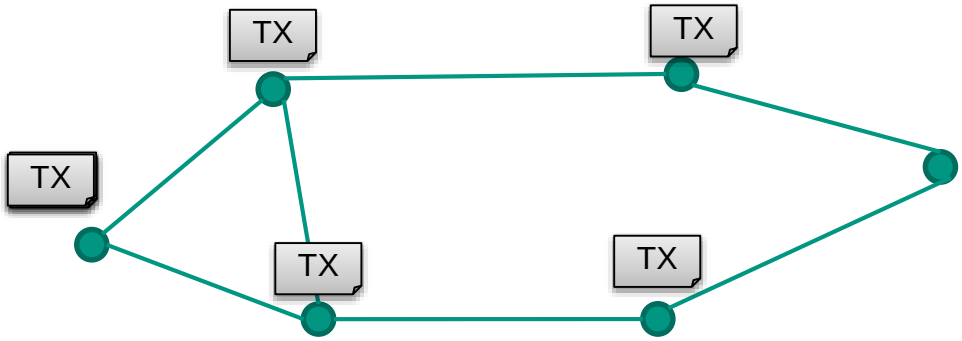
Timing Analysis for Inferring the Topology of the Bitcoin Peer-to-Peer Network

Till Neudecker, Philipp Andelfinger, Hannes Hartenstein

Steinbuch Centre for Computing (SCC) and Institute of Telematics,
DSN Research Group, Prof. Hartenstein

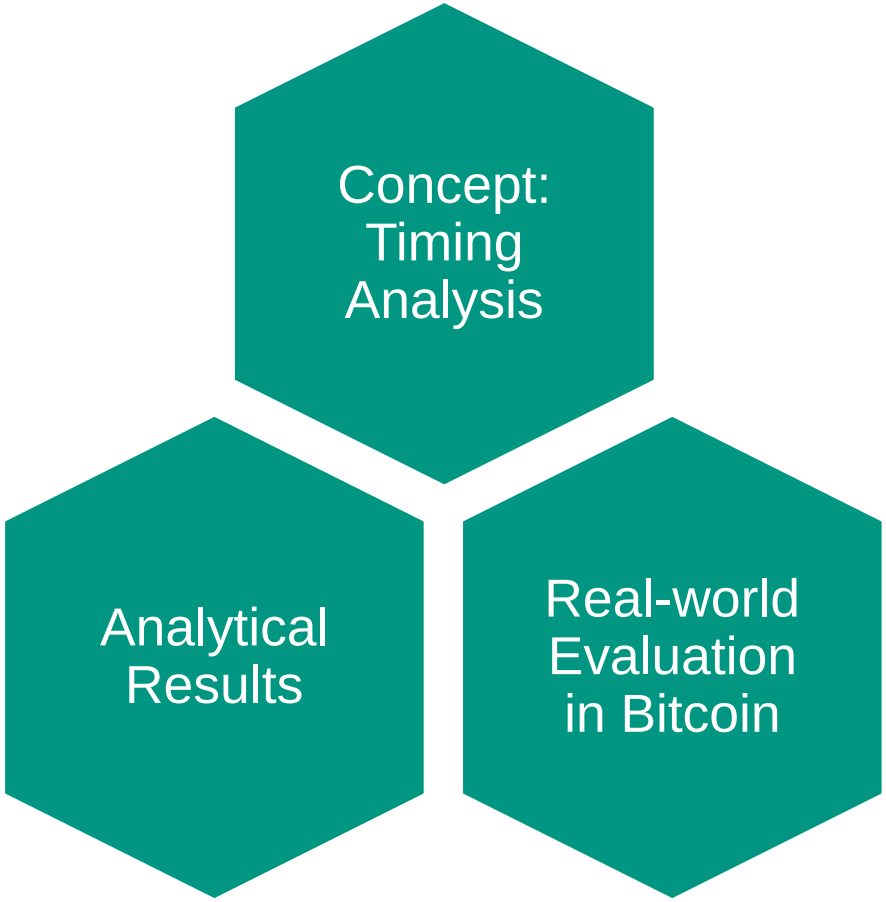
Motivation

- Bitcoin's infrastructure is a **flooding Peer-to-Peer (P2P) network**
- Transactions and Blocks are flooded through the network
  - Each new message is rebroadcasted to all neighbors
- Every node must receive them – fast!
  - Not fast → inefficient mining, double spending, …



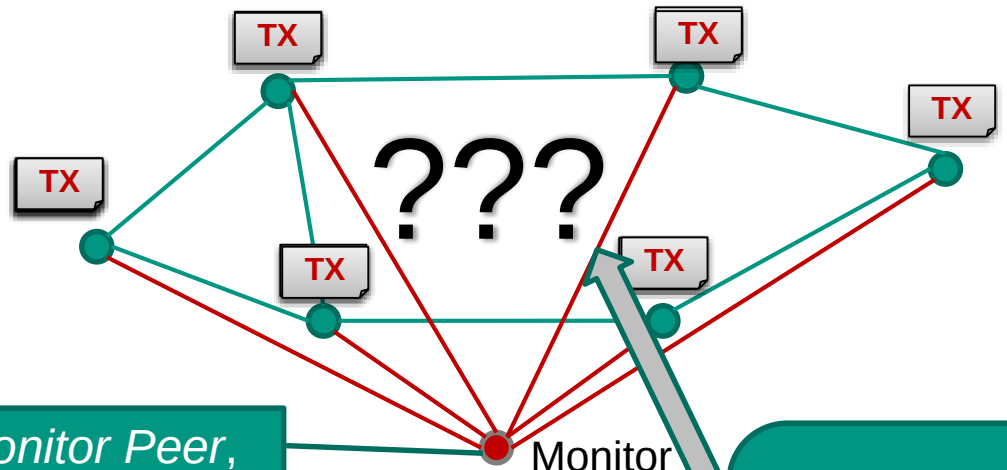
- We might learn something by observing the flooding process
  - Which peer brought the transaction into the network?
  - How are the connections between peers (i.e. the topology of the overlay network)?
  - Knowledge can be beneficial for attacker, network designer, researchers

