Know-Thy-Neighbor

Approximate Proof-of-Location

Sabrina Kall

EPFL ICC

June 6, 2019

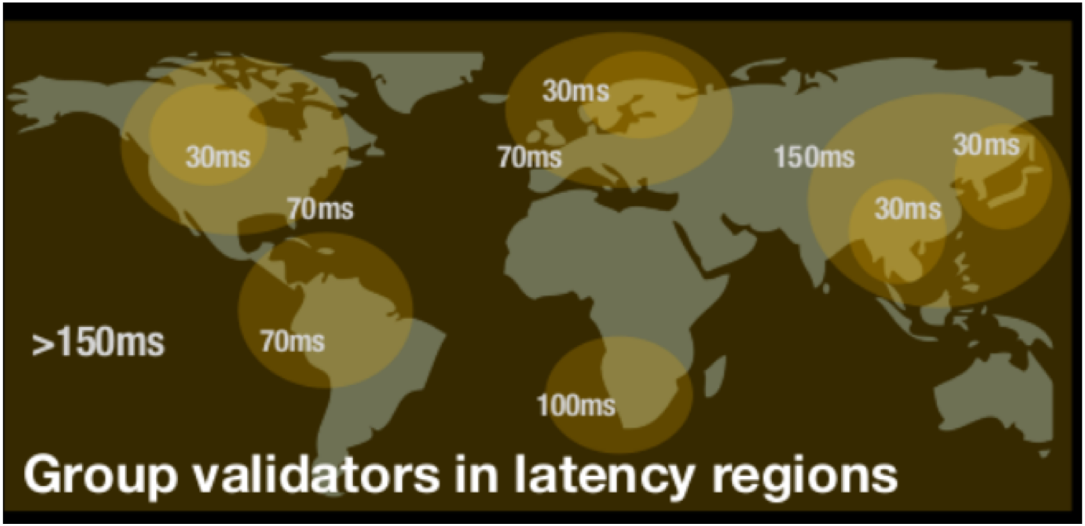
Responsible
Prof. Bryan Ford
EPFL / DEDIS

Supervisors
Cristina Basescu,
Kelong Cong
EPFL / DEDIS

Abstract

Imagine you're at a coffee shop…

«Trust-but-Verify »



Nyle’s Goals

- Goal: Validate transactions *fast*
- How : Use only close validators
- Problem: Finding close validators (« regions »)

Our Goals

- Goals:
  - Find close validators (efficiently)
  - Exclude (most) malicious validators
  - Do not exclude (any) honest validators
- How :
  - Secure Latency Measurement Protocol
  - Blacklisting Algorithm

Finding latencies

- Goal: Finding close validators
- How : Ping ?

Nope – Ping is not enough !

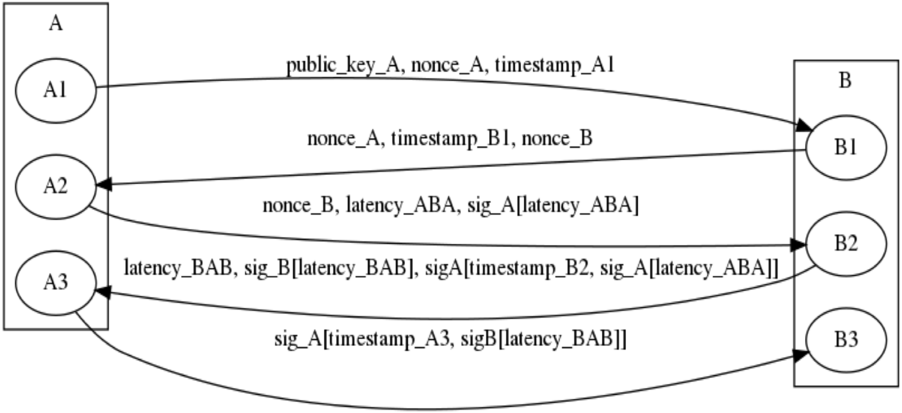
- MITM Attacks
- Replay Attacks
- Malicious nodes lying about results
- Etc...

