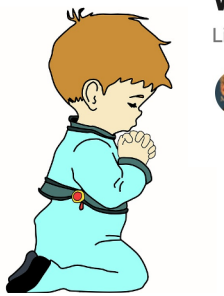
TU Delft

Payment channel networks solve blockchain's problems!









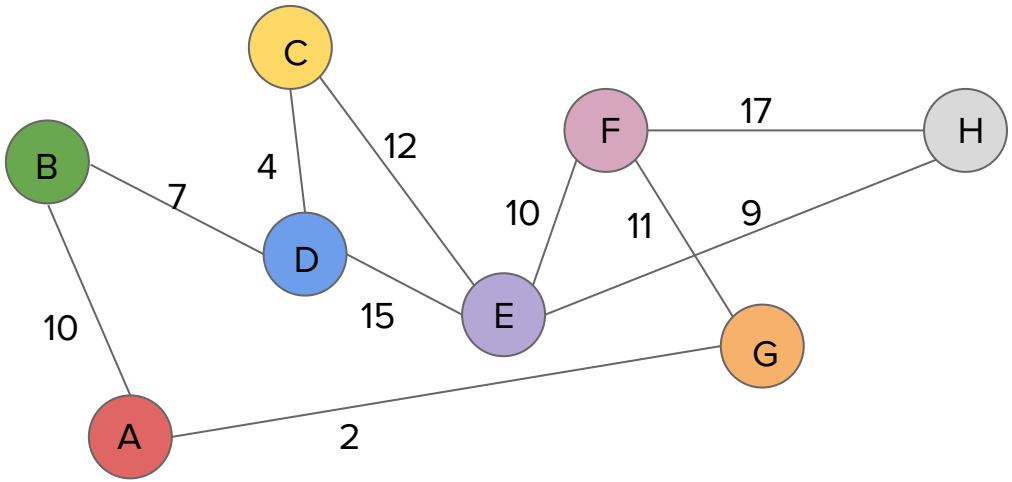
Payment channels

- Create channels with verified funds, this is public information
- Make payments by exchanging signed certificates containing new state
- Close channels and redeem funds by publishing latest state

Payment channel networks

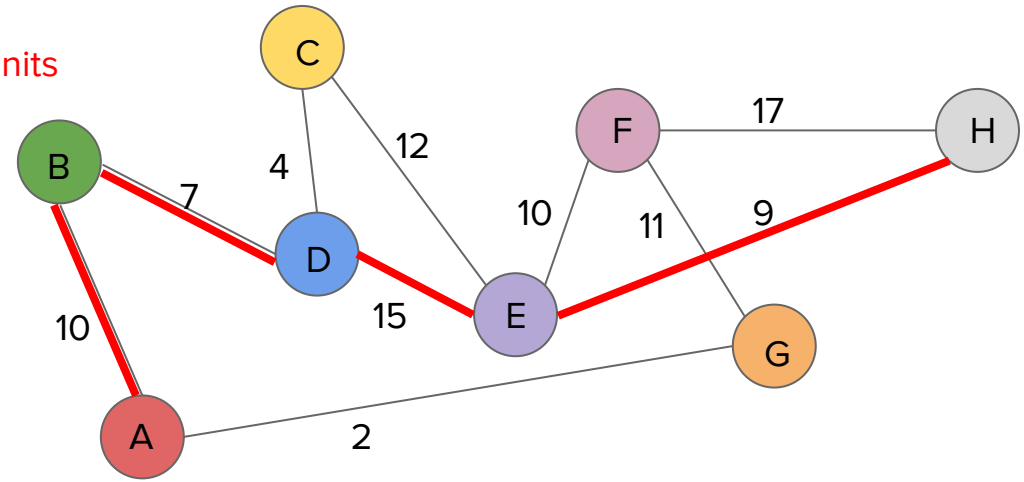
- Any participant can have any number of open channels
- Many connections can be made between pairs of participants forming a network

Nodes → participants
Edges → open channel
Edge weight → channel balance



Routing payments

- A path must be found where each link weight has enough credit to support the payment in question
- Challenges:
  - Many channels
  - Changing balances
  - Atomicity
- Example: A → H, 5 units



Global routing vs local routing

- Global routing
  - Sender decides entire path
  - Sender can have outdated knowledge of the topology
- Local routing
  - Next hop is chosen by the local node
  - Forwarding decision is always based on up-to-date information
  - No single node is aware of the entire path (not even the sender)
  - Overlay structure
    - Trees, meshes, etc.