Agenda


Concept:
Timing
Analysis
Analytical
Results
Real-world
Evaluation
in Bitcoin

How to observe flooding process

Assumption: Each flooded message is uniquely identifiably



TX_1

IP	Time
w.x.y.z	0
a.b.c.d	20
a.s.d.f	20
q.w.e.r	40
y.x.c.v	40
h.j.k.l	60

Monitor Peer, connected to all other peers

Problem statement

Use **this information** to derive **network topology**

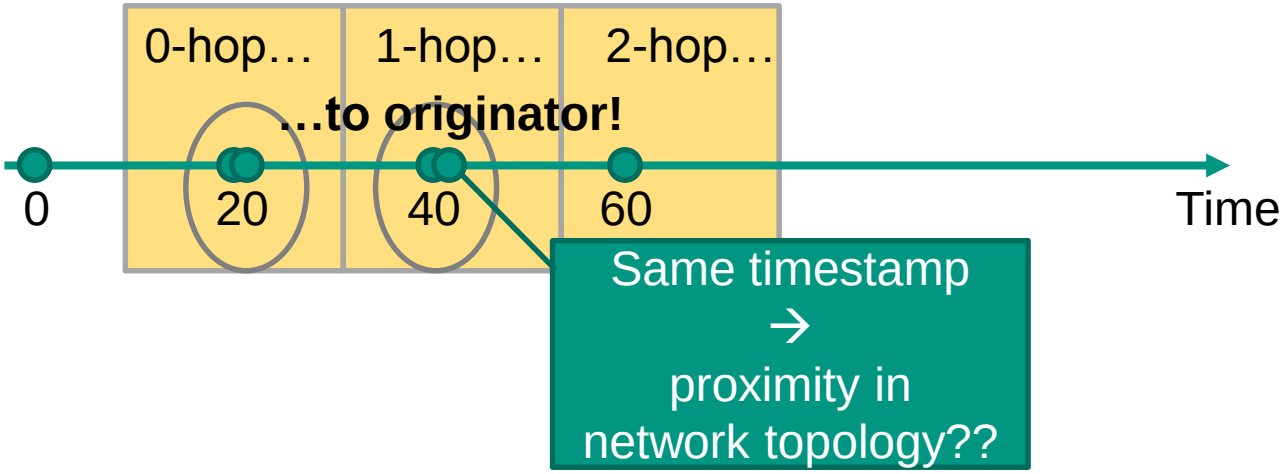
Assumption: We can connect to arbitrary peers of the network!

Related Work

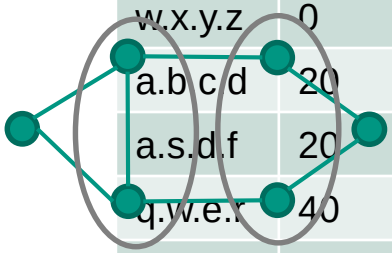
- Extract Bitcoin network topology
  - All use either side channel or rely on specific identifiable behavior
    - IP-Address Exchange: Coinscope [1], Marker IP Addresses [2]
    - Identifiable Behavior: e.g. [3]
- Timing Analysis in *Tor* (assuming a global-passive adversary (GPA))
  - Problem is similar but different
  - Differences
    - No linking between input and output messages because of onion encryption possible in Tor
    - Goal of attacker is different (circuits (end-to-end connections) in Tor vs. direct connections in flooding network)
    - …
  - In our paper: Mapping between both problems

[1] Miller, Andrew, et al. "Discovering Bitcoin's public topology and influential nodes." (2015).
[2] Biryukov, Alex, Dmitry Khovratovich, and Ivan Pustogarov. "Deanonymisation of clients in Bitcoin P2P network." *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014.
[3] Koshy, Philip, Diana Koshy, and Patrick McDaniel. "An analysis of anonymity in bitcoin using p2p network traffic." *International Conference on Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2014.

What can we learn from the data?



IP	Time
w.x.y.z	0
a.b.c.d	20
a.s.d.f	20
q.w.e.r	40
y.x.c.v	40
h.j.k.l	60

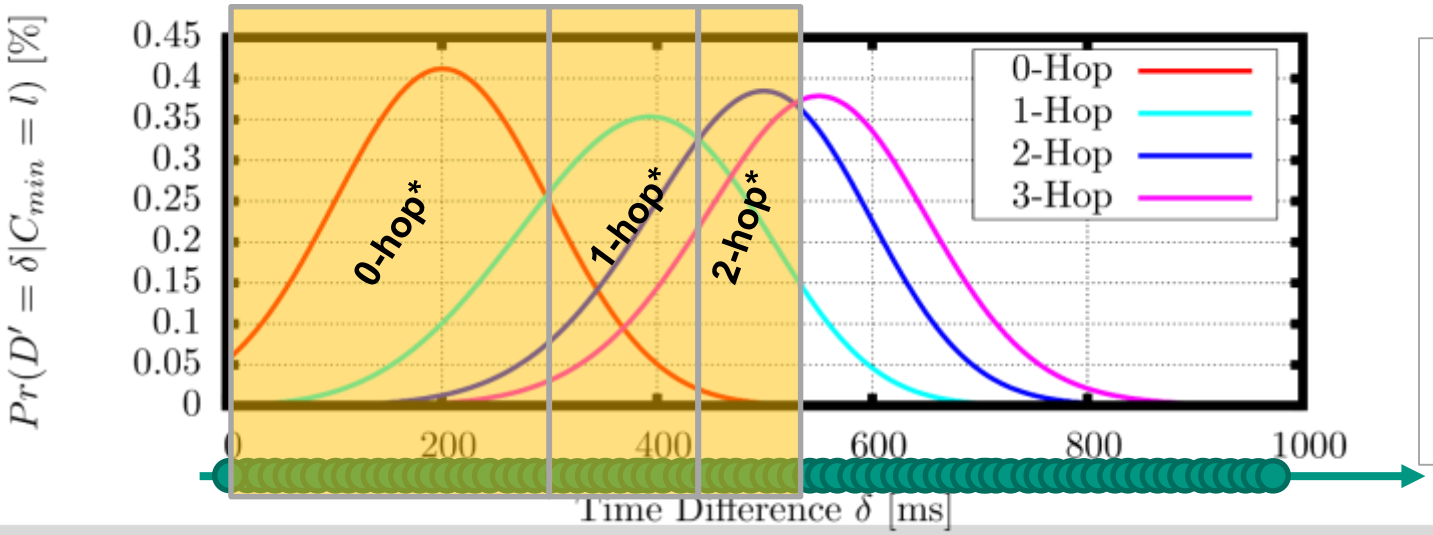


- Two peers rebroadcasting a message at the same time does not imply proximity in the network topology
- Time differences between the first peer that broadcasted the message in the network (aka the ***originator***) and other peers indicate **distance** in the network