Secured Latency

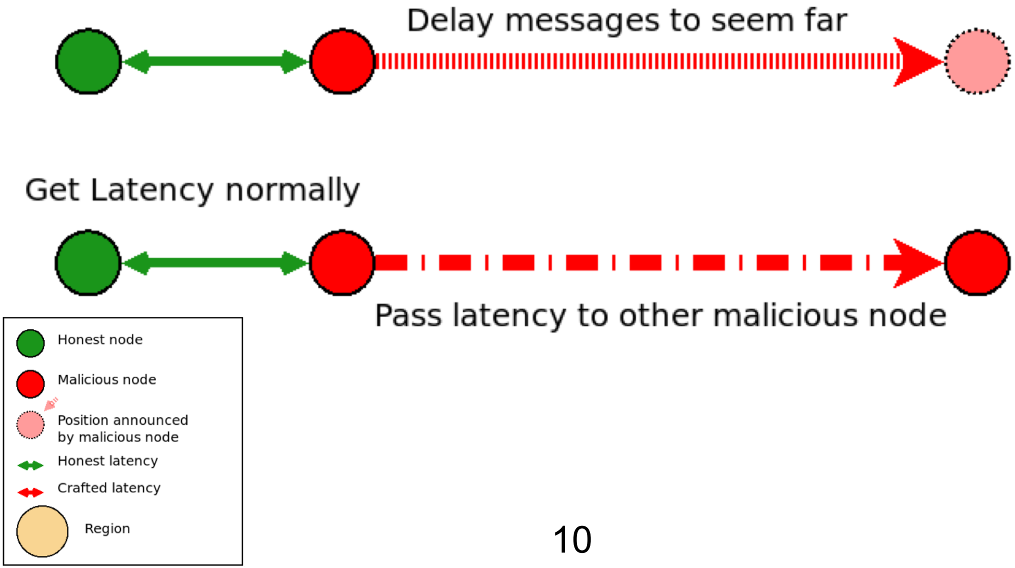
- What node A writes in blockchain for B:
- **sig B[timestamp B, sig A[latency ABA]]**

- What node B writes in blockchain for A :
- **sig A[timestamp A, sig B[latency BAB]]**

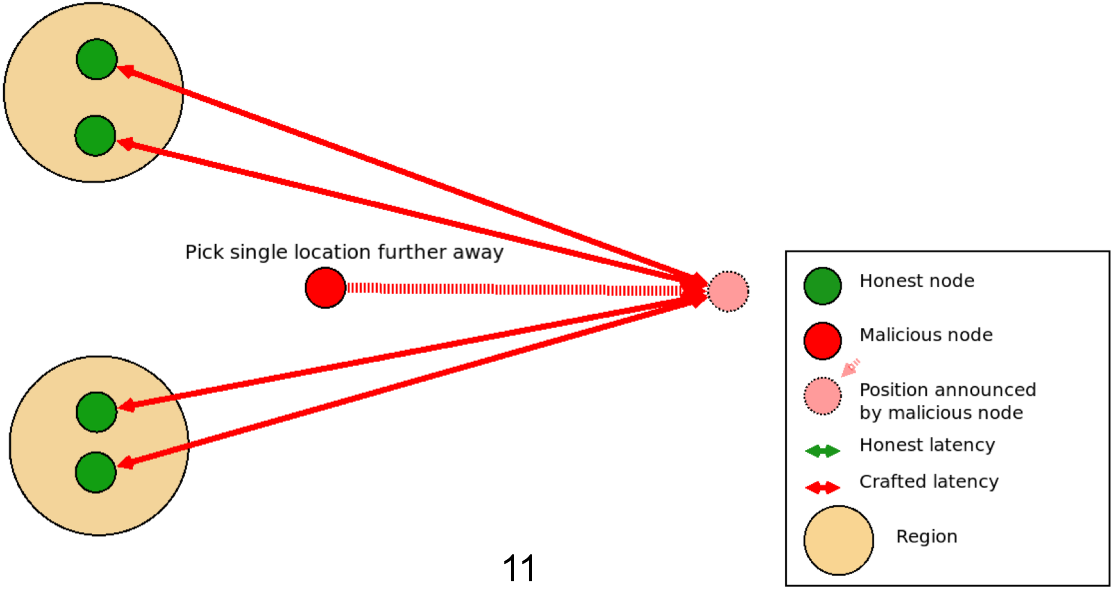
Secure messaging protocol


A
A1
A2
A3
B
B1
B2
B3
public_key_A, nonce_A, timestamp_A1
nonce_A, timestamp_B1, nonce_B
nonce_B, latency_ABA, sig_A[latency_ABA]
latency_BAB, sig_B[latency_BAB], sigA[timestamp_B2, sig_A[latency_ABA]]
sig_A[timestamp_A3, sigB[latency_BAB]]