Attacking PCNs

- Attacker incentives
  - Make money
  - Deanonymize participants
  - Destabilize financial instruments
- On- and off-path attackers
- Global routing requires onion encryption to hide full path
  - Vulnerable to malicious path selection (e.g. loops)
- **Structural attacks** are focused on attacker *placement* in the topology
  - Also depends on the topology itself

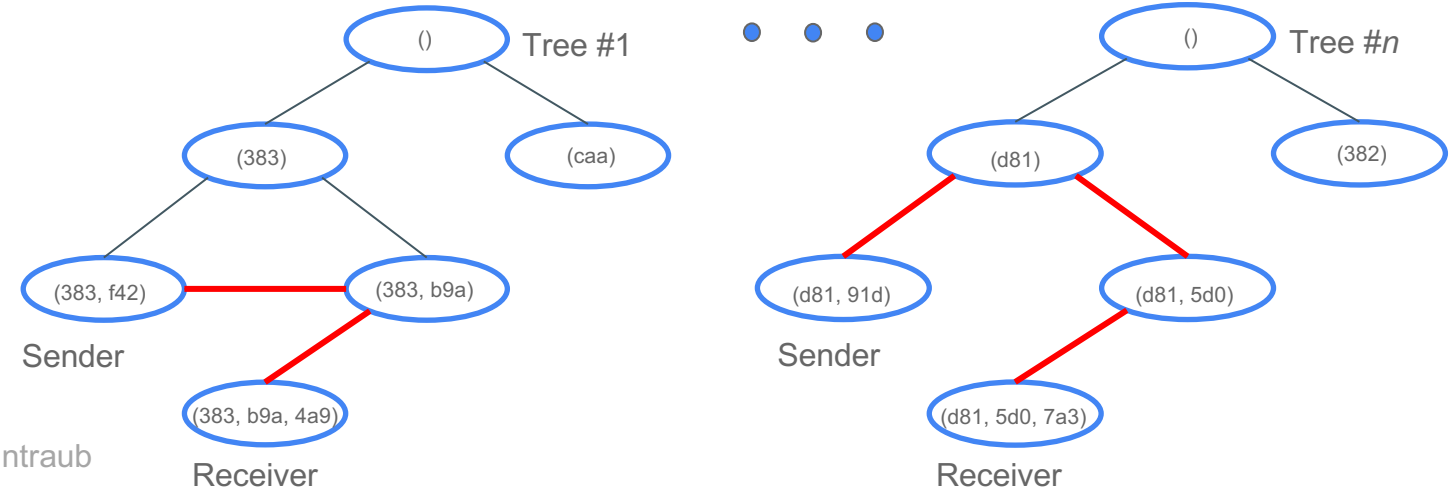
This talk

Are PCNs using local routing vulnerable to structural attacks?

- Evaluate the effectiveness of existing attacks as a function of attacker placement
- Propose countermeasures and model their effectiveness
- For concreteness, we focus on the effect of **payment griefing** on the **SpeedyMurmurs** routing algorithm (Roos et al. 2018)

SpeedyMurmurs (Roos et al. 2018)

- Spanning tree-based overlay
- Forwarding based on
  - Direct knowledge of channel balances
  - Distance from recipient
- Payment shares routed in parallel on *n* trees

