UNIVERSITY OF PISA
AND
SANT’ ANNA
SCHOOL OF ADVANCED STUDIES

Discovering and Securely Storing a Network Topology

Master Degree in Computer Science and Networking

Candidate
Francesco Balzano

Supervisor
Fabrizio Baiardi

Academic Year
2017/18

Contents

1.Introduction

- Knowing the network topology is a fundamental task. Anyway, this is seldom possible.
- So we implemented a network discovery tool.
- Since the network topology is a critical piece of information, we also implemented a blockchain to securely store it.

2. Related Works

2.1 Network Topology Inference

- Techniques that collect information to reconstruct a topology as close as possible to the real one.

- We target **router level**.

- Focus on **traceroute-based** methods.

- Traceroute is affected by flaws such as **aliases** and **non-responding routers**.

2. Related Works

2.1 Network Topology Inference

- Our tool adopts **iTop**, a technique to reconstruct a topology even if the network includes non-responding routers.

- The algorithm consists of three phases:
    1. Build a virtual topology.
    2. Compute merge options.
    3. Merge links.

2. Related Works

2.2 Blockchain

- A data structure that stores a secured, agreed-upon and shared ledger without requiring a trusted, centralized authority.
- Bitcoin blockchain is a tamper-proof data structure thanks to the use of cryptography, hash pointers and Proof-of-Work (PoW).
- A hash pointer is a pointer with a cryptographic hash of the information it refers to.
- The PoW is the distributed consensus algorithm used in Bitcoin.

2. Related Works

2.2 Blockchain

- Since PoW is CPU greedy, we discard it in favour of the Ripple Protocol Consensus Algorithm *(RPCA)*.
- RPCA proceeds in rounds. Each round consists of three phases:
  - **Open:** Each node sends and collects transactions over the network.
  - **Establish:** Each node exchanges proposals with its peers, trying to reach an agreement.
  - **Accept:** Each node applies the agreed transactions to the prior ledger to generate the new one.

3. Network Topology Inference Implementation

The inference tool is written in Python and runs on two kinds of nodes:

- The **monitors**: at the edge of network, they build a first snapshot of the network topology.
- The **sensors**: inside the network, they track network changes and update the topology.

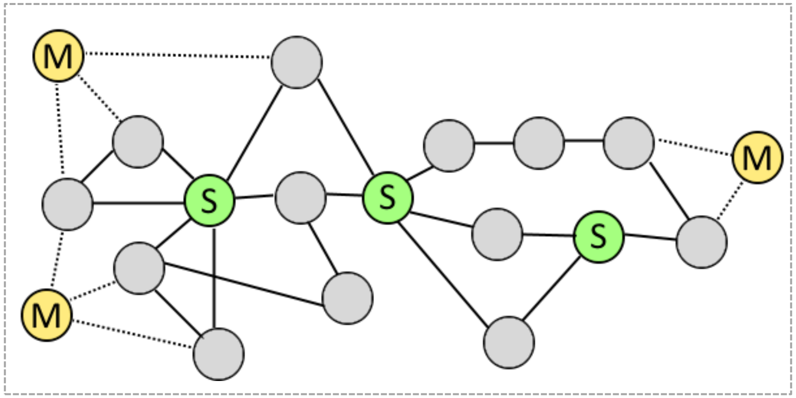


Fig.1. A sample topology, where monitors are labeled as “M” and sensors as “S”.

4. Blockchain Implementation

- The blockchain is based on Ripple and written in Python.
- We distinguish *clients* and *servers* of the blockchain.
- Cryptographic tools are employed.
- It is a permissioned blockchain.
- Sensors and monitors are blockchain clients.
- Servers provide a graphical representation of the stored topology.

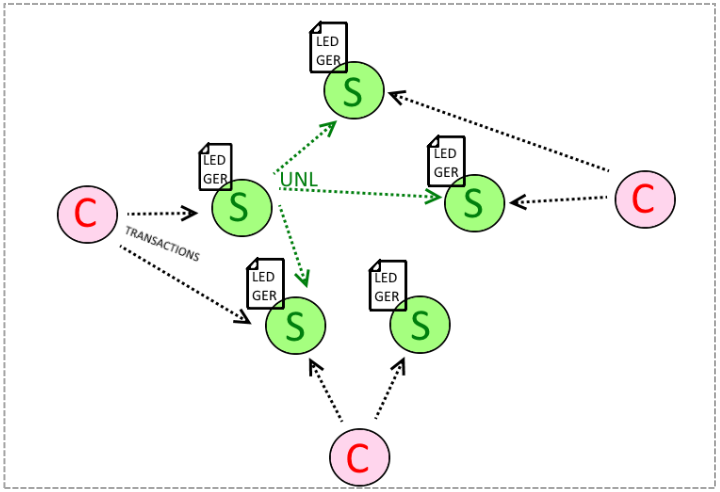


Fig.2. Blockchain architecture.
Clients are labeled as "C", servers as "S".