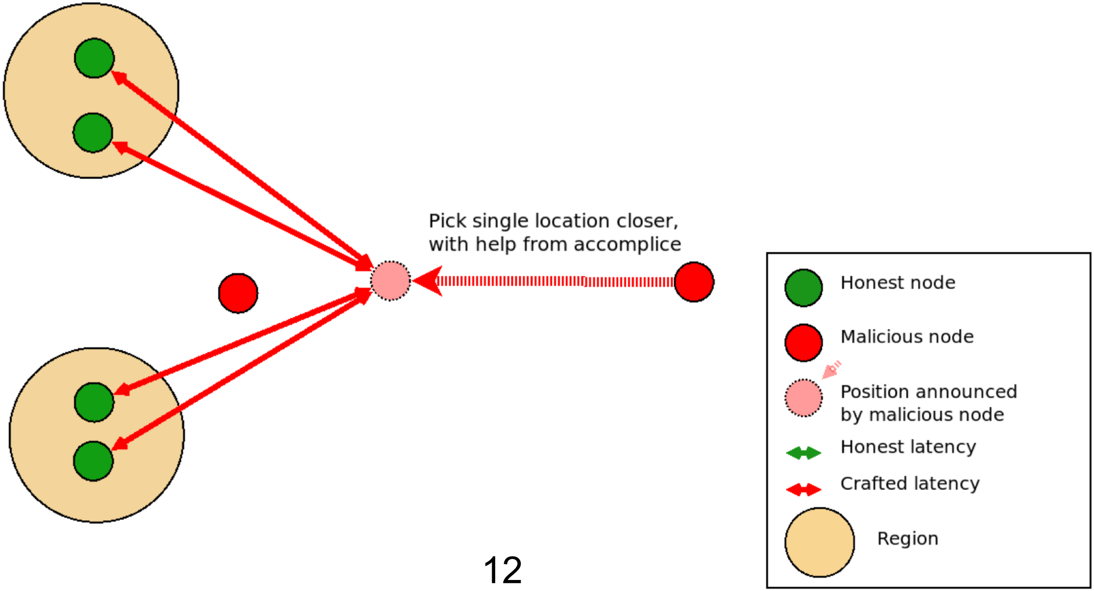
Remaining cases to handle


Delay messages to seem far
Get Latency normally
Pass latency to other malicious node
Honest node
Malicious node
Position announced by malicious node
Honest latency
Crafted latency
Region

Case 1 : Moving away

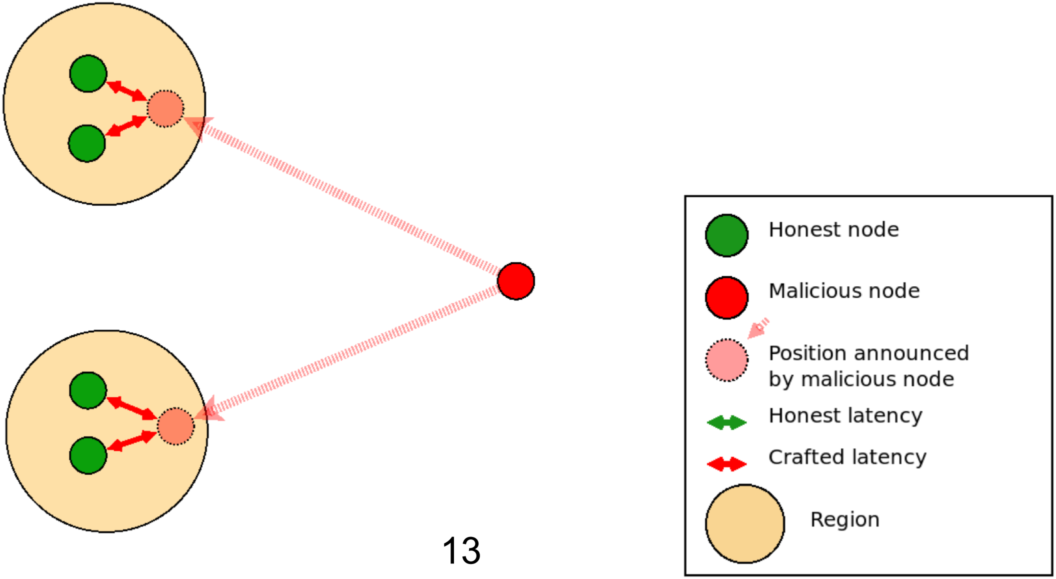

Pick single location further away
Honest node
Malicious node
Position announced by malicious node
Honest latency
Crafted latency
Region