Compare data against propagation delay model

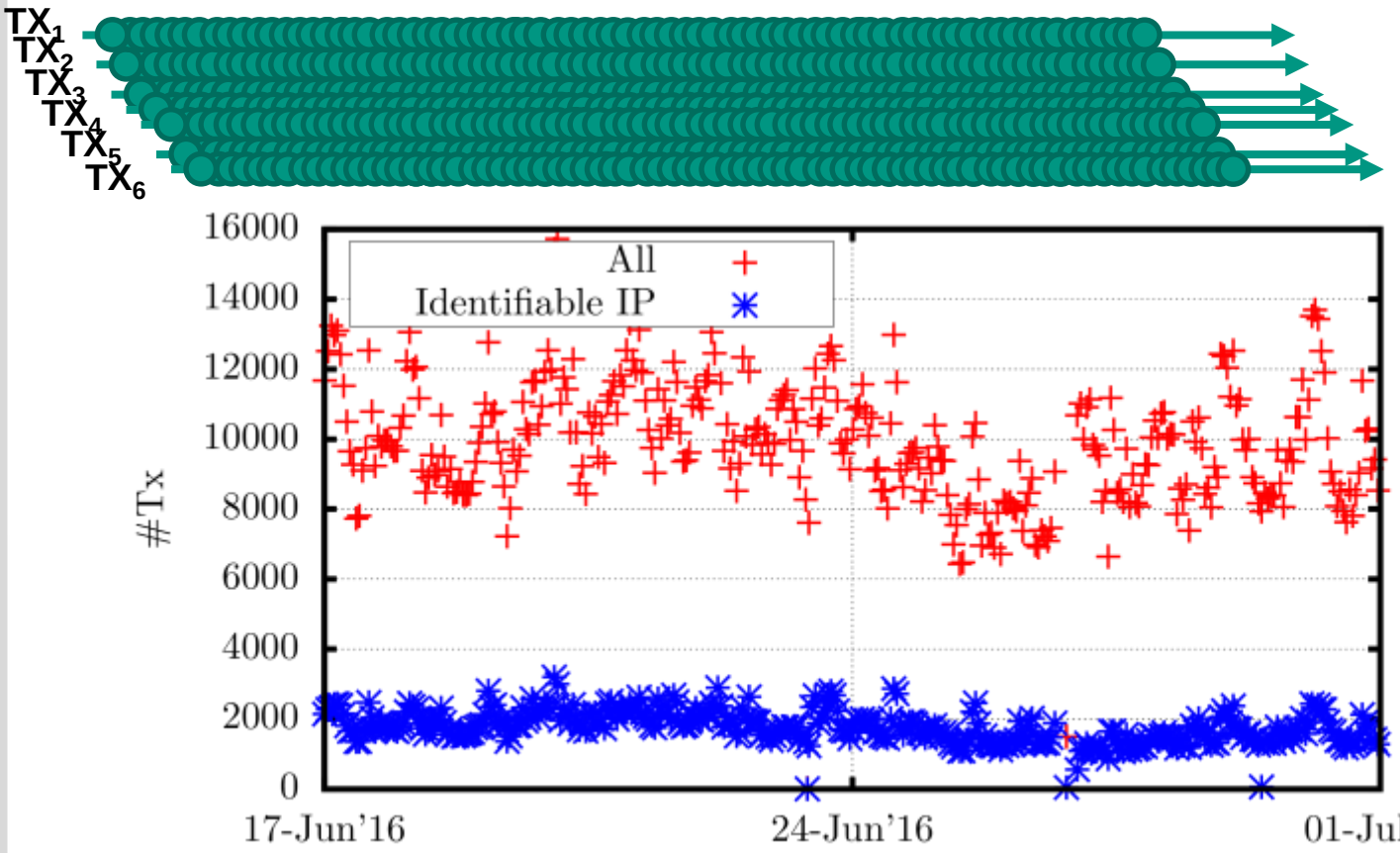
- Where do we assume 0-hop? Where 1-hop?
- We need a propagation delay model to compare to!

- In the paper: Analytical propagation delay model (approximate, ER-network, …)
  - Answers: If peers are *n* hops apart, how likely is a time difference of δ?



* A-priori probabilities ignored here for the sake of simplicity. Example Network. For Details see paper.

