Network Size Estimation

Christian Grothoff

Technische Universität München

June 8, 2012

fsnsg

Motivation

Purpose of Network Size Estimation

- Human curiosity
- Detection of unusual events
- Value of the botnet
- Tuning parameter

Structured Methods [4]

- Assume DHT with equal key distribution between peers
- (average) distance between keys is $\frac{1}{n}$

Non-local Structured Methods

- Each iteration, perform a "GET" request for a random key
- Observe distance d to closest peers to the key
- Calculate average $n \approx \frac{1}{d}$ over many rounds
- Cost: $O(n \cdot \log n)$ per round for the network

Local Structured Methods [1]

Basic Idea

- Observe DHT routing table
- Suppose there are p_k entries in bucket k
- Calculate size $n \approx p_k \cdot 2^k$
- Average over all non-full buckets
- Cost: no network overhead

Problems

- The formula above is intuitive but wrong.
- Polot proposed using a weighted average with buckets weighted based on the number of peers per bucket. That is also wrong.

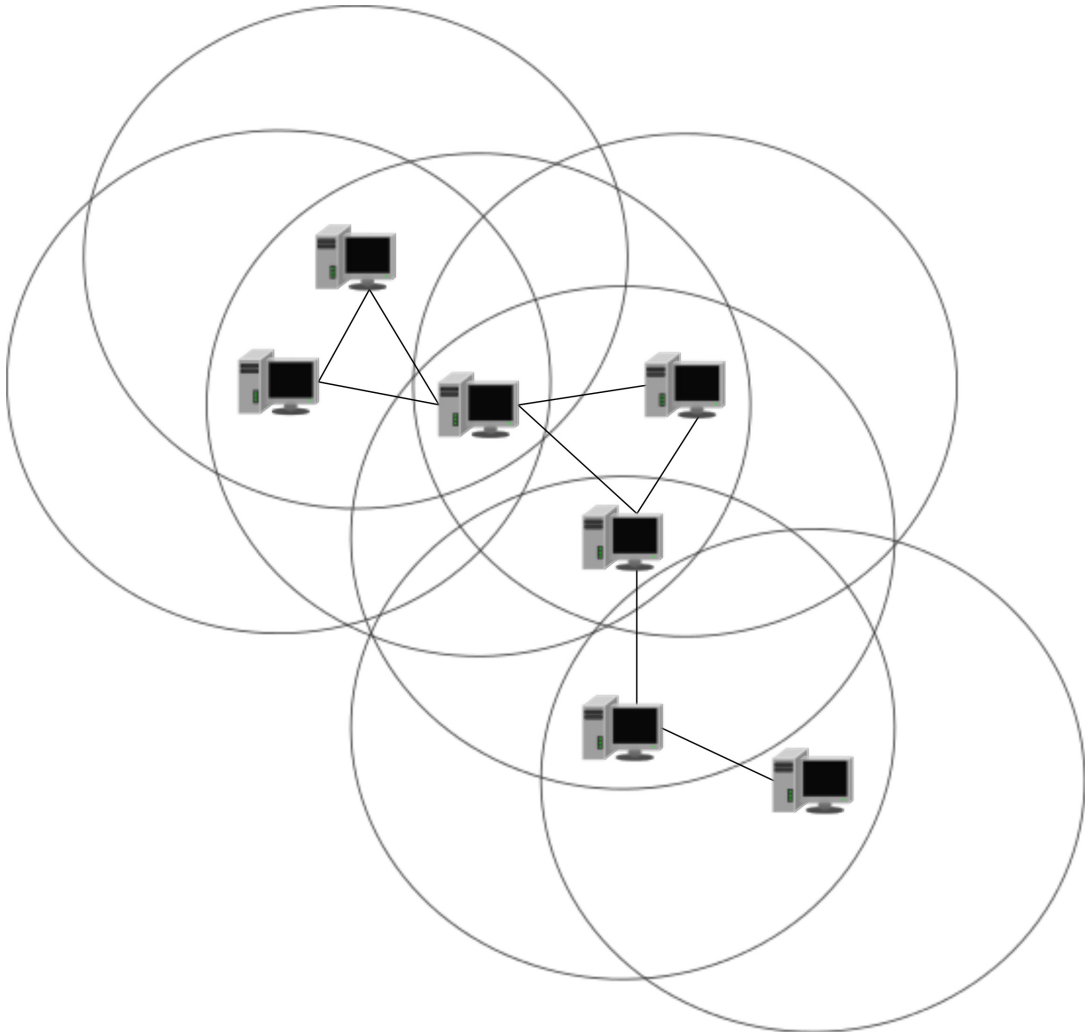
[1] Adapting blackhat approaches t
scenarios", MS Thesis, TUM, 2

Unstructured Methods

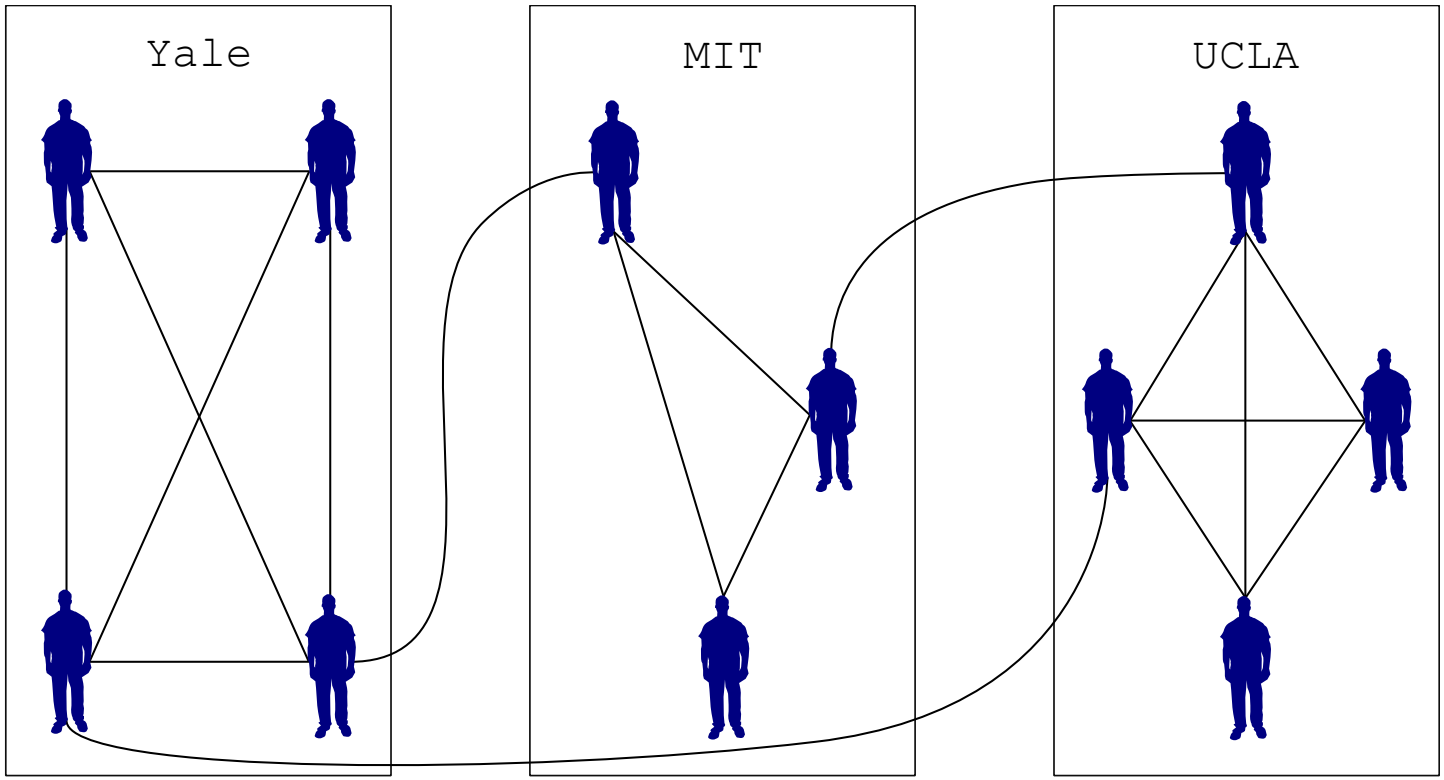
- Sample and Collide
- Hop Sampling
- Gossip-based aggregation
- Gossipico