5. Experimental Results

5.1 Network Topology Inference

- We used *Mininet,* a network emulator that builds virtual test networks whose hosts support standard Linux software.
- We deployed sensors on virtual test networks and generated traffic among the network nodes.
- The performance of the inference tool is assessed in tems of the following metrics:

True Positive (TP)	A node belongs to the real topology and to the inferred one
True Negative (TN)	A node neither belongs to the real topology nor to the inferred one
False Positive (FP)	A node belongs to the inferred topology but not to the real one
False Negative (FN)	A node belongs to the real topology but not to the inferred one
Precision (P)	It measures the precision of the inferred topology. P = |TP| / |TP+FP|
Recall (R)	It is a measure of completeness. R = |TP| / |TP+FN|
F1-Measure (F1)	The harmonic average of P and R. F1 = 2 * ((P*R)/(P+R))

5. Experimental Results

5.1 Network Topology Inference

Each simulation averages the metrics collected in 20 independent experiments.

Simulation 1. One router, two subnets, variable number of hosts (3, 10 and 50 per subnet), two sensors.

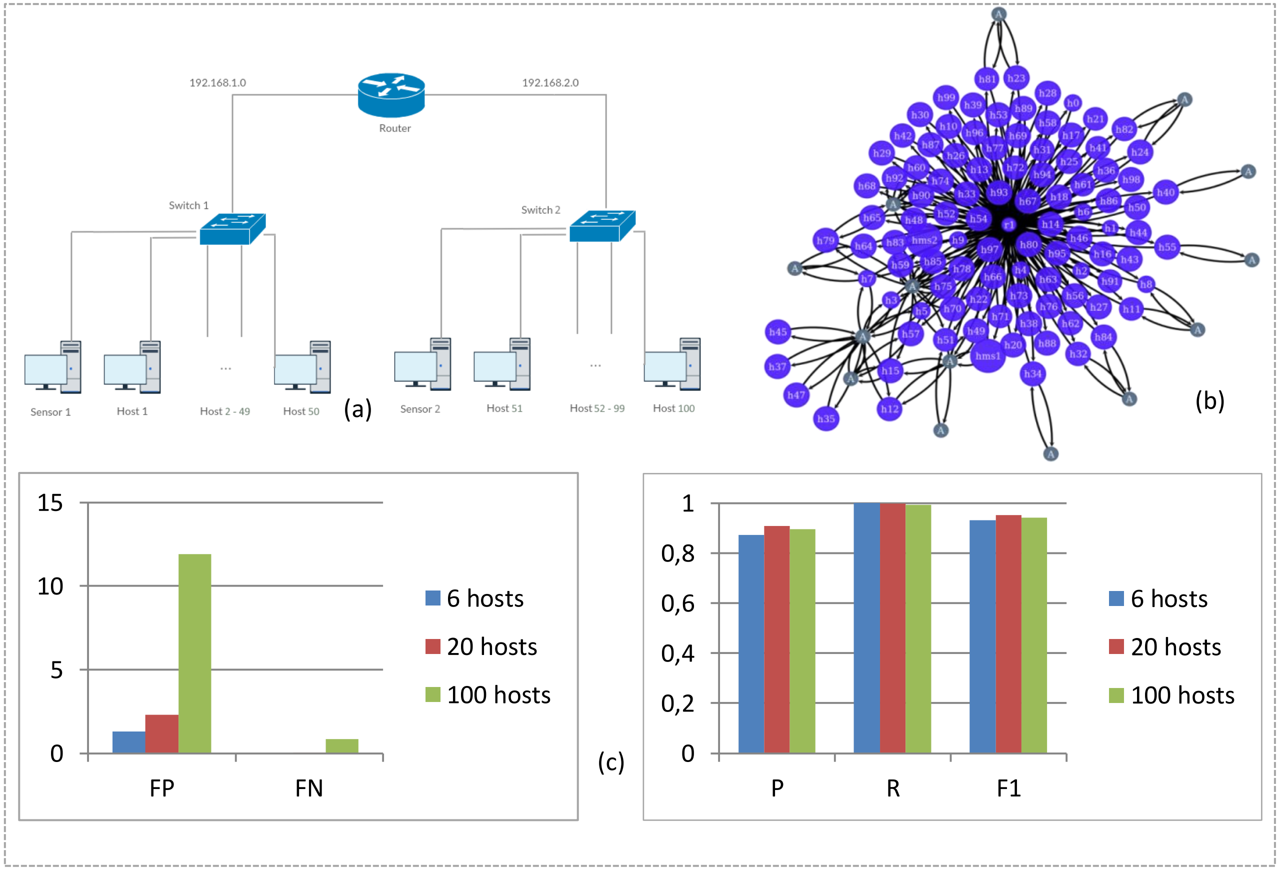
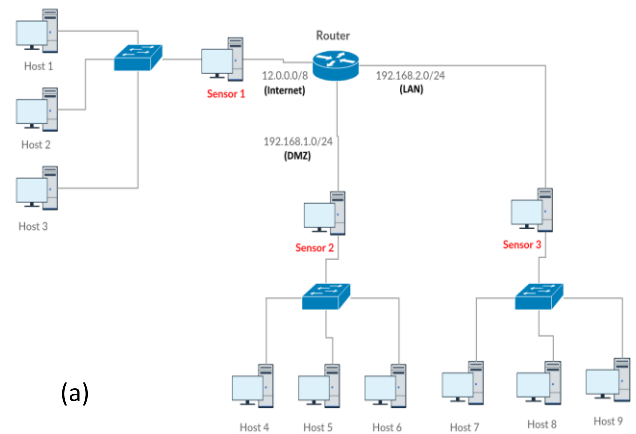