From one observation to many observations

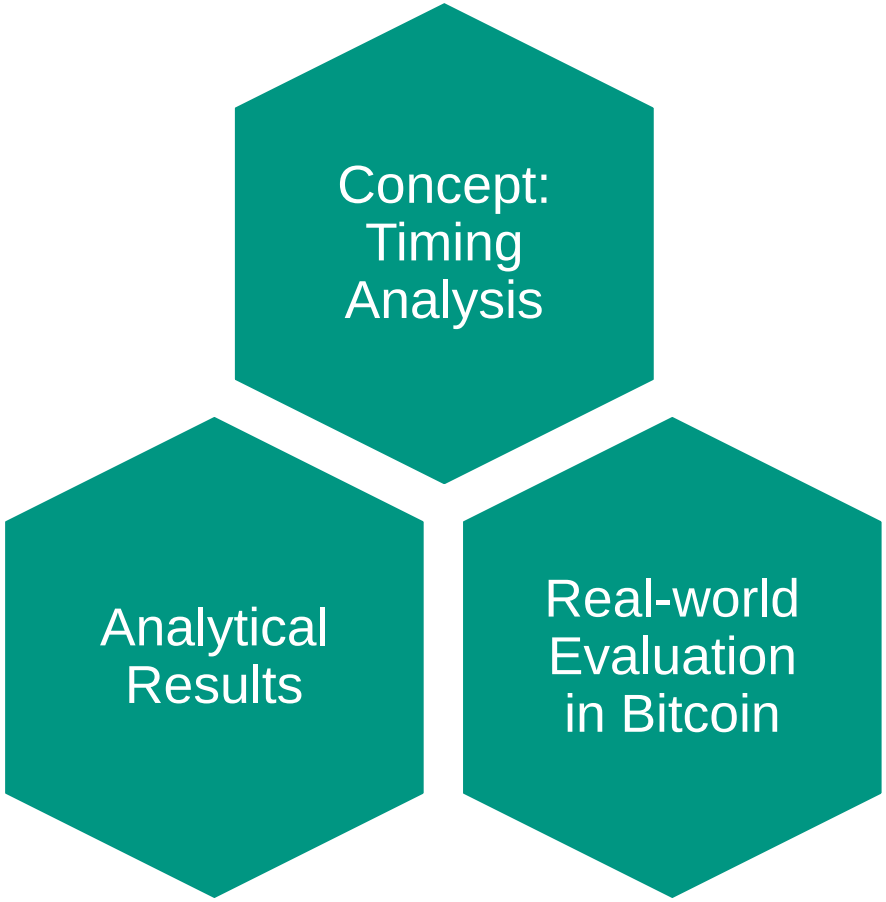


~ 2,000 relevant Tx per Hour

More plots: dsn.tm.kit.edu/bitcoin

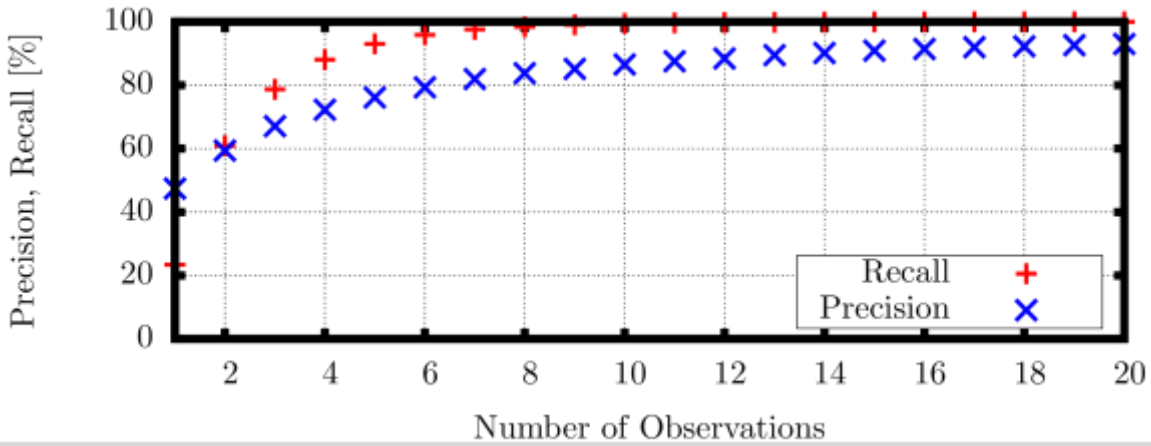
- Combine several observations (i.e. from several messages) with *maximum likelihood estimation* → see formalization in paper

Agenda


Concept:
Timing
Analysis
Analytical
Results
Real-world
Evaluation
in Bitcoin

Analytical evaluation

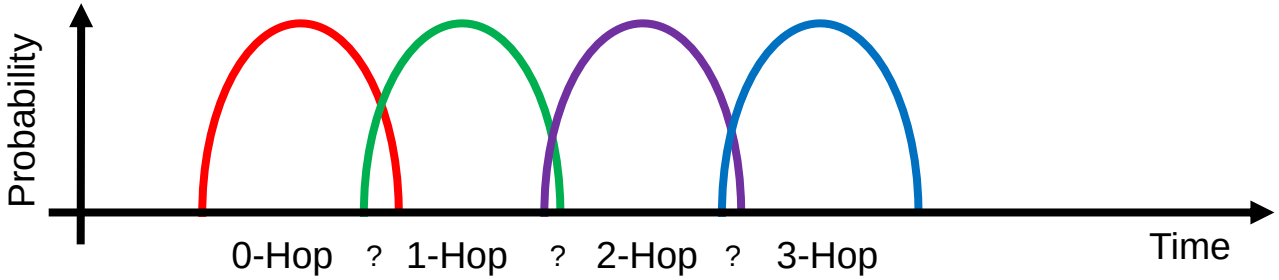
- How well concept may work in reality (“upper bound”):
- Method: Simulated network with known parameters
  - a.k.a. perfect conditions for detection!
- Detect only whether edge between two peers exists (*0-hop*) or not
  - False Positives → Low *Precision*
  - False Negatives → Low *Recall*



Works very well –
under perfect conditions.
Worse in reality →
see later (Bitcoin)