Unstructured Networks — Physical

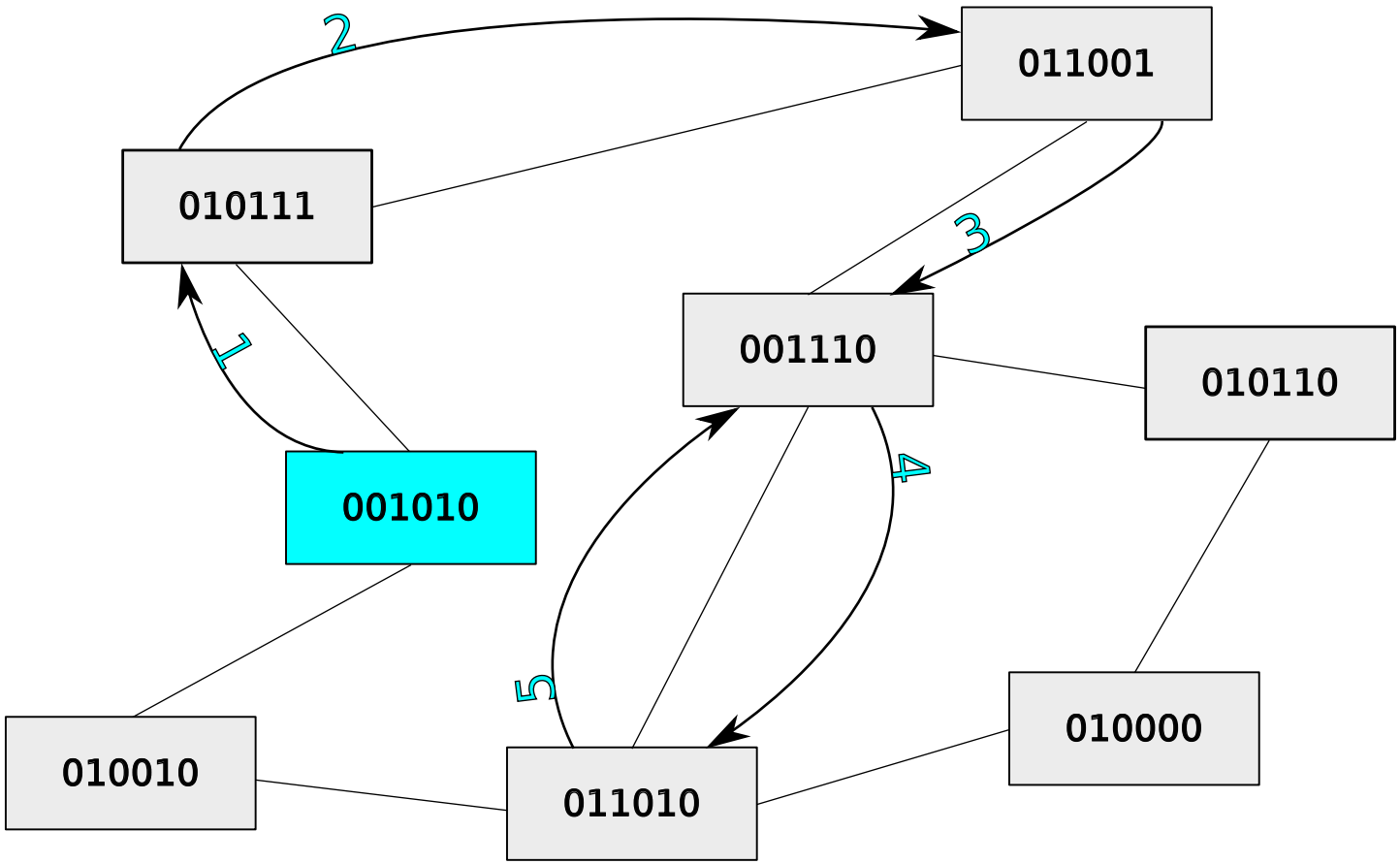
Unstructured Networks — Wireless



Unstructured Networks — Social



Sample and Collide [3]



Sample and Collide

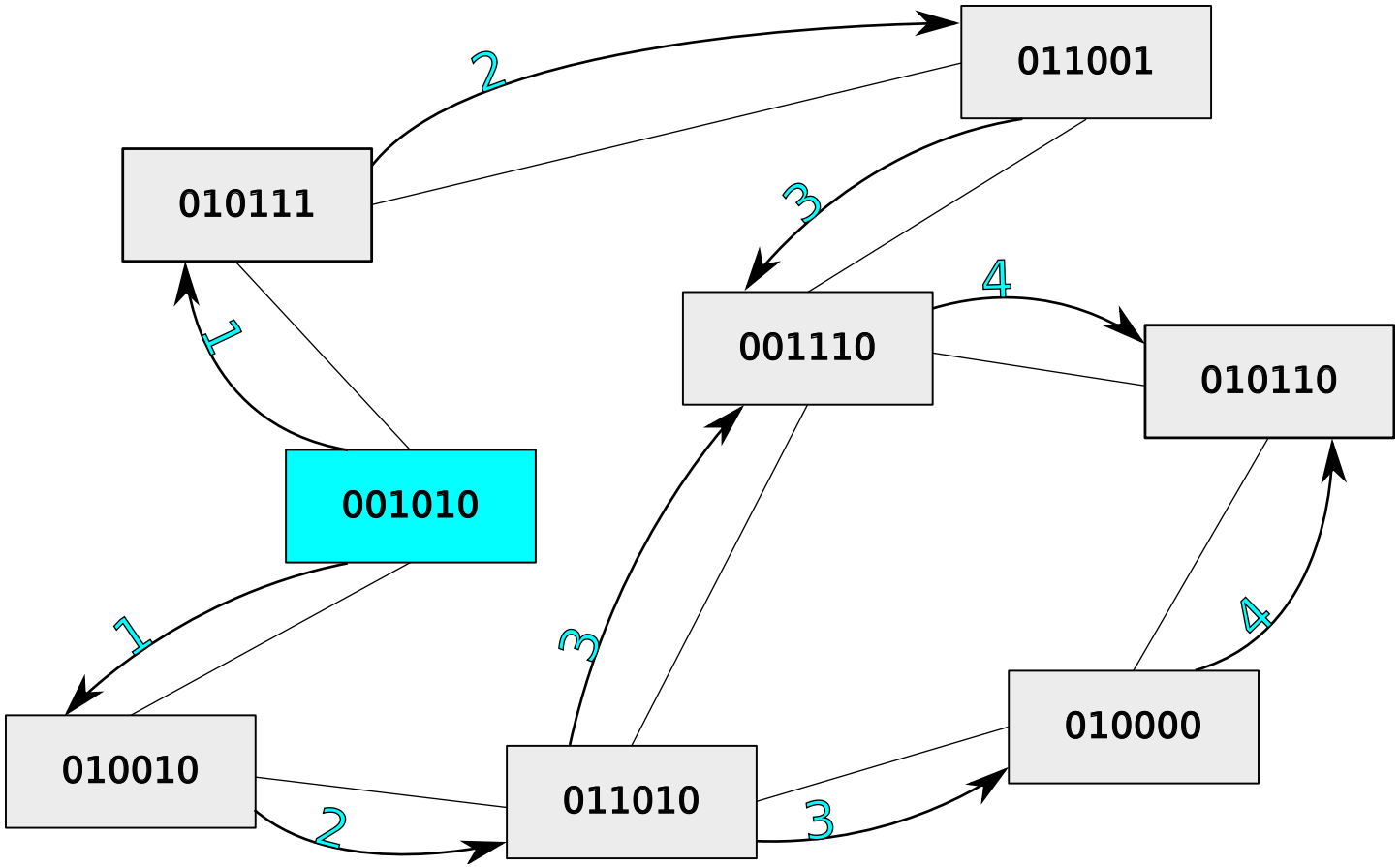
Advantages

- $O(\sqrt{n})$ messages (message size: $O(\sqrt{n})$ or worse?)

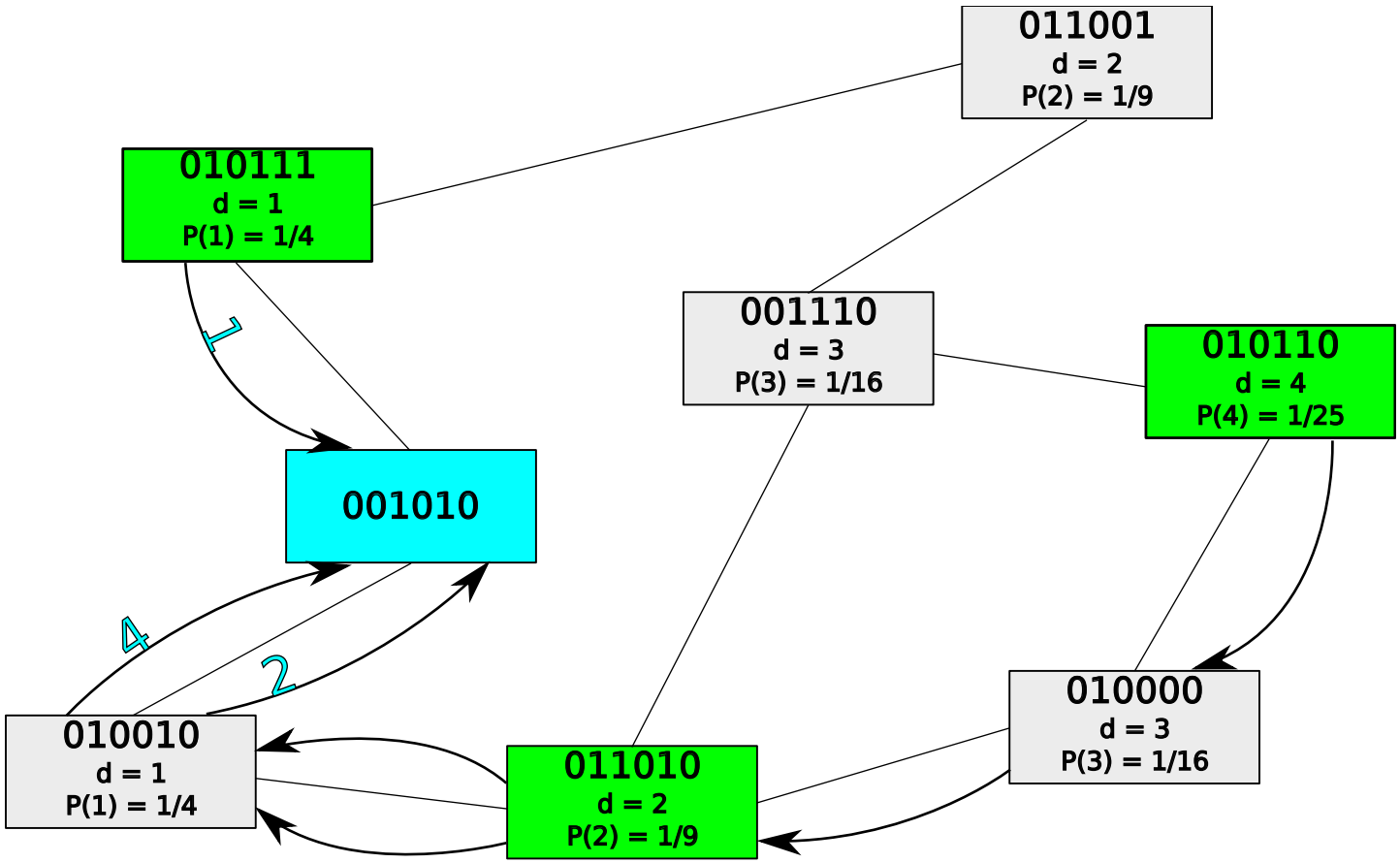
Disadvantages

- Message size could be up to $O(\sqrt{n})$ depending on the topology
- Fails in some topologies (ring, highly clustered networks) depending on sampling method
- Centralized or each node has do the estimate, complexity in that case is $O(n^2)$ per round
- Design fails to address denial-of-service potential
- m malicious participants can force size estimate of m^2

Hop Sampling [2]



Hop Sampling



Hop Sampling

Advantages

- Works with all topologies
- $O(|E|)$ messages (message size: $O(1)$)

Disadvantages

- High load on neighbors of initiator
- If each node needs an estimate: $O(|N| \cdot |E|)$ per round
- If not centralized, each node has to keep $O(|N|)$ state to track distance to origin
- Design fails to address denial-of-service potential
- Tiny fraction of malicious participants can always create significant size overestimates

Gossip-based aggregation [1]

- One node starts with a value of $v_i = 1$, all others with $v_o = 0$
- Select a random edge (A, B) and update values to $\frac{v_A + v_B}{2}$
- If node A leaves, set $v_B := v_B + v_A$ for some neighbour B of A

$\Rightarrow$ Value globally converges to $\frac{1}{|N|}$

Gossip-based aggregation

Advantages

- Work with all topologies
- Network wide agreement
- $O(|N|)$ messages in total (message size: $O(1)$)

Disadvantages

- Start point agreement problem
- Slow convergence
- Very vulnerable to denial-of-service
- Very vulnerable to result manipulation
- One malicious participant affect the whole network

Gossipico [5]

- New gossip-based network counting algorithm (2012)
- Significantly higher precision than previous algorithms
- Significantly better performance than previous algorithms
- Uses "count" to accumulate number of peers in network
- Uses "beacons" to structure counting message propagation

Count

Count does not Scale

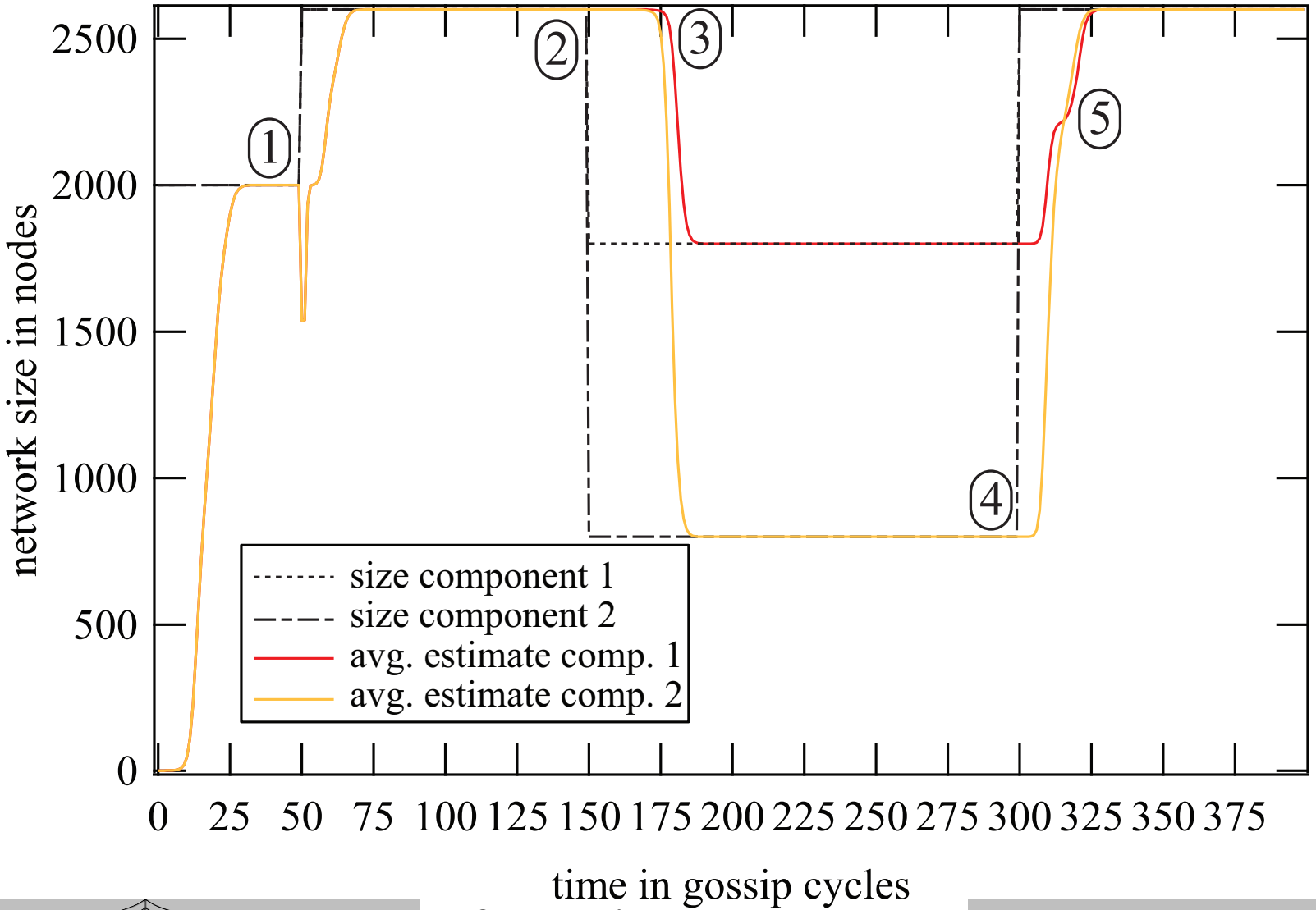
Spreading the Result

Beacons

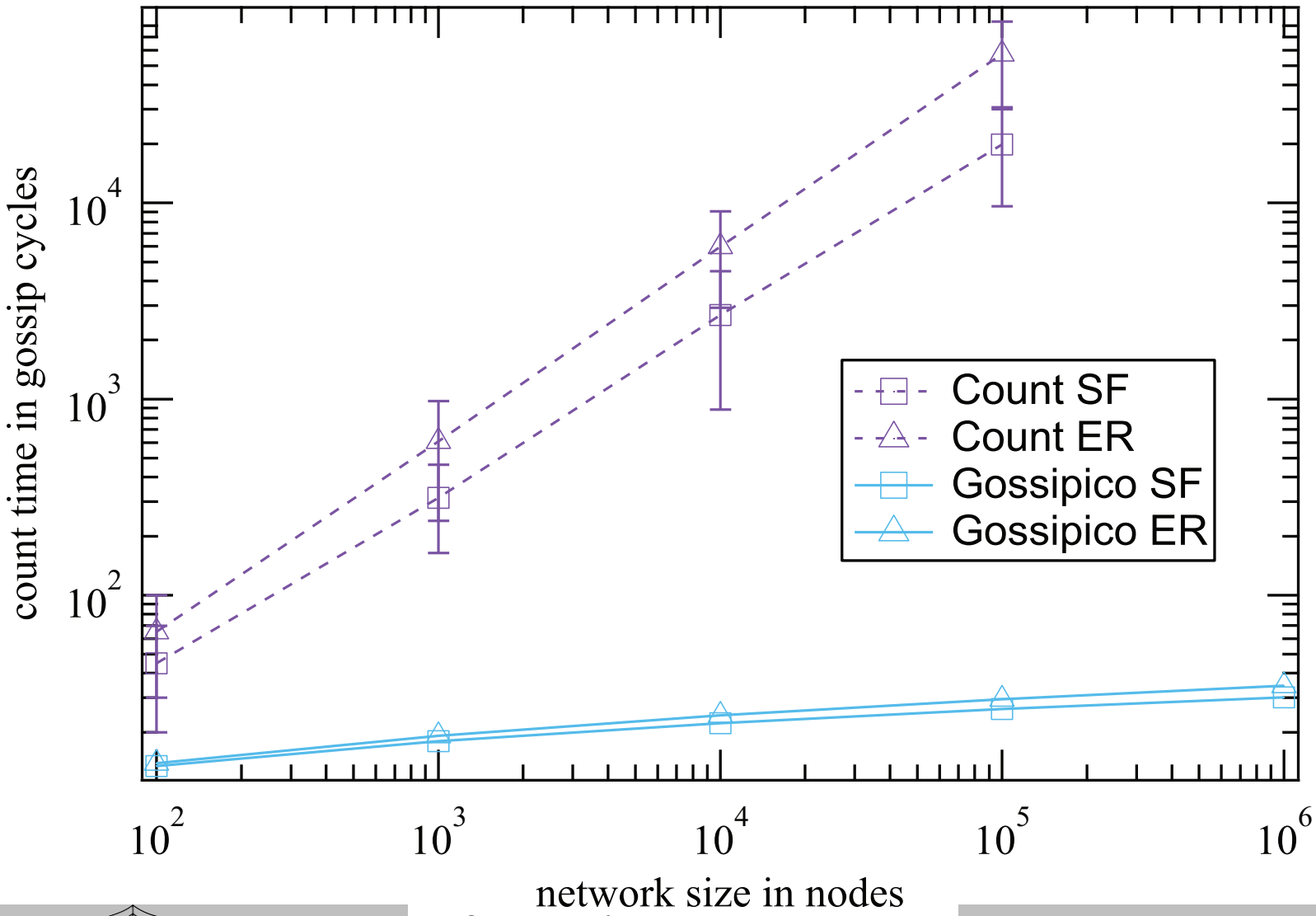
Towards Shortest Paths

Topology Changes

Gossipico Precision



Gossipico Performance



Gossipico aggregation

Advantages

- Quite efficient
- Very good precision
- No designated node to start the process
- Supports churn

Disadvantages

- Not secure against denial-of-service attacks
- Malicious participants can change result to any value

Network Size Estimation in GNUnet [2]

Functional Goals

- Supports churn
- Fully decentralized
- Efficient
- All peers obtain the network size estimate
- Operates in unstructured topologies

[2] Evans, Polot, Grothoff: "Efficient and Secure Decentralized Network Size Estimation", Networking 2012

Intuitive Idea

- Set of elements distributed in a space
- Pick a random spot
- Measure distance to nearest element
- More elements $\Rightarrow$ smaller distance, more *overlapping*

Intuitive Idea

Intuitive Idea

Intuitive Idea

Intuitive Idea

Intuitive Idea - Applied to networks

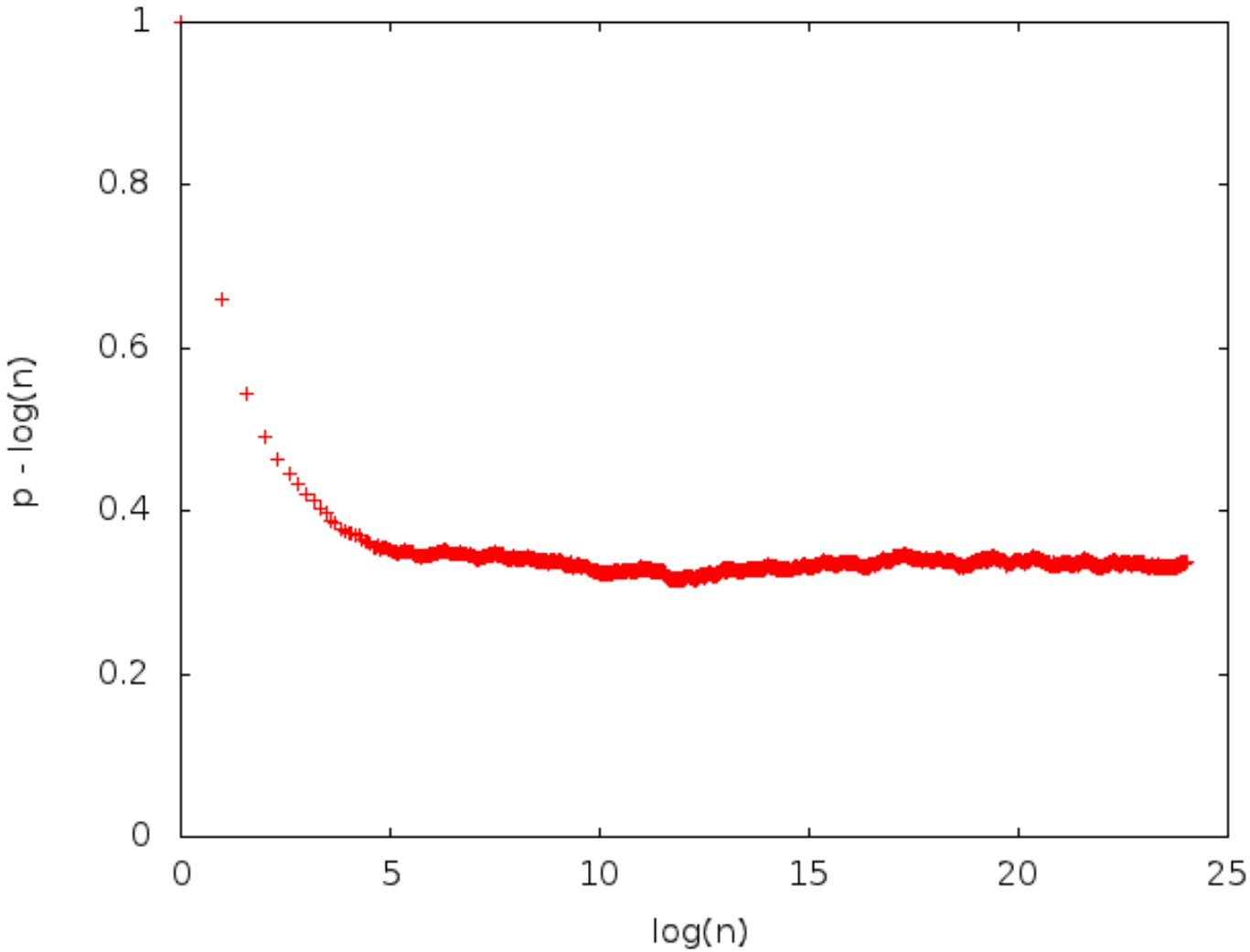
- Space: all possible IDs
- Population: randomly distributed peer IDs
- Overlap: number of leading bits in common with a random ID

Theorem

Let $\overline{p}$ be the expected maximum number of leading overlapping bits between all n random node identifiers in the network and a random key. Then the network size n is approximately

$$2^{\overline{p}-0.332747}$$

Empirical Measurement of the 0.33... correction



Proof (1/3)

Let X be the random variable for all n identifiers and let X_i be the number of overlapping bits for an individual random node identifier i.

The probability that a single random node identifier i overlaps with at least α bits with a random key is

$$P(X_i \geq \alpha) = 2^{-\alpha}. \tag{1}$$

Then, the probability that a single random node identifier overlaps with less than α bits with a random key is

$$P(X_i < \alpha) = 1 - 2^{-\alpha}. \tag{2}$$

The probability that the maximum number of leading overlapping bits for all n random nodes is strictly less than α is

$$P_n(X < \alpha) :=: P\left(\bigwedge_i X_i < \alpha\right) = (P(X_i < \alpha))^n = \left(1 - 2^{-\alpha}\right)^n.$$

Proof (2/3)

Then $E_n(X)$, the expected maximum number of leading overlapping bits between n random node identifiers in the network is:

$$
\begin{aligned}
E_n(X) :=: \sum_{\alpha=0}^{\infty} \alpha \cdot P_n(X = \alpha) &= \sum_{\alpha=1}^{\infty} P_n(X \geq \alpha) \\
&= \sum_{\alpha=1}^{\infty} (1 - P_n(X < \alpha)) = \sum_{\alpha=1}^{\infty} \left(1 - \left(1 - 2^{-\alpha}\right)^n\right) \\
&= \sum_{\alpha=1}^{\log_2 n} \left(1 - \left(1 - 2^{-\alpha}\right)^n\right) + \sum_{\alpha=\log_2 n+1}^{\infty} \left(1 - \left(1 - 2^{-\alpha}\right)^n\right)
\end{aligned}
$$

Suppose n is sufficiently large such that we can use $\lim_{n\to\infty}(1 - \frac{x}{n})^n = e^{-x}$.

Proof (3/3)

By substituting $\beta := \alpha - \log_2 n$ and $\gamma := \log_2 n - \alpha$ we then get:

$$E_n(X) = \log_2 n - \sum_{\gamma=0}^{\log_2 n-1} \left(1 - 2^{\gamma-\log_2 n}\right)^n + \sum_{\beta=1}^{\infty} \left(1 - \left(1 - 2^{-(\beta+\log_2 n)}\right)^n\right.$$

$$= \log_2 n - \sum_{\gamma=0}^{\log_2 n-1} \left(1 - \frac{2^\gamma}{n}\right)^n + \sum_{\beta=1}^{\infty} \left(1 - \left(1 - \frac{2^{-\beta}}{n}\right)^n\right)$$

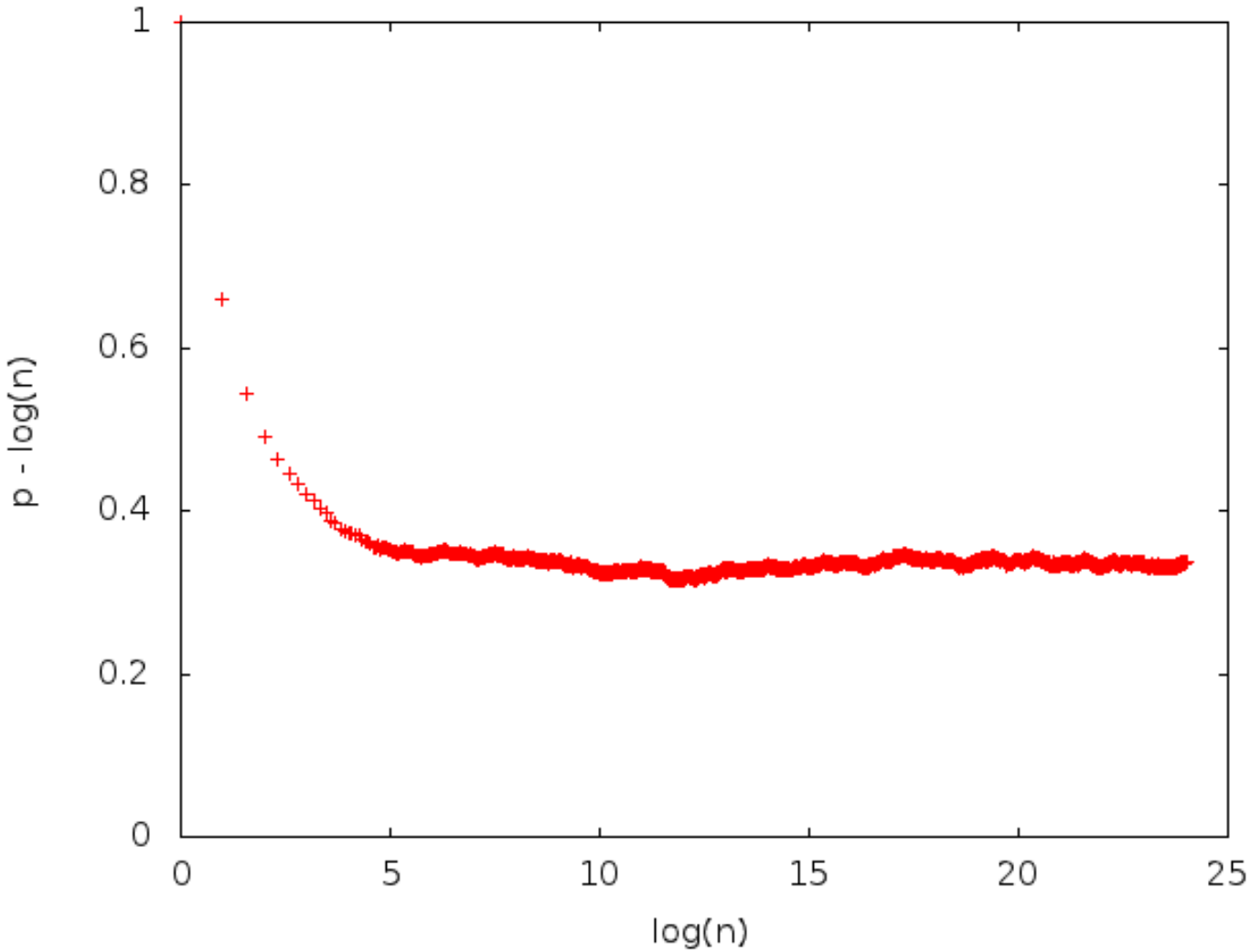
$$\approx \log_2 n - \sum_{\gamma=0}^{\log_2 n-1} e^{-2^\gamma} + \sum_{\beta=1}^{\infty} \left(1 - e^{2^{-\beta}}\right)$$

$$\approx \log_2 n - 0.521865 + 0.854613 = \log_2 n + 0.332747$$

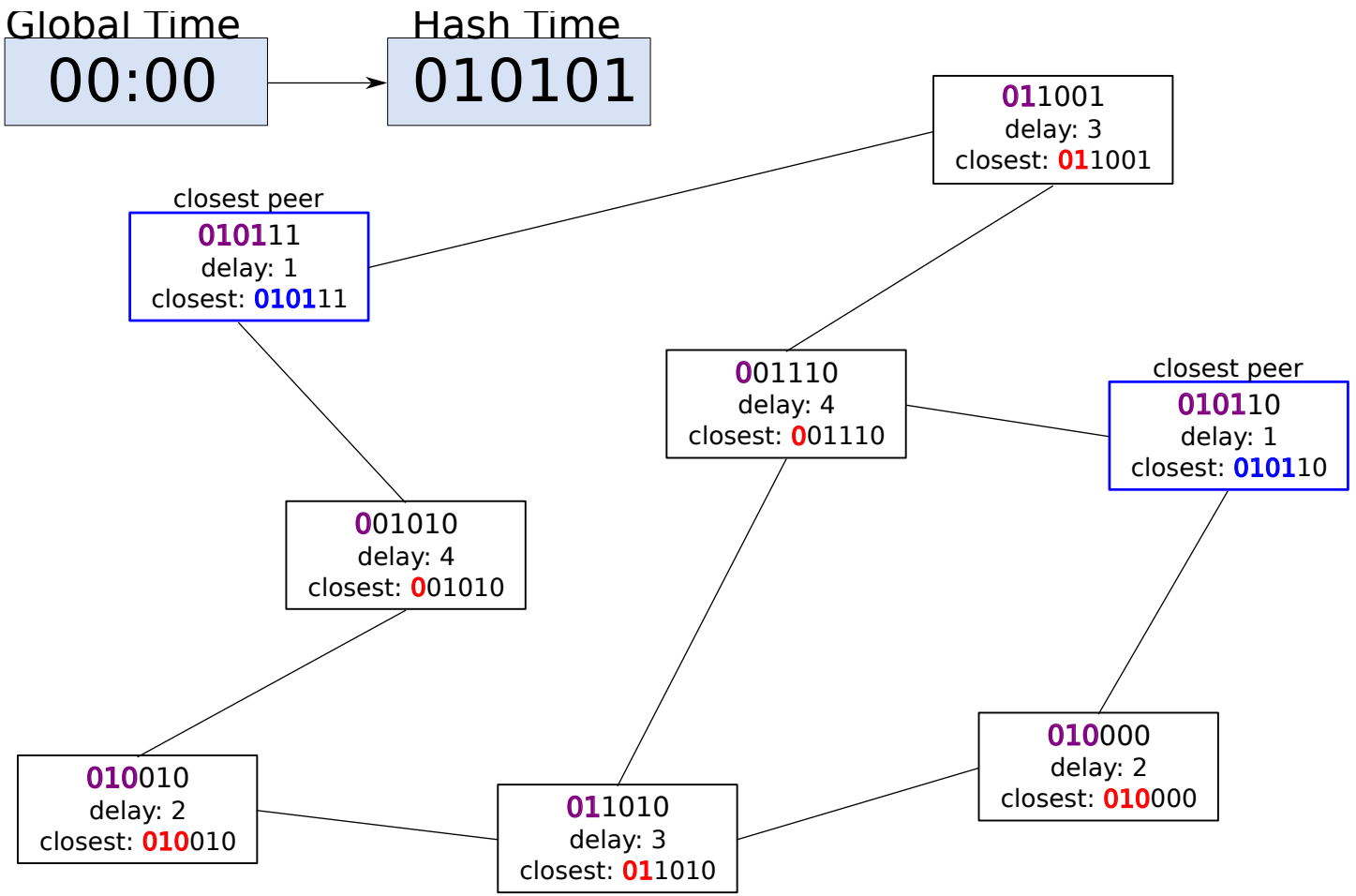
Thus, for sufficiently large values of n,

$$E_n(X) \approx \log_2 n + 0.332747. \quad (3)$$

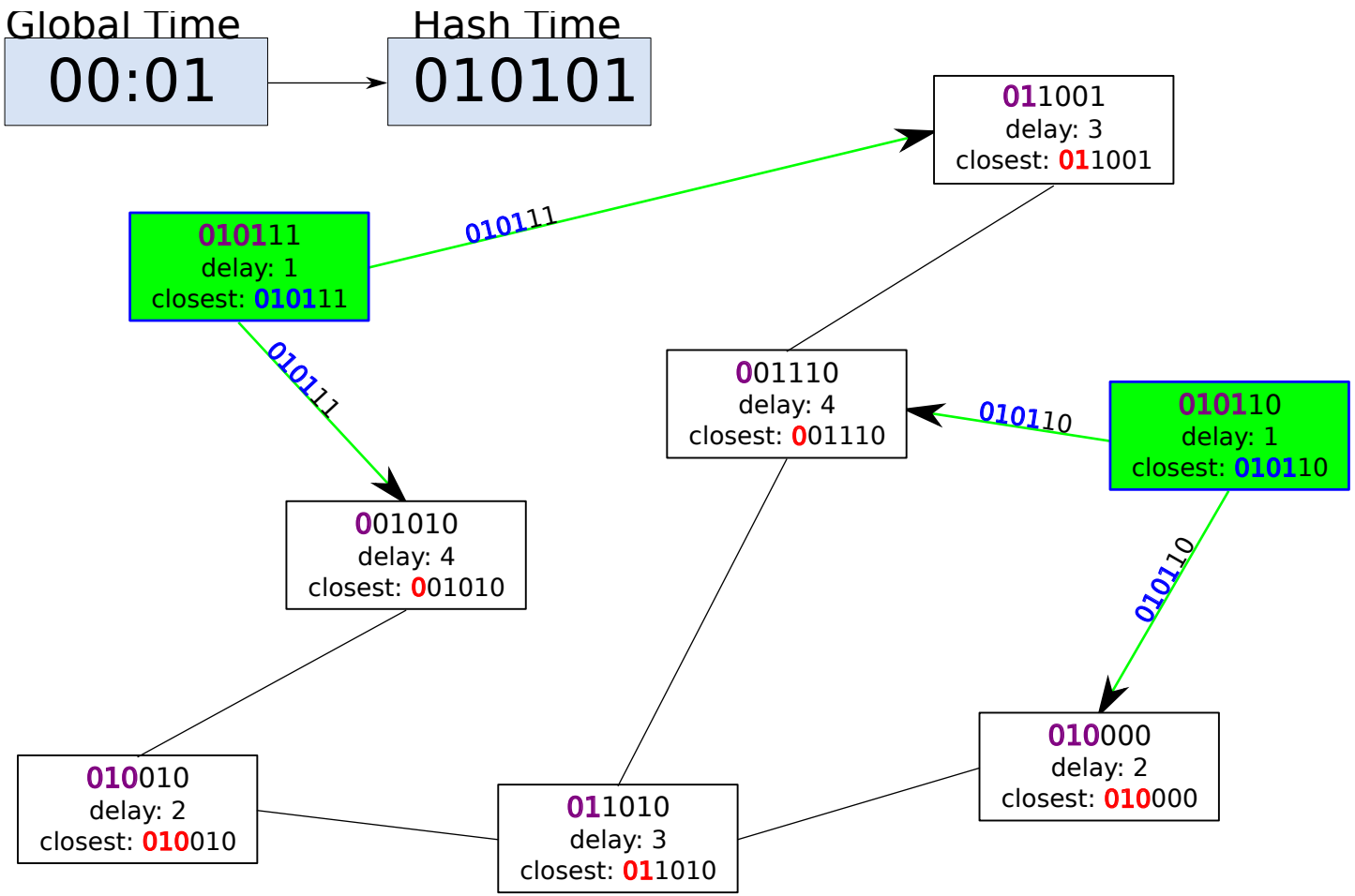
Empirical Measurement of the 0.33... correction



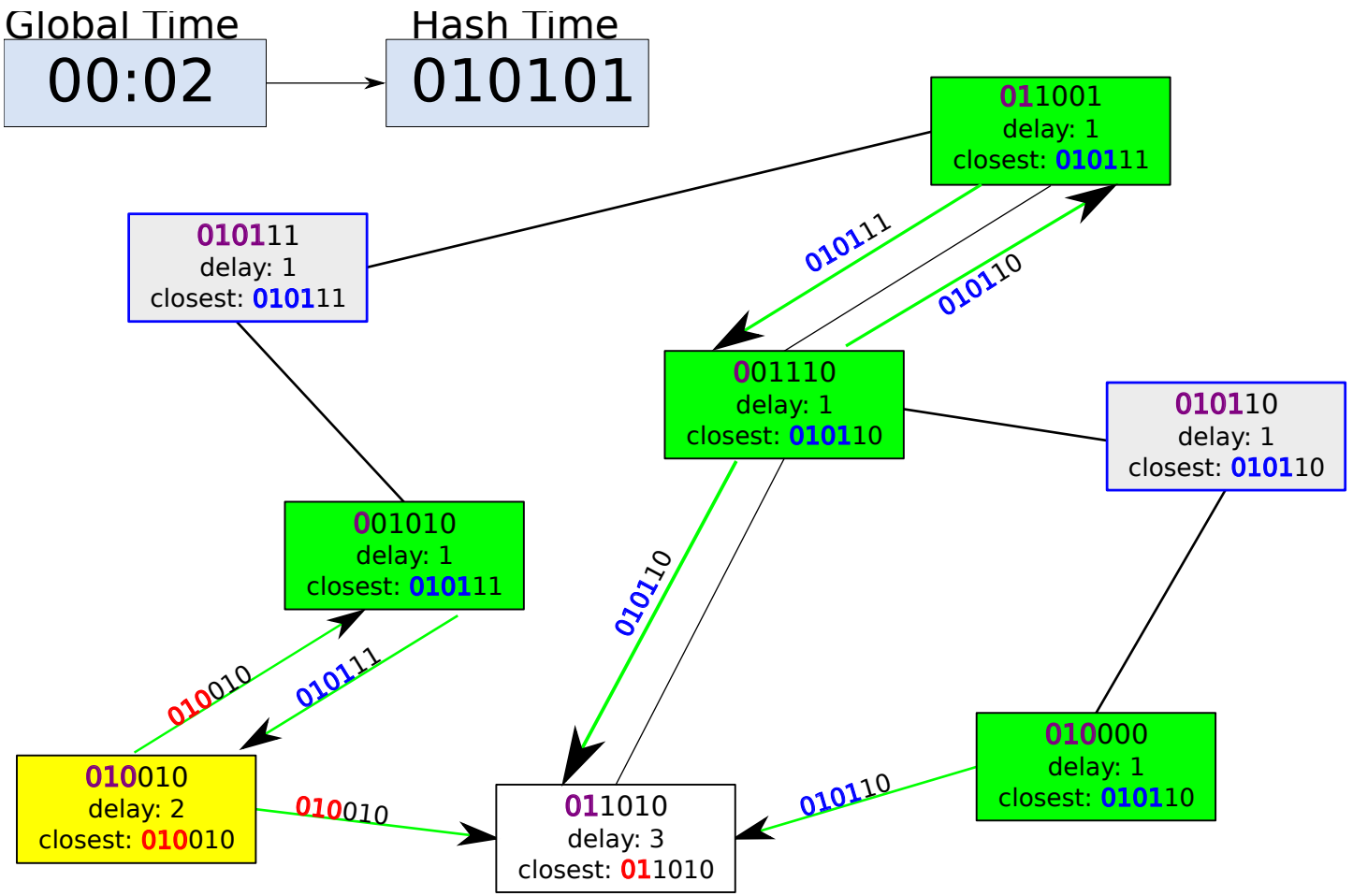
Time: 0


Global Time
00:00
Hash Time
010101
011001
delay: 3
closest: 011001
closest peer
010111
delay: 1
closest: 010111
001110
delay: 4
closest: 001110
closest peer
010110
delay: 1
closest: 010110
001010
delay: 4
closest: 001010
010010
delay: 2
closest: 010010
011010
delay: 3
closest: 011010
010000
delay: 2
closest: 010000

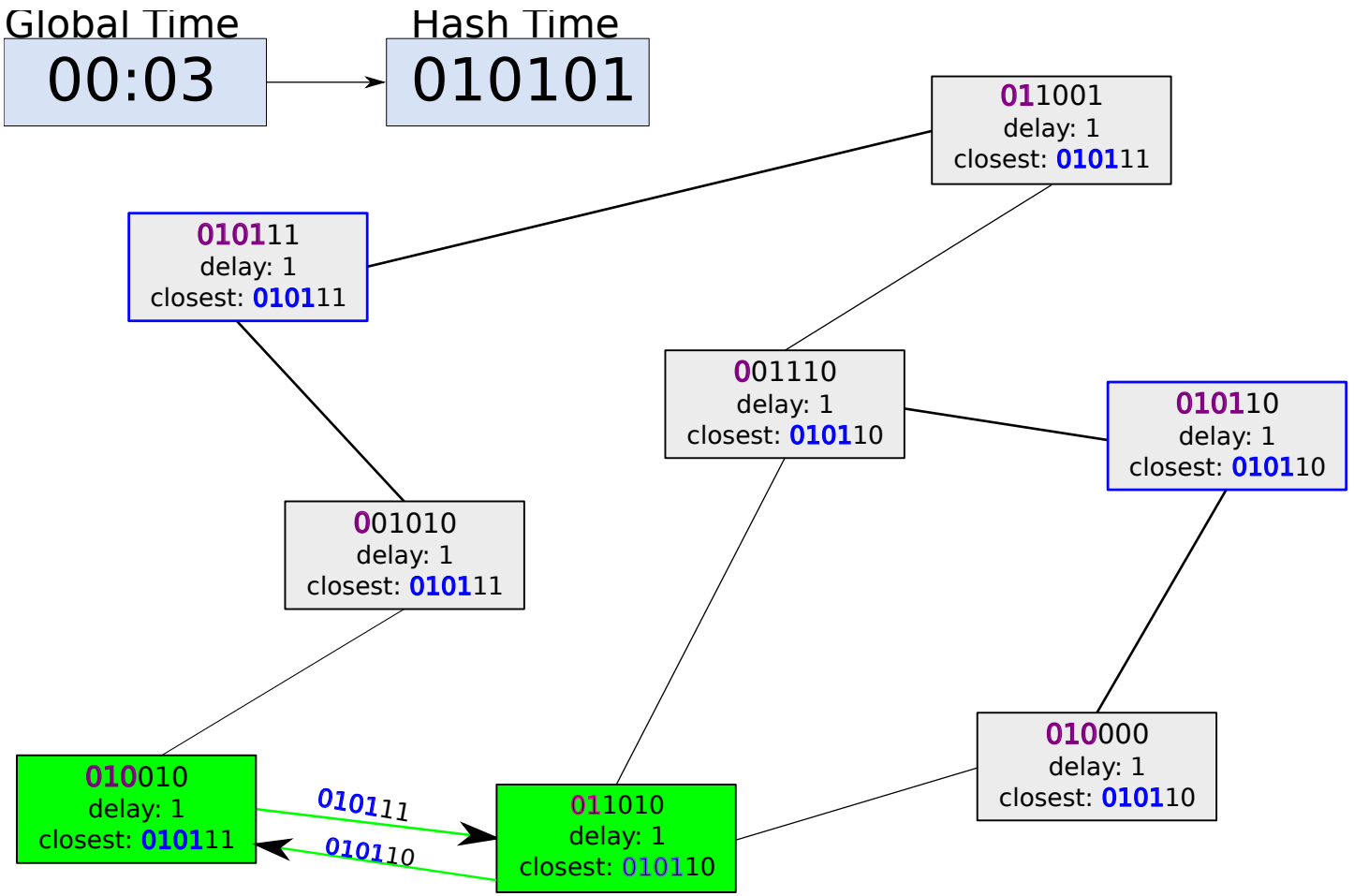
Time: 1


Global Time
00:01
Hash Time
010101
011001
delay: 3
closest: 011001
010111
delay: 1
closest: 010111
010111
010111
001110
delay: 4
closest: 001110
010110
010110
delay: 1
closest: 010110
001010
delay: 4
closest: 001010
010110
010010
delay: 2
closest: 010010
011010
delay: 3
closest: 011010
010000
delay: 2
closest: 010000

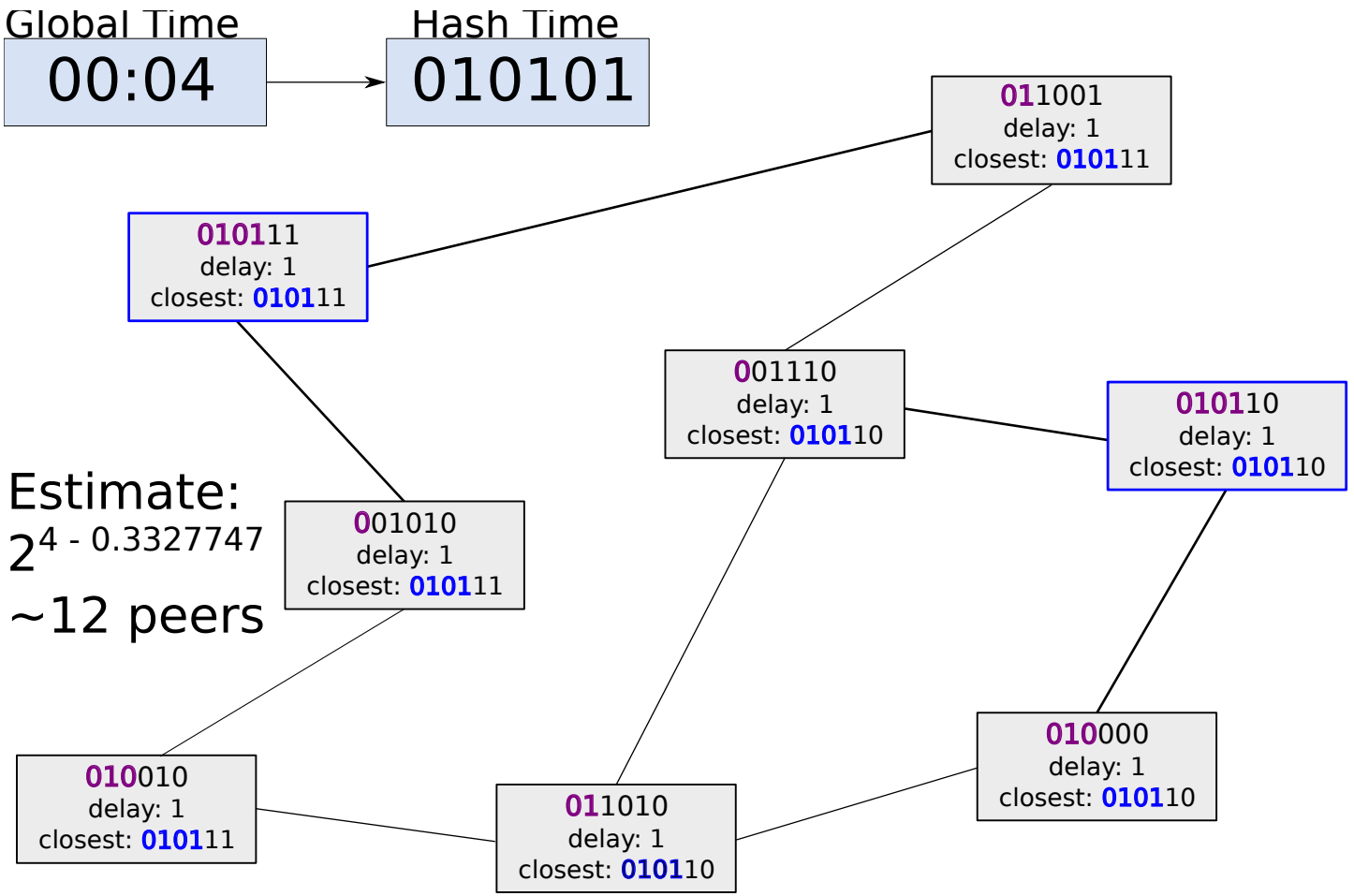
Time: 2