Case 2 : Moving closer



Pick single location closer, with help from accomplice

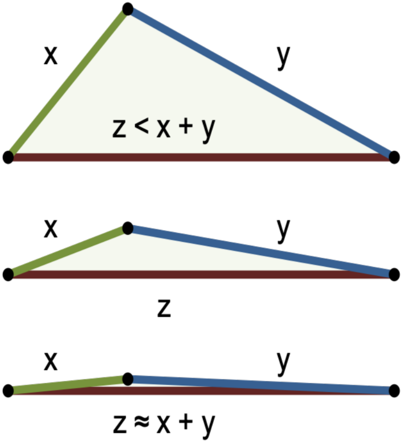
Case 3 : Moving to multiple locations



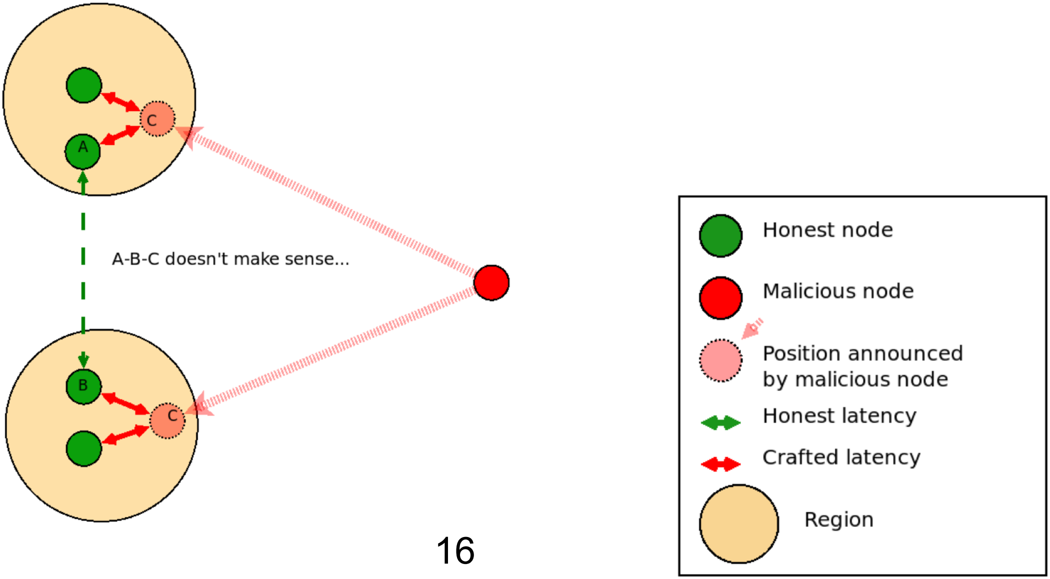
Blacklisting Algorithm

- Goal: Exclude dishonest nodes (but not honest nodes !)
- How : Publish latencies and use a blacklisting algorithm to find dishonest nodes

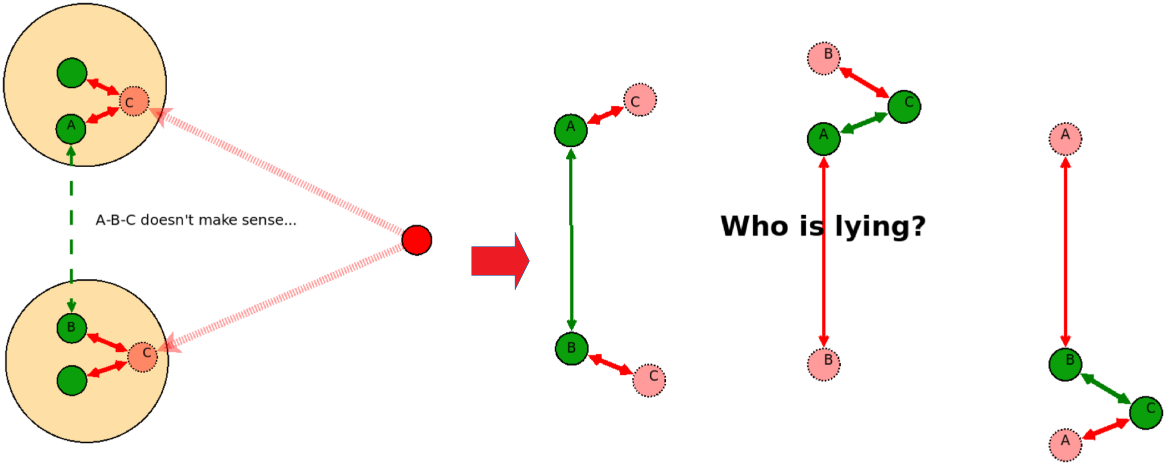
The Basics : Triangle Inequality



Detecting malicious nodes



Detecting malicious nodes



A-B-C doesn't make sense...