Countermeasure: Trickling

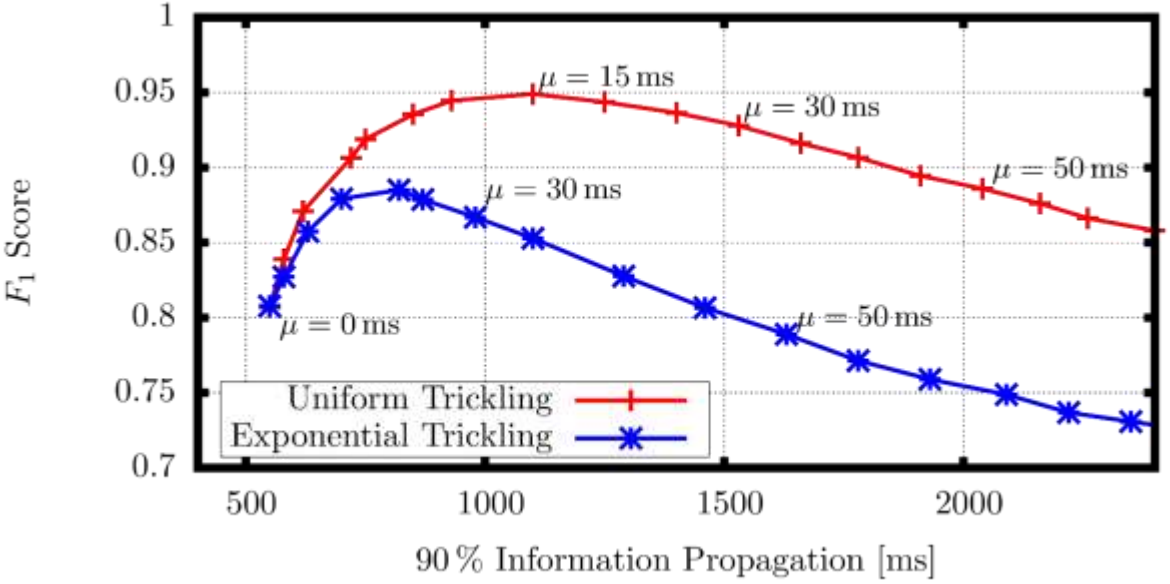
- In order to make timing analysis harder, peers may choose to randomly delay messages (*trickling*)
- Intuitively this broadens the shape of each curve and, therefore, increases the overlap



- However, it also slows down propagation speed
- Evaluate tradeoff between cost and gain of trickling:
  - Simulated trickling with different distributions and parametrizations

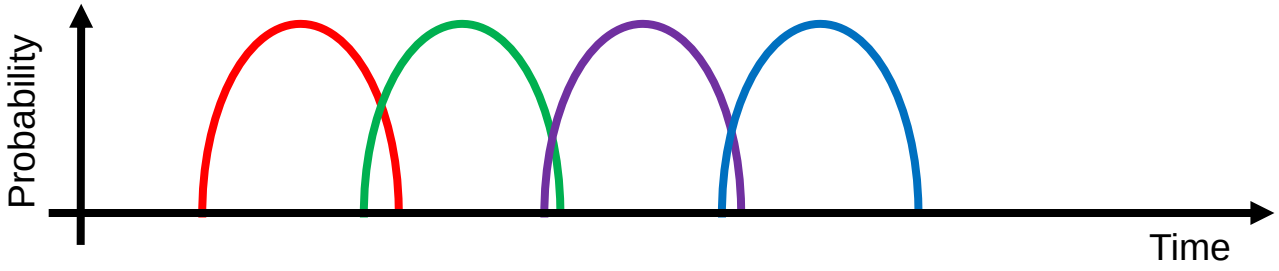
Countermeasure: Trickling (II)

= better detection



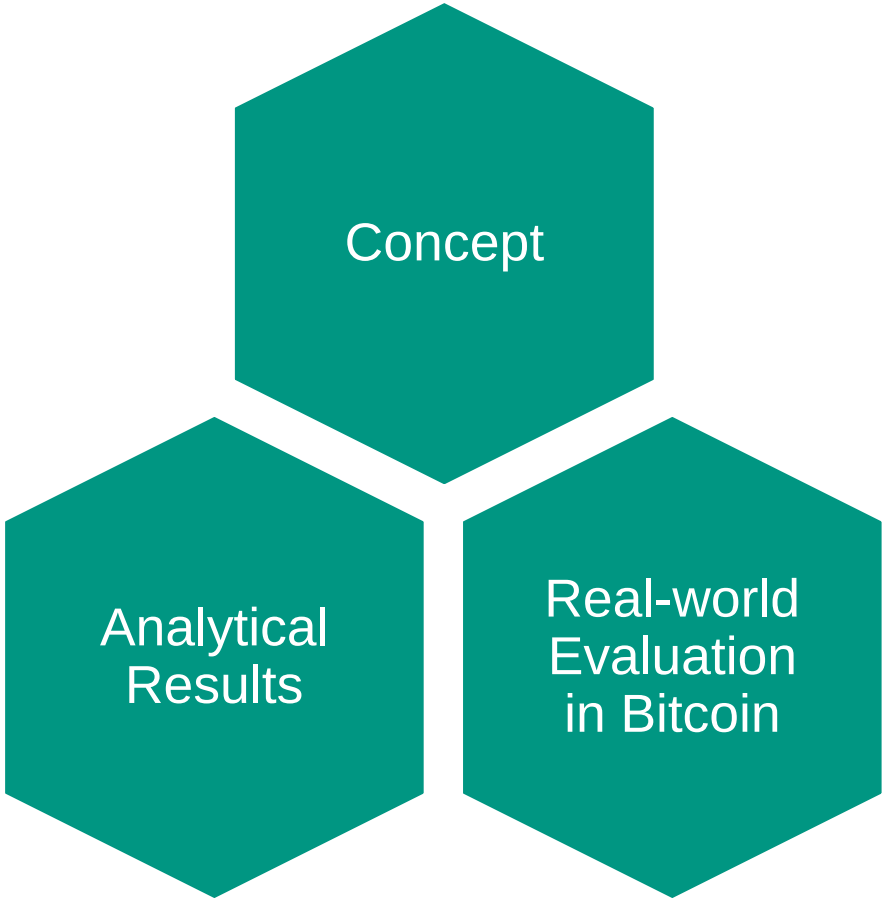
= slower propagation

- Trickling may (if badly parametrized) actually facilitate timing analysis!
- Why? Because it not only broadens shape but also shifts the mean:

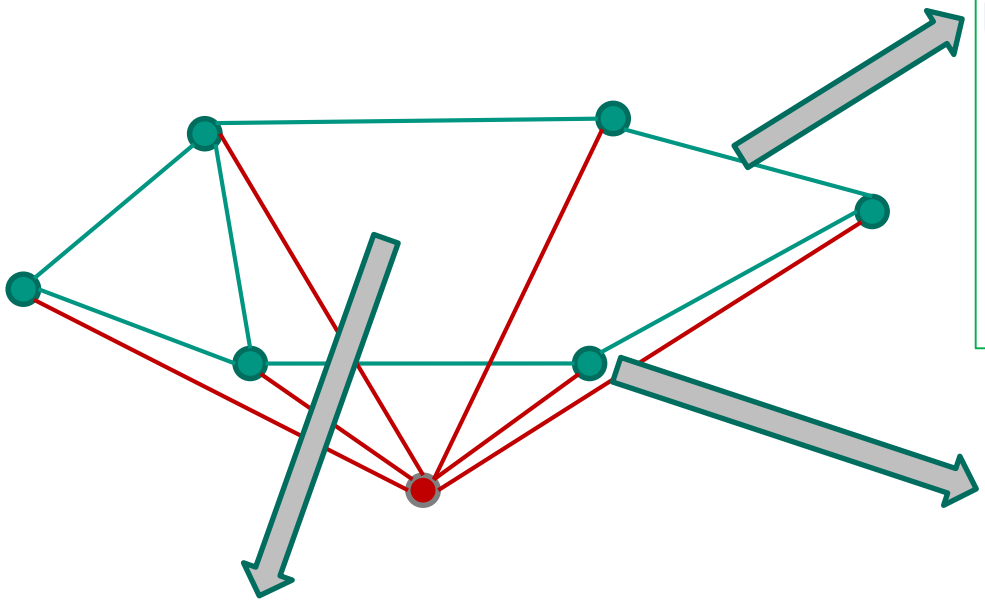


- But: Trickling can still prevent identifying the originator

Agenda


Concept
Analytical Results
Real-world Evaluation in Bitcoin

Bitcoin – Network Model



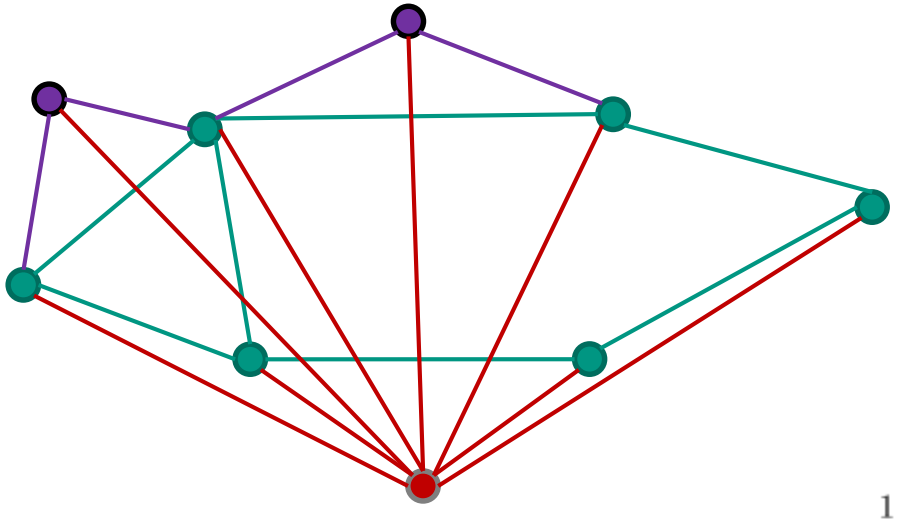
- Network Latency
  - Monitor → Remote Peer
    - Measured
  - Remote → Remote Peer
    - Estimated based on distance

- Client delay
  - Knowledge from client source code
  - Measured by comparing *application level ping* vs. *system level ping*

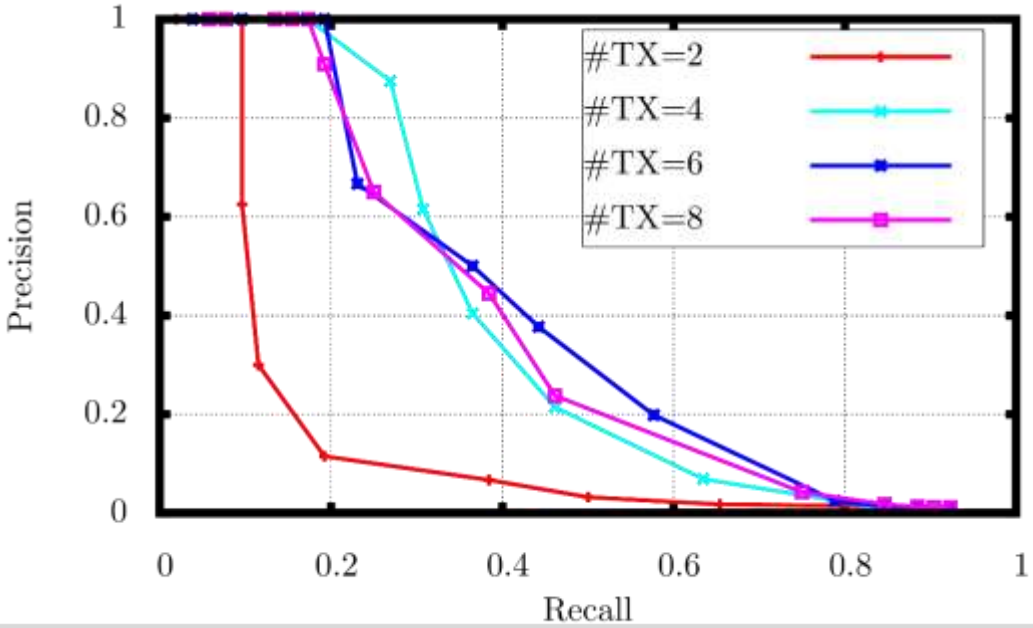
- Node degree distribution
  - Simulated different distributions
  - Compared resulting propagation delay to measured propagation delay
  - Chose the distribution with the least deviation

Details → Paper

Bitcoin – Ground Truth Validation



- Two remote network peers – controlled by us
  → known connections
- Each peer issued a number of transactions
- We monitored the information propagation and guessed the neighbors of our remote peers



- Best overall:
  40% Recall, 40% Precision
- No further improvement after ~6 transactions
- Many reasons for “bad” detection possible (i.e., any bad estimation)

- Sensitivity can be chosen depending on analysis’ goal

Conclusion & Outlook

- Timing analysis based on information inherent to flooding networks
- Trickling can help preventing timing analysis…
  - …but should be very carefully parameterized
- Real-World Evaluation in Bitcoin:
  - Timing analysis still works…
  - …but Precision & Recall are far from what is theoretically achievable