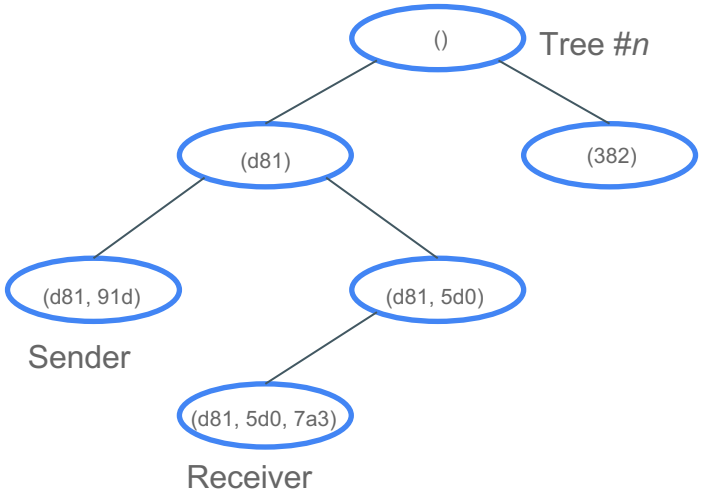


Threat model

- Attacker can corrupt arbitrary, formerly honest, nodes
- Attacker can collude out-of-band, *i.e.*, attacker information is public
- Attacker knowledge
  - Topology
  - *Initial* channel balances
  - Routing algorithm
- Non-adaptive

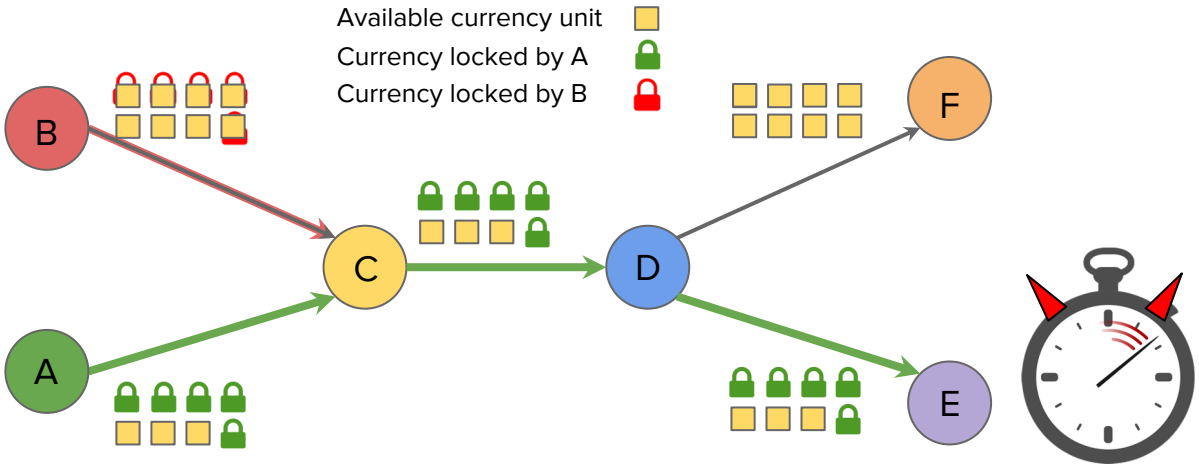
Knowledge of tree depth

- Every node knows its parent's coordinate
- The length of that coordinate leaks its tree depth



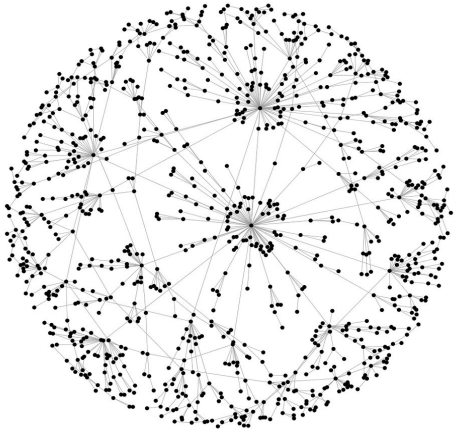
Payment griefing attack

- B → F (5 units) cannot find a route because no path has enough free funds
- Can even delay until payment times out, thus voiding the payment and costing the attacker nothing
- Forwarding nodes who are participating to collect transaction fees also lose out