Who is lying?

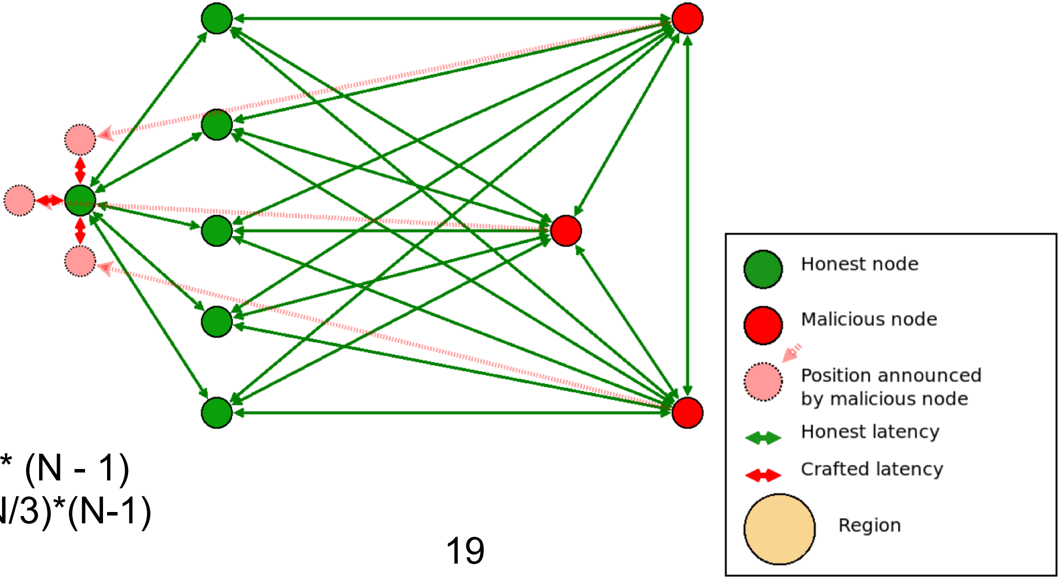
Basic Blacklisting

1) Find all triangle inequality violations

2) Remove nodes involved in too many TI violations

Choice of threshold (Worst case scenario)

N = 9
L = 3

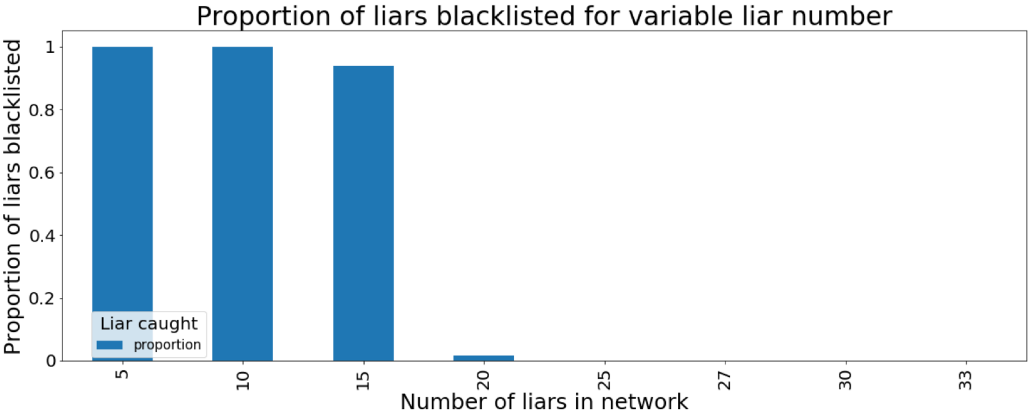


θ = L * (N - 1)
   = (N/3)*(N-1)

Basic Blacklisting

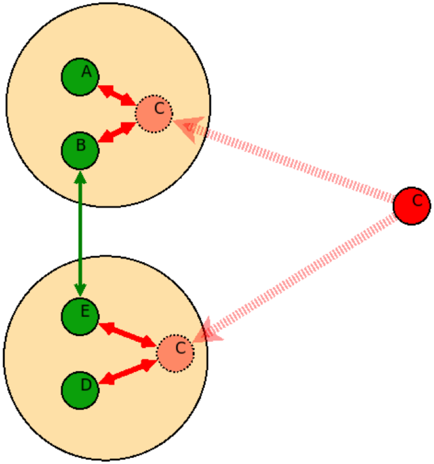
- Pros :
  - We don’t blacklist honest nodes
- Cons :
  - We end up not blacklisting very many malicious nodes