- Outlook:
  - Is further optimization for Timing Analysis in Bitcoin network possible?
  - What is an optimal choice of countermeasures (and its parametrization)?
  - How can gained information be “useful” except for *out of interest*
    - cf. Network Tomography

Backup

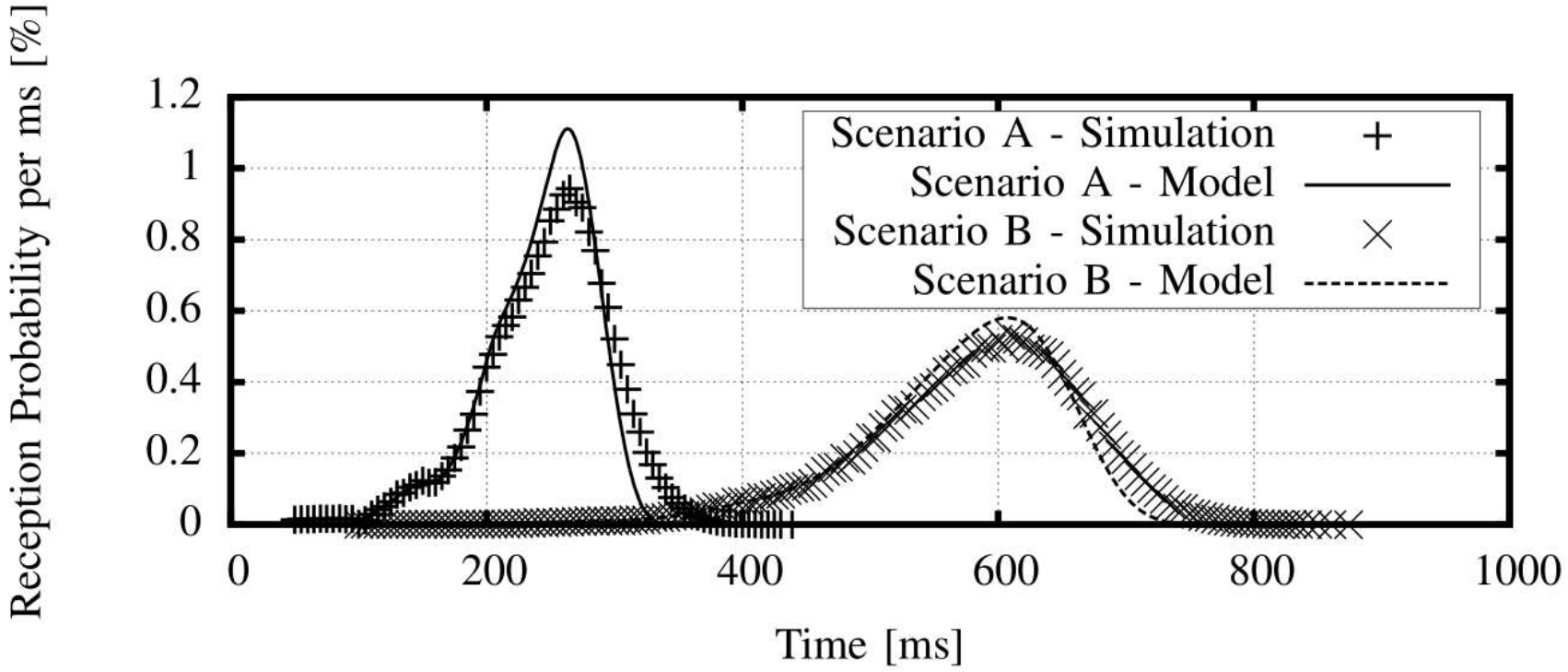


Fig. 8. Propagation Delay: Comparison of Model to Simulation

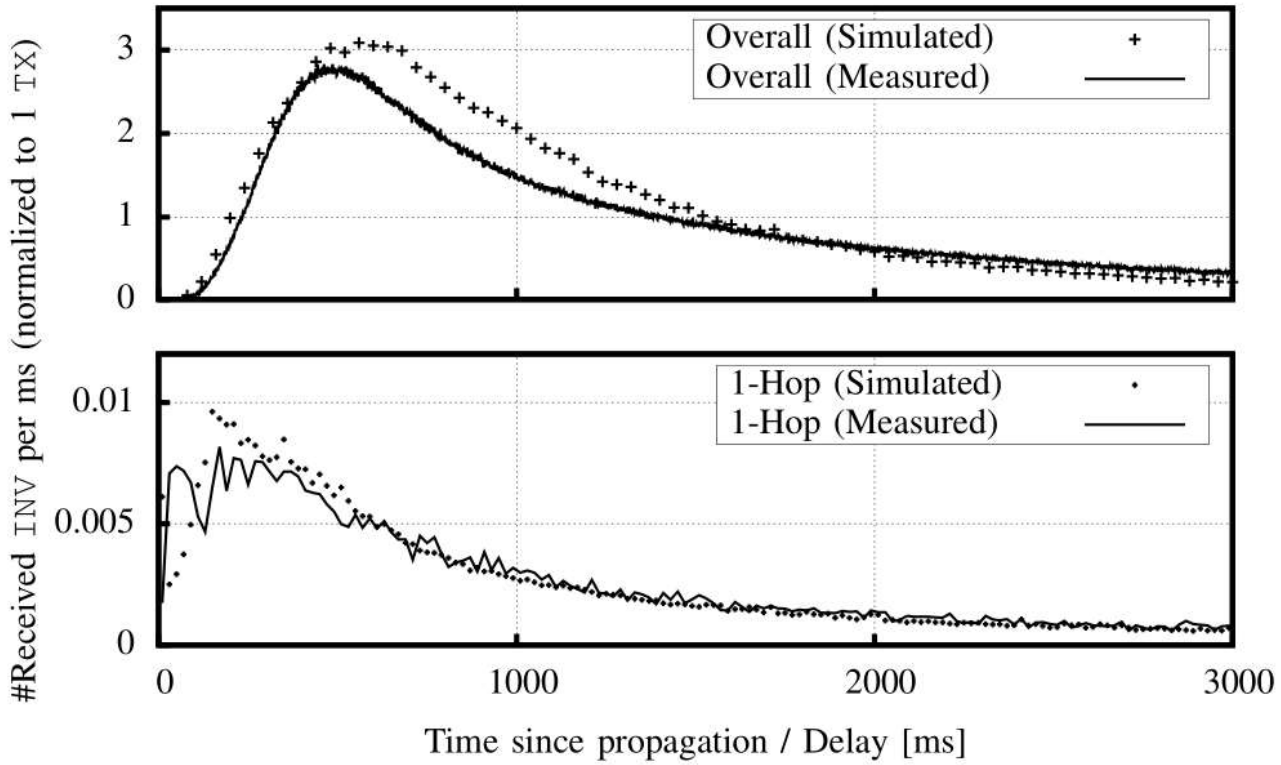


Fig. 6. Comparison between measured and simulated INV propagation delay as histogram data; limited