Simulation and datasets

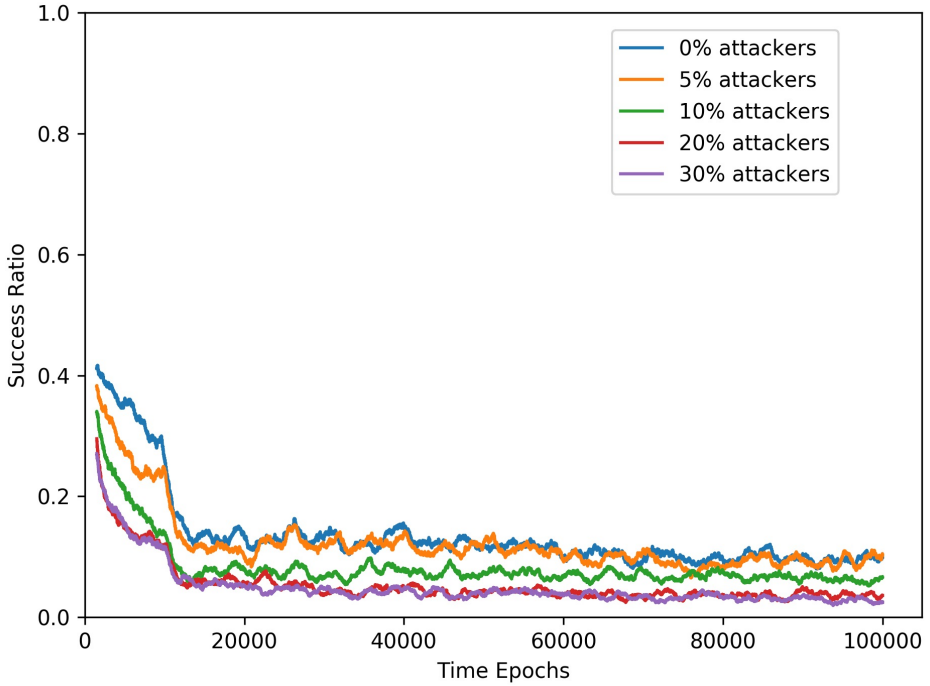
- Extended the simulator from Roos *et al.*
- Generated datasets representing network graphs and transactions
- Scale-free topology (Rohrer *et al.* 2019), 10k nodes (https://1ml.com/statistics)
- 100k transactions
  - Transaction senders/recipients sampled from a Poisson distribution
  - Transaction values sampled from a Pareto distribution
- Assigned channel balanced by “routing in reverse”



A scale-free topology
Courtesy of Simon Cockell, Flickr

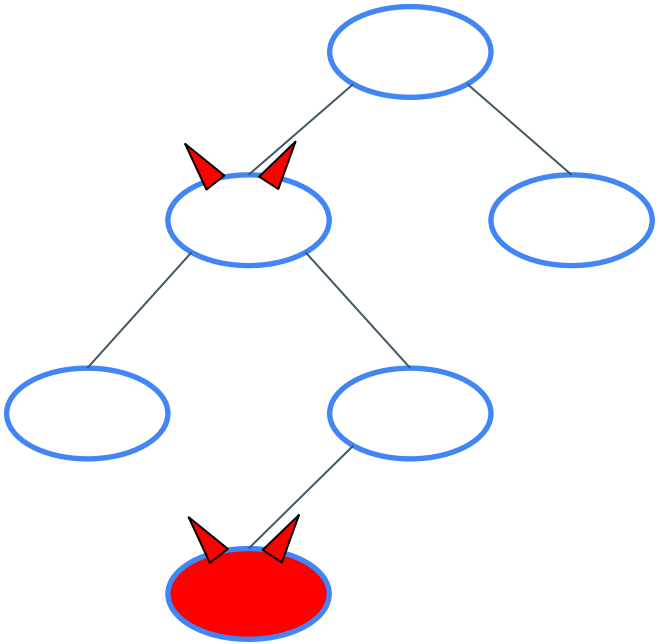
Baseline attacks

- Simulations on a synthetic network
  - Scale-free graph
  - 10k nodes
  - 100k transactions
- Griefing attack
- SpeedyMurmurs
- Attackers are randomly distributed