Basic Blacklisting



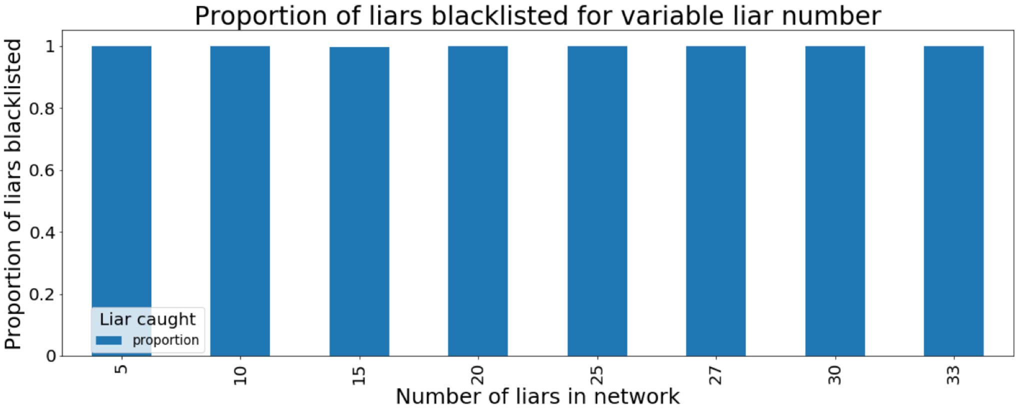
Enhanced Blacklisting

- Who blames who ?
- Blacklist nodes with many accusers



TI Violations
A-**C**-D
A-**C**-E
B-**C**-D
B-**C**-E

Enhanced Blacklisting



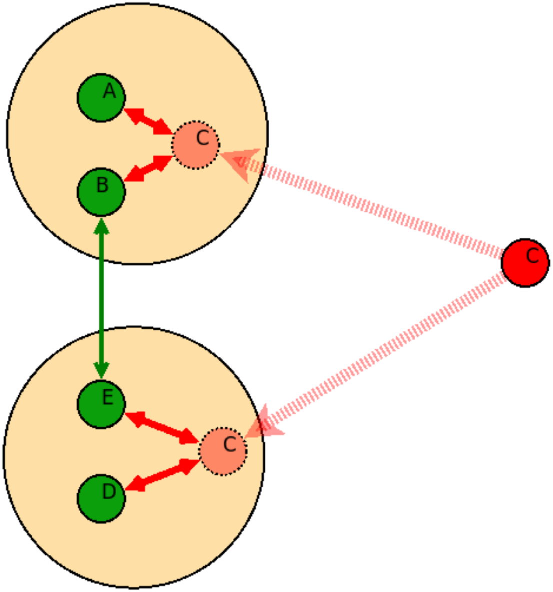
Enhanced Blacklisting

- Pros :
  - Still don’t blacklist honest nodes
  - Find more malicious nodes than basic triangle inequality
- Cons :
  - More expensive per node than triangle inequality
  - Still doesn’t catch all malicious nodes

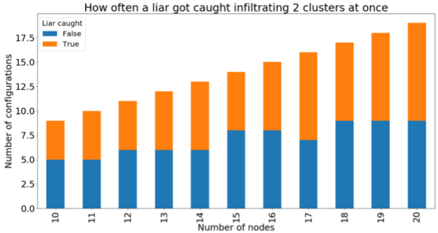
Fooling the Enhancement

- A malicious node can escape detection by
  - making its lies more realistic
  - lying to fewer nodes
- AKA : behaving better

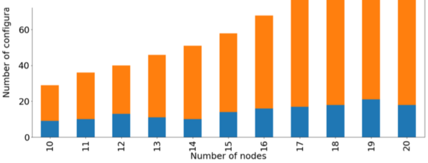
How does this work with (non-overlapping) regions ?