Fig.3: Original topology (a), inferred topology (b) and computed metrics (c).

5. Experimental Results

5.1 Network Topology Inference

Simulation 2. One router, three subnets, up to three sensors.



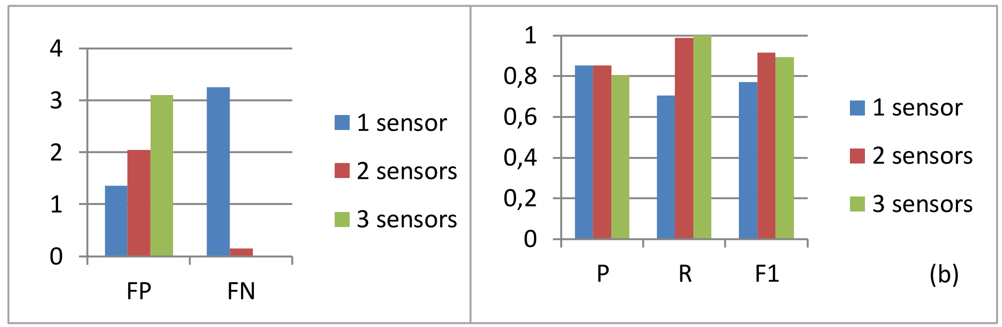
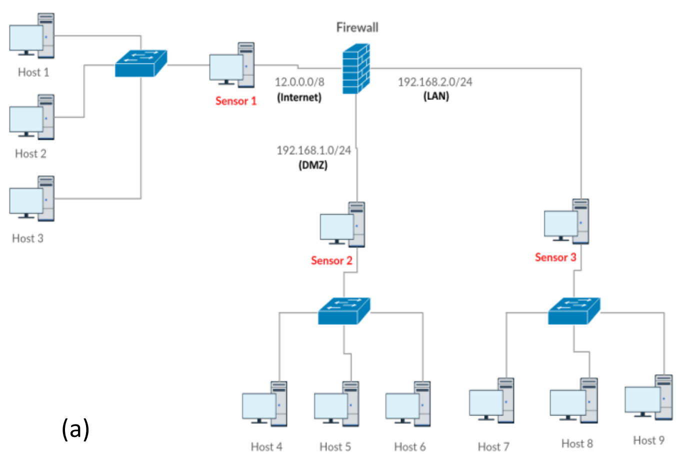


Fig. 4: Original topology (a) and computed metrics (b) for simulation 2.

Simulation 3. One firewall, three subnets, up to three sensors.



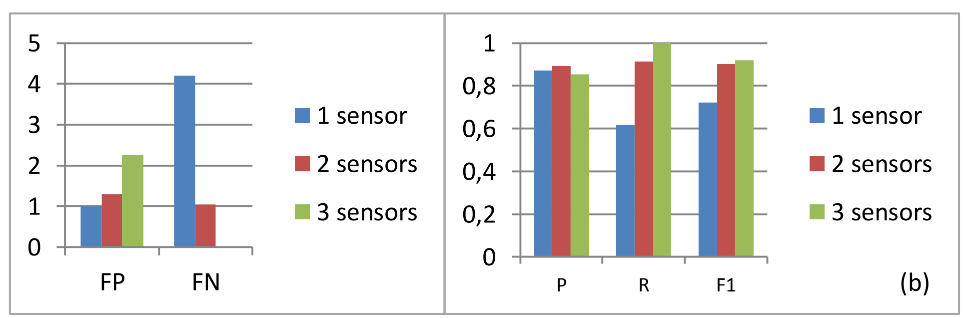
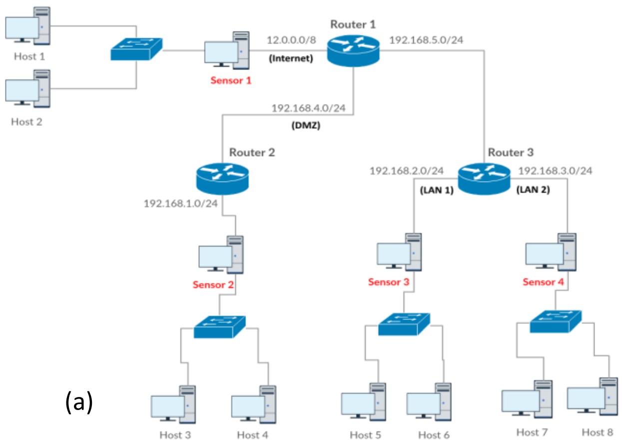