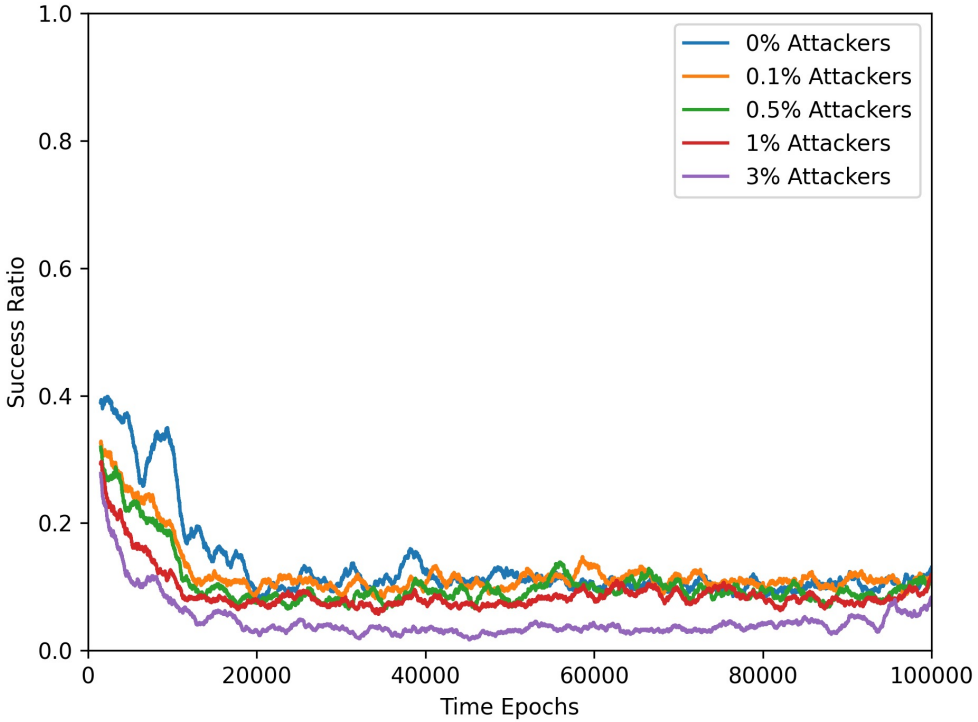
Exploiting the tree structure

- Random selection may choose a poorly placed node
- Roots and nodes close to the roots see more transactions



Tree-based attack

- Simulations on a synthetic network
  - Scalefree graph
  - 10k nodes
  - 100k transactions
- Griefing attack
- SpeedyMurmurs
- Select attackers based on tree depth



Betweenness centrality

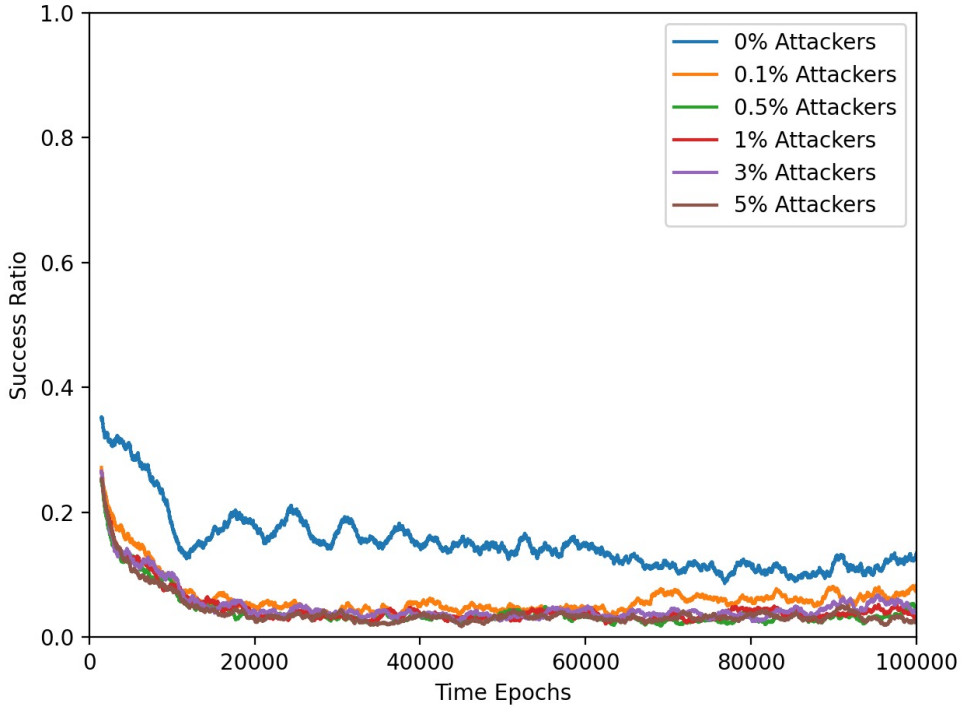
- The centrality of node, *n*, is

$$c_b(n) = \sum_{s,r,n \in N} \frac{\sigma_{srn}}{\sigma_{sr}}$$

- Where:
  - *N* is the set of nodes in the graph
  - σ_{sr} is the number of shortest paths between *s* and *r*
  - σ_{srn} is the number of shortest paths between *s* and *r that include n*

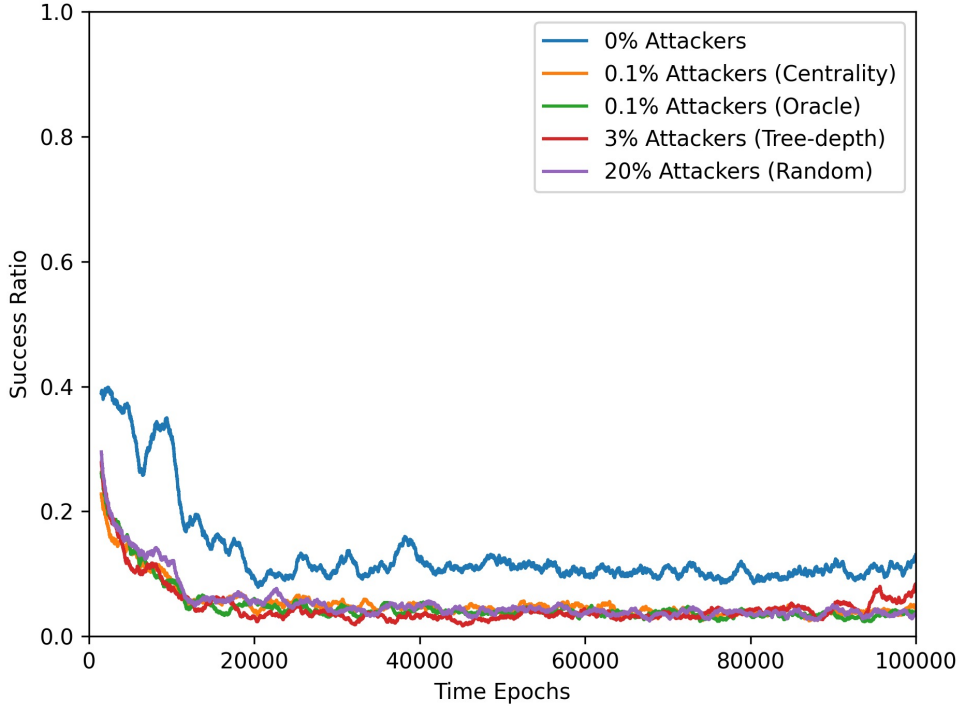
Centrality-based attack

- Simulations on a synthetic network
  - Scalefree graph
  - 10k nodes
  - 100k transactions
- Griefing attack
- SpeedyMurmurs
- Attackers are selected by the highest betweenness centrality



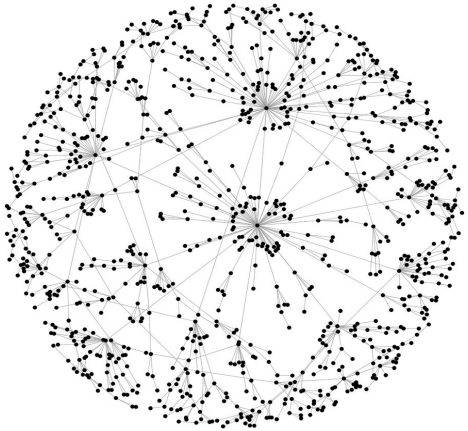
Attack comparison

- The centrality-based attack reaches maximum effectiveness as quickly as an oracle