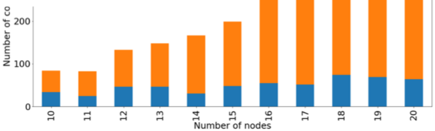
Infiltrating regions (1 liar)



← 2 regions



← 3 regions



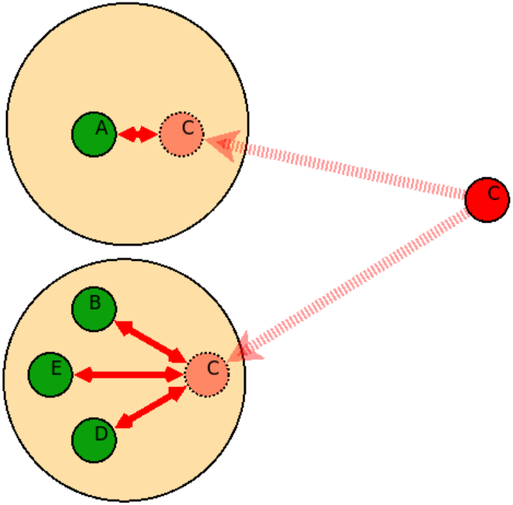
In given configuration, liar was...

🟧 Caught

🟦 Not Caught

4 regions →

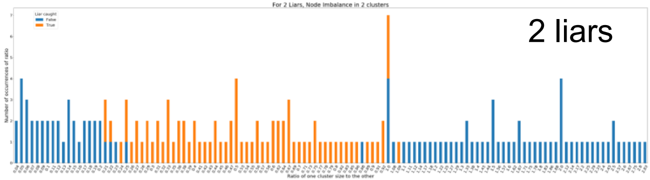
When does infiltration go undetected ?



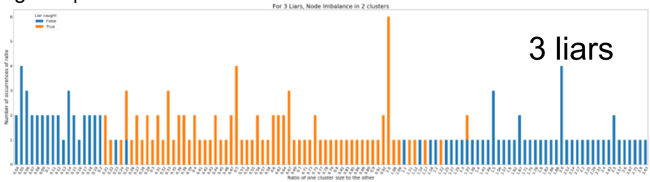
TI Violations
A-**C**-B
A-**C**-D
A-**C**-E

When can we infiltrate regions ?

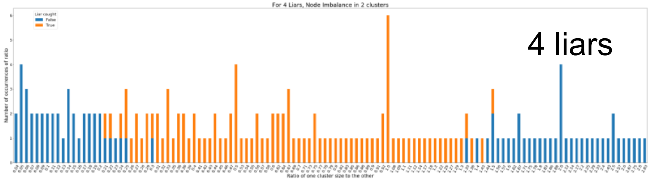
$$\frac{|\text{Region 1}|}{|\text{Region 2}|}$$



(a) 2 malicious nodes



(b) 3 malicious nodes



(c) 4 malicious nodes

In given configuration, liar was...

🟧 Caught

🟦 Not Caught

Conclusions

- Infiltrating too many regions →detected
- Too many infiltrating region →detected
- Imbalanced region sizes make small regions vulnerable – but small regions implies few nodes affected

Summary

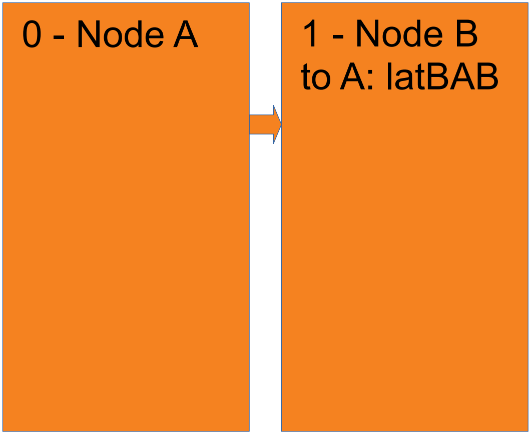
- A messaging protocol for the secure exchange of latencies between nodes
- An algorithm capable of detecting malicious nodes attempting to infiltrate multiple regions

Messaging Protocol in Detail

Stored Latencies

0 - Node A

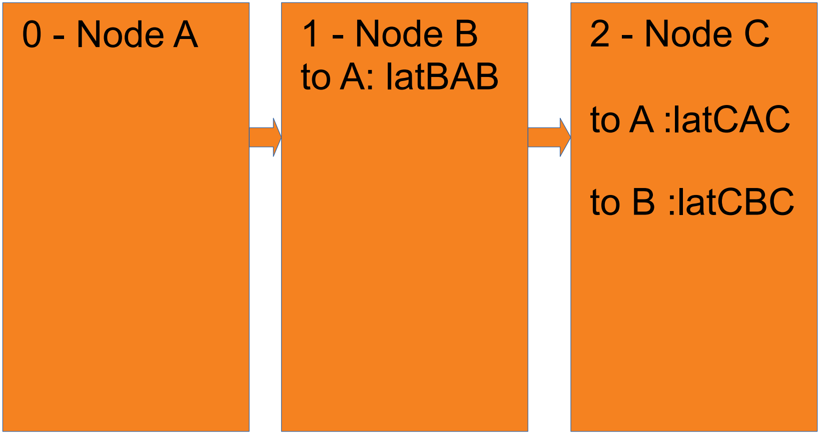
Stored Latencies



0 - Node A

1 - Node B
to A: latBAB

Stored Latencies



0 - Node A

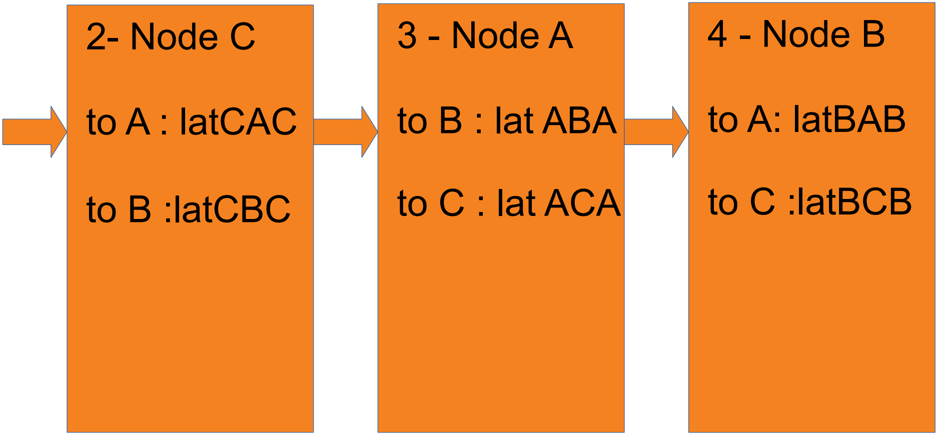
1 - Node B
to A: latBAB

2 - Node C

to A :latCAC

to B :latCBC

Stored Latencies



2- Node C

to A : latCAC

to B :latCBC

3 - Node A

to B : lat ABA

to C : lat ACA

4 - Node B

to A: latBAB

to C :latBCB

Blacklist Enhancment : Example

- We want to find out if node n is honest
- We compute the strikes for all the nodes using only triangles which n is part of

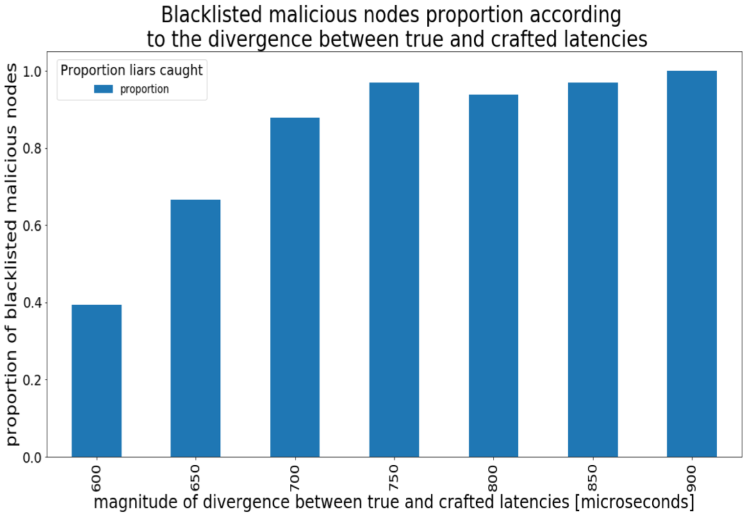
Case n honest

- If the second node s in the triangle is honest, it will only receive strikes if the third node is a liar
- This happens at most N/3 times, once for each liar
- s receives at most N/3 strikes
- There exist at least (2N/3) – 1 honest nodes

Case n honest

- Ergo : n honest => we can find at least h = (2N/3) – 1 nodes with ≤ N/3 strikes
- Contrapositive :
- If we cannot find h nodes with ≤ N/3 strikes => n is not honest

Lying inconsistently – Lie Size


Blacklisted malicious nodes proportion according
to the divergence between true and crafted latencies
Proportion liars caught
proportion
proportion of blacklisted malicious nodes
magnitude of divergence between true and crafted latencies [microseconds]