Fig. 5: Original topology (a) and computed metrics (b) for simulation 3.

5. Experimental Results

5.1 Network Topology Inference

Simulation 4. Three routers, four subnets, up to four sensors.



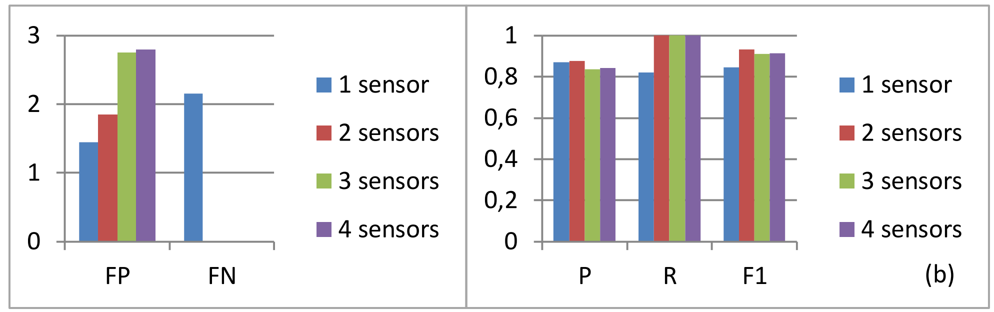
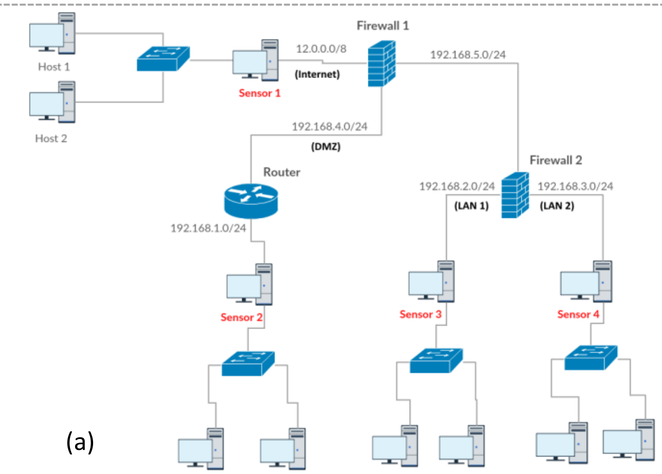


Fig. 6: Original topology (a) and computed metrics (b) for simulation 4.

Simulation 5. One router, two firewalls, four subnets, up to four sensors.



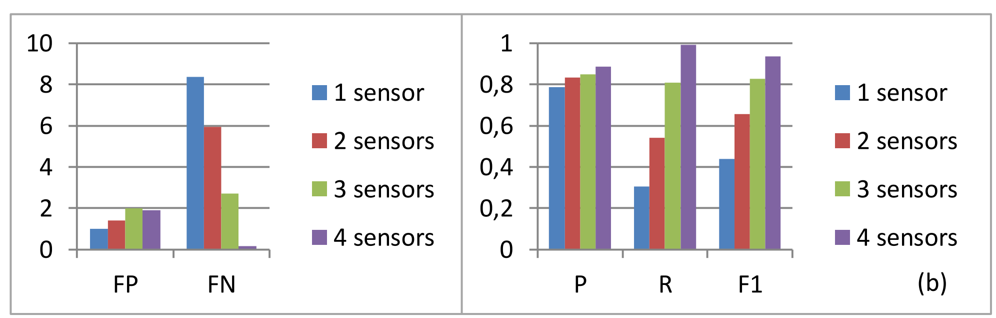
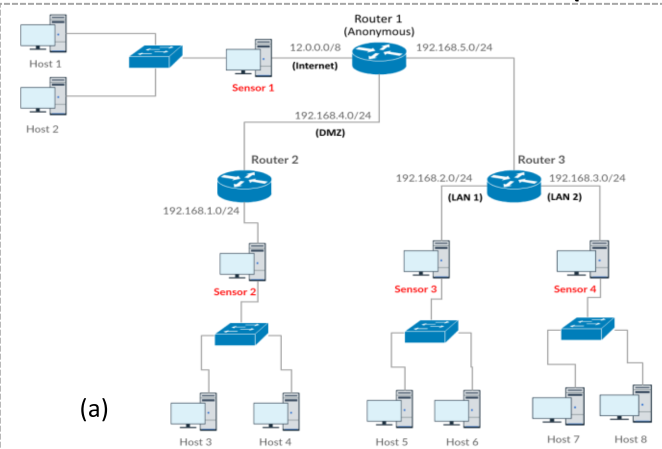


Fig. 7: Original topology (a) and computed metrics (b) for simulation 5.

5. Experimental Results

5.1 Network Topology Inference

Simulation 6. Three routers (one anonymous), four subnets, up to four sensors.



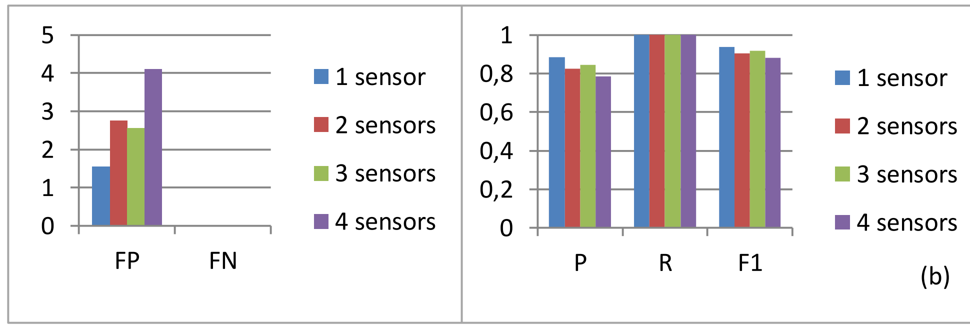
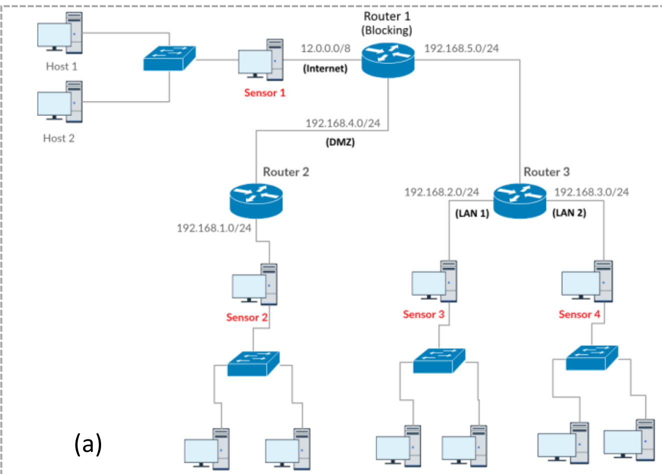


Fig. 8: Original topology (a) and computed metrics (b) for simulation 6.

Simulation 7. Three routers (one blocking), four subnets, up to four sensors.



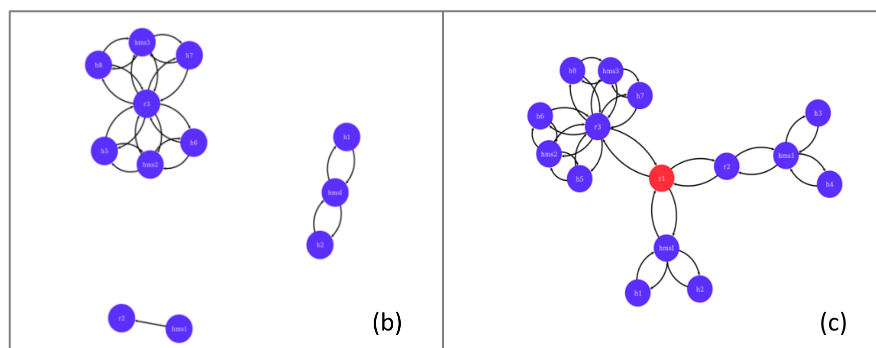


Fig. 9: Original topology (a) and inferred topology without (b) and with (c) the heuristic.

5. Experimental Results

5.1 Network Topology Inference

Simulation 8. Four firewalls, five tree-structured subnets, up to four sensors.

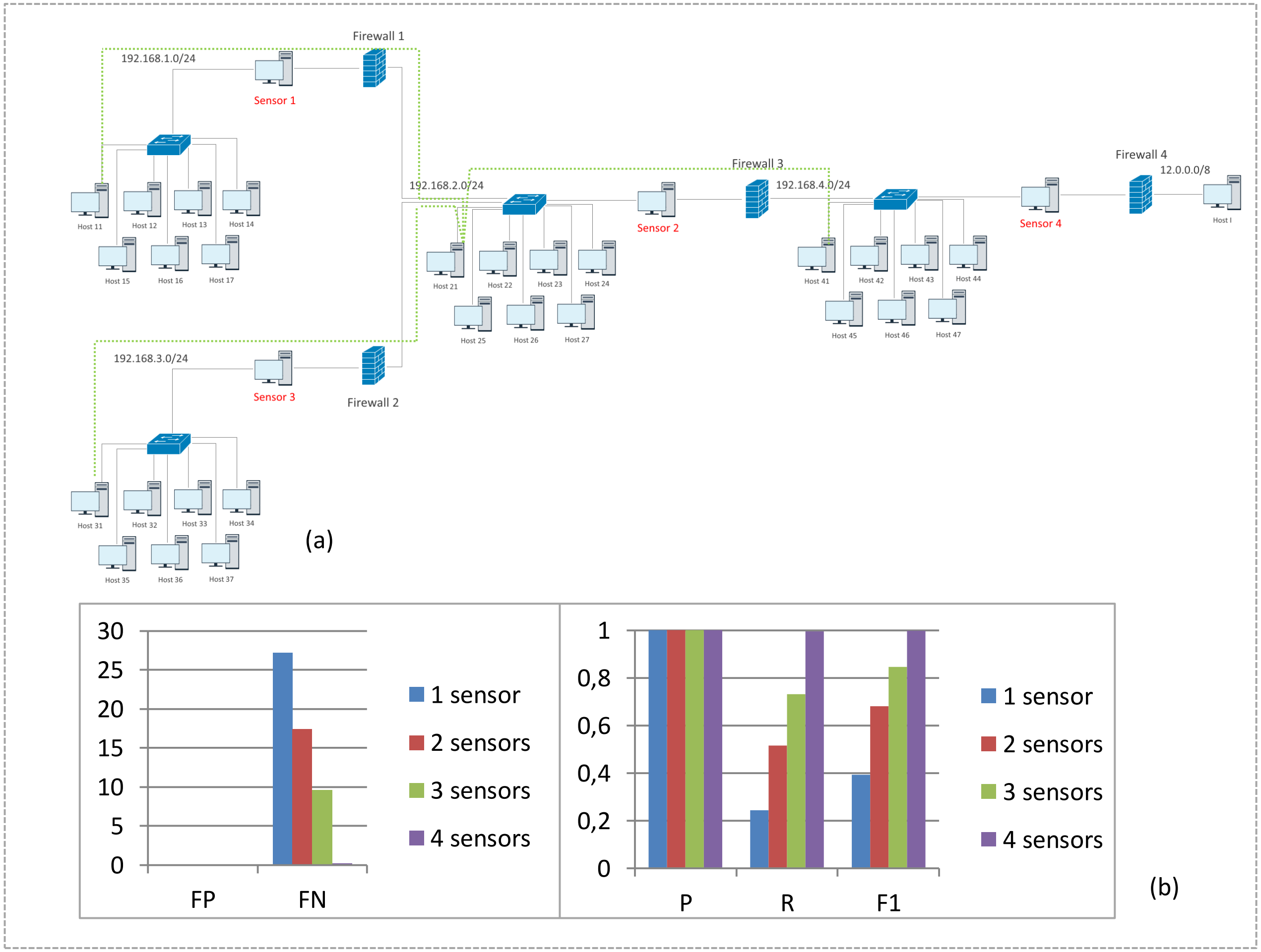


Fig. 10: Original topology (a) and computed metrics (b) for simulation 8.

5. Experimental Results

5.2 Blockchain

- We measured the time to reach consensus by varying the **quorum** values and the number of malicious nodes.
- Each simulation is the average of 20 independent experiments.
- The plotted execution times are referred to node 1.
- Consensus is reached in all the experiments.
- The architecture is shown in Fig. 11.

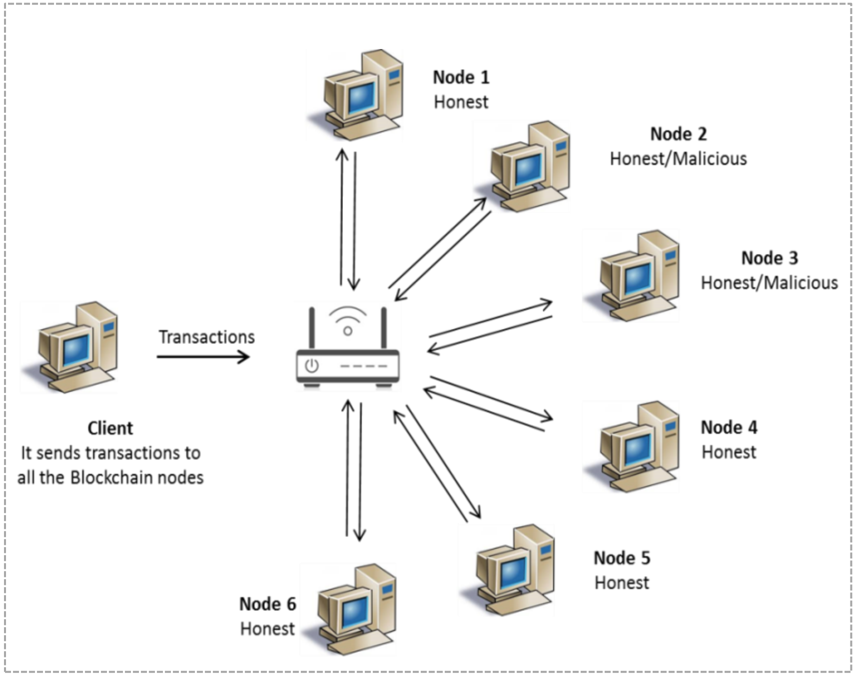


Fig. 11: Architecture for the blockchain simulations.

5. Experimental Results

5.2 Blockchain

Simulation 1. 500 honest transactions, quorum = 60%, up to two malicious nodes.

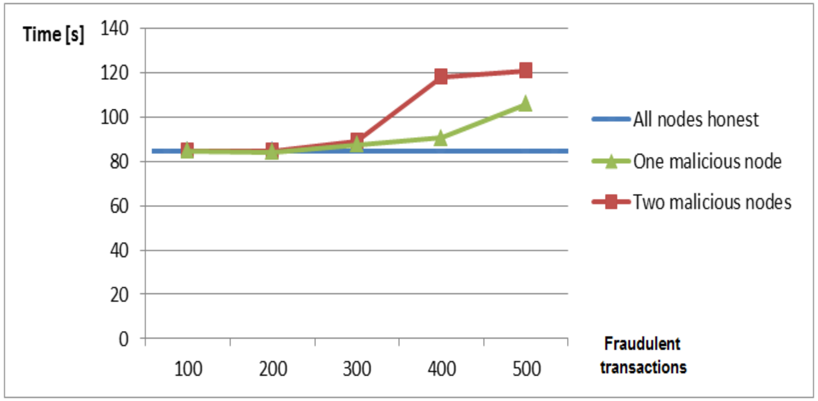


Fig. 12: Simulation 1. Execution times for one consensus round as a function of the number of fraudolent transactions inserted by malicious nodes.

Simulation 2. 500 honest transactions, distinct values for the quorum and the number of malicious nodes, emphasis on the single phases.

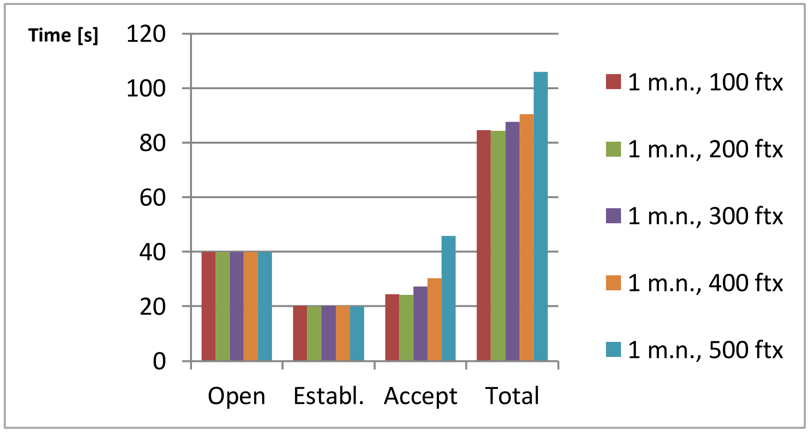


Fig. 13: Simulation 2a. Quorum = 60%, one malicious node inserting up to 500 fraudolent transactions.

5. Experimental Results

5.2 Blockchain

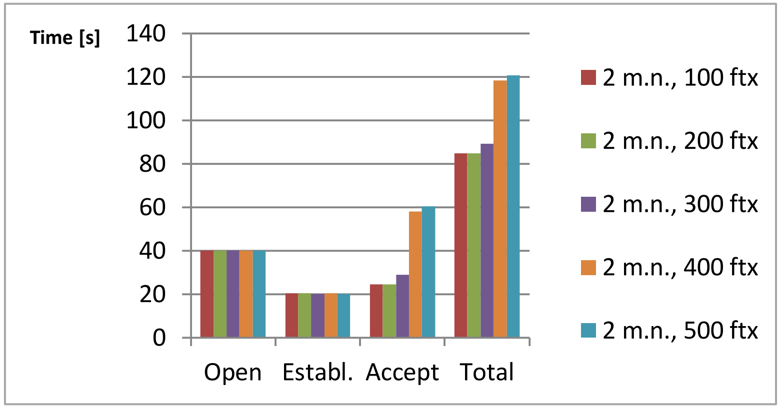


Fig. 14: Simulation 2b. Quorum = 60%, two malicious nodes inserting up to 500 fraudolent transactions each.

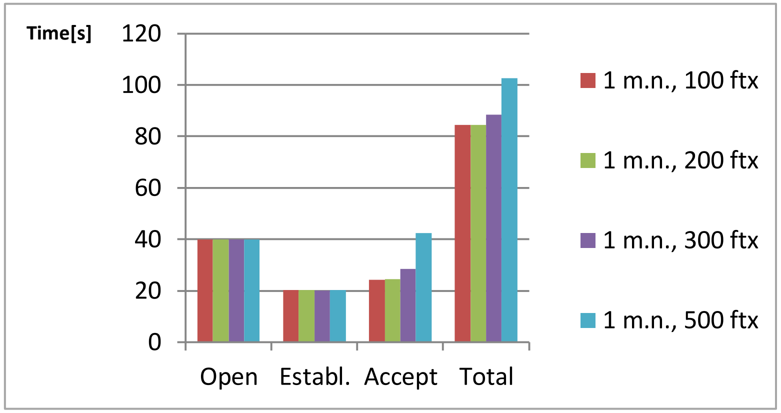


Fig. 15: Simulation 2c. Quorum = 80%, one malicious node inserting up to 500 fraudolent transactions.

Simulation 3. Variable number of honest transactions, all the nodes honest, three distinct quorum values.

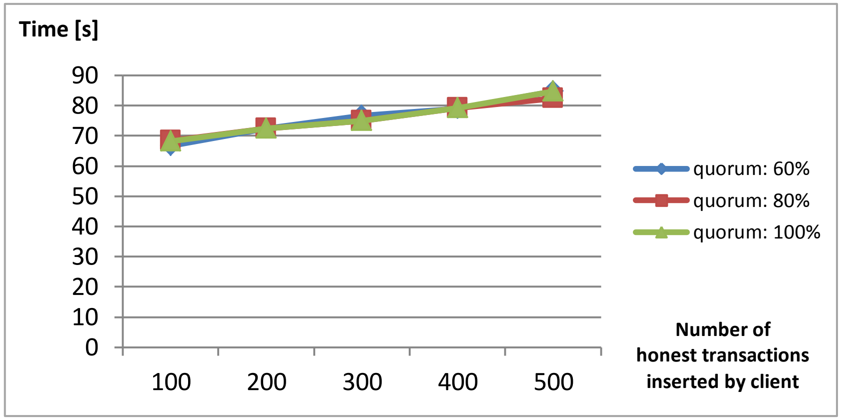


Fig. 16: Simulation 3. Execution times for one consensus round as a function of the number of inserted transactions.

5. Experimental Results

5.2 Blockchain

Simulation 4. 500 honest transactions, quorum = 80%, one malicious node that does not actively take part to the consensus process.

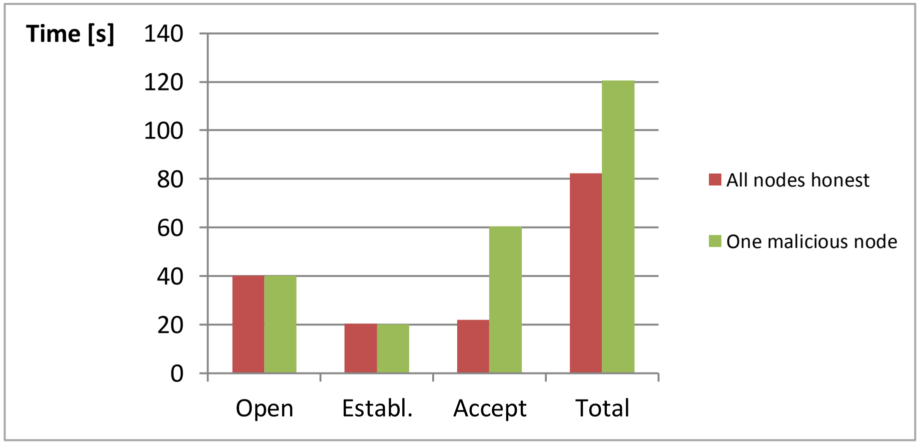


Fig. 17: Simulation 4. Execution times of the three phases of one consensus round when the malicious node does not actively take part to the consensus process.

6. Conclusions

- We implemented and assessed a tool that discovers a network topology and securely stores it using a blockchain.
- Topology Inference:
  - One sensor per subnet is a sufficient
- Blockchain:
  - Distinct quorum values do not affect the time to reach consensus. Instead, attackers injecting fraudulent transactions slow down the consensus process.
- Limitations:
  - Inference tool tested on an emulator.
  - Only target router-level topology.
  - Do not provide alias-resolution techniques.
- Future developments:
  - Feed the inferred topology to a vulnerability scanner.

Thank You

For Your Attention