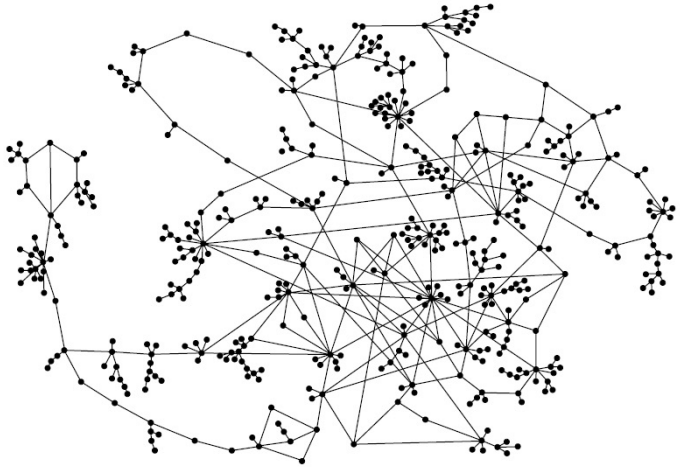


Countermeasures

- Networks are susceptible due to centralization
- We attempt to mitigate by creating a less centralized network



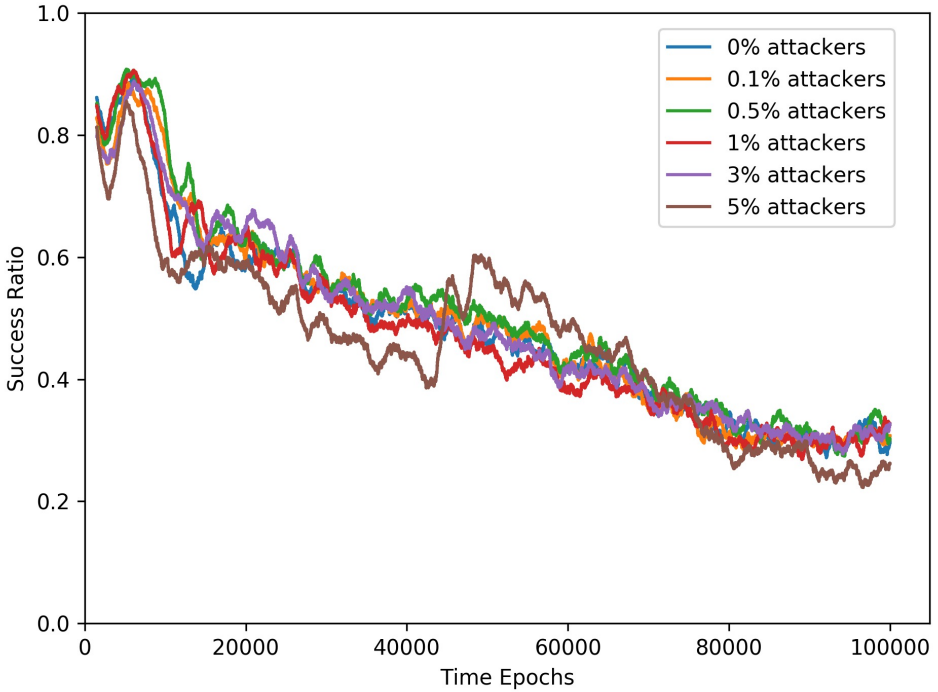
Scale-free



Small-world

Attack on small-world

- Generated a Newman-Watts-Strogatz network with
  - 10k nodes
  - 100k transactions
- Again, attackers are selected by betweenness centrality



Future work

- Dynamic routing algorithms in PCNs are insufficiently understood despite growing importance
- Certain network structures are less vulnerable, incentives necessary to generate those structures is not well understood

Summary

- Blockchains have problems: throughput, efficiency, privacy
- Off-chain transactions through payment channels solve many issues
- Payment channels are vulnerable to griefing
- Location, location, location
- Centrality is expensive
- Small-world networks may be powerful defense

Contact: weintraub.b@northeastern.edu