to direct neighbors of originating peer (bottom) and for the complete network (top). Both networks parametrized with $\gamma = -2.3$. Normalization: shown values correspond to the creation of *one* transaction.

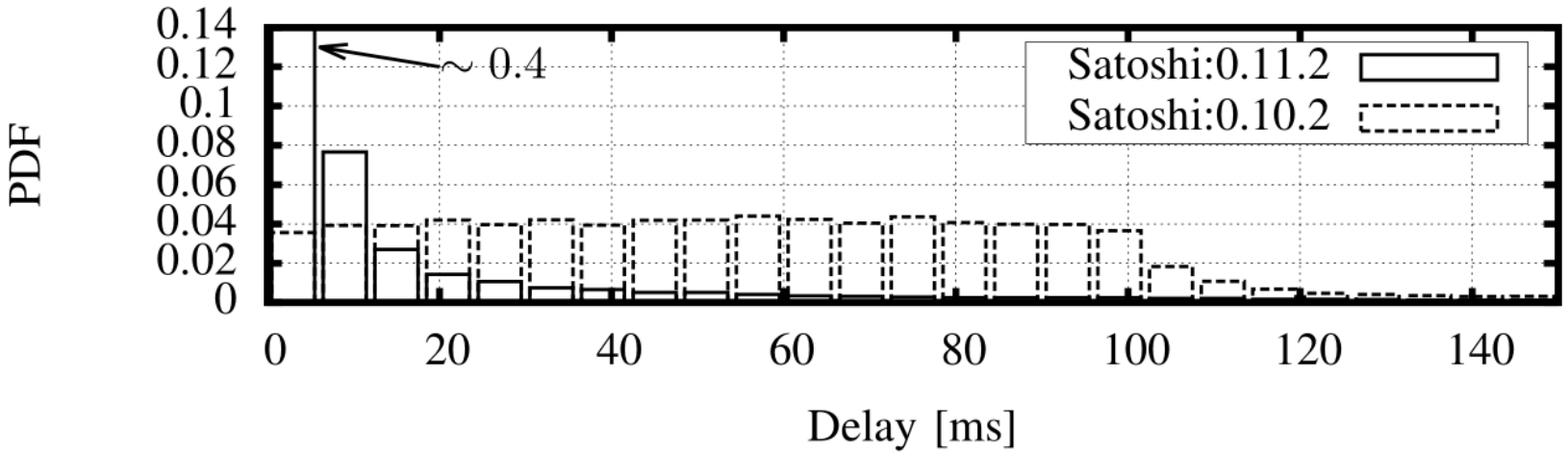


Fig. 5. Unintentional client delay for the bitcoin reference client version 0.10.2 and 0.11.2.

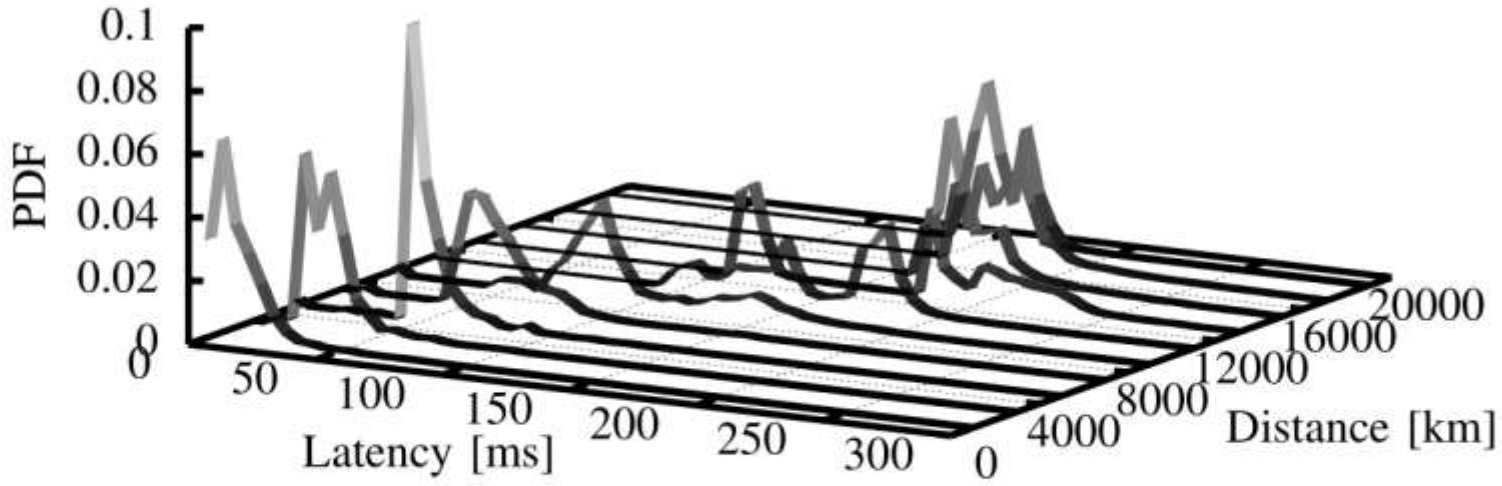


Fig. 4. Latency distribution broken down by geographical distance between measurement node and foreign peer. Binsize = 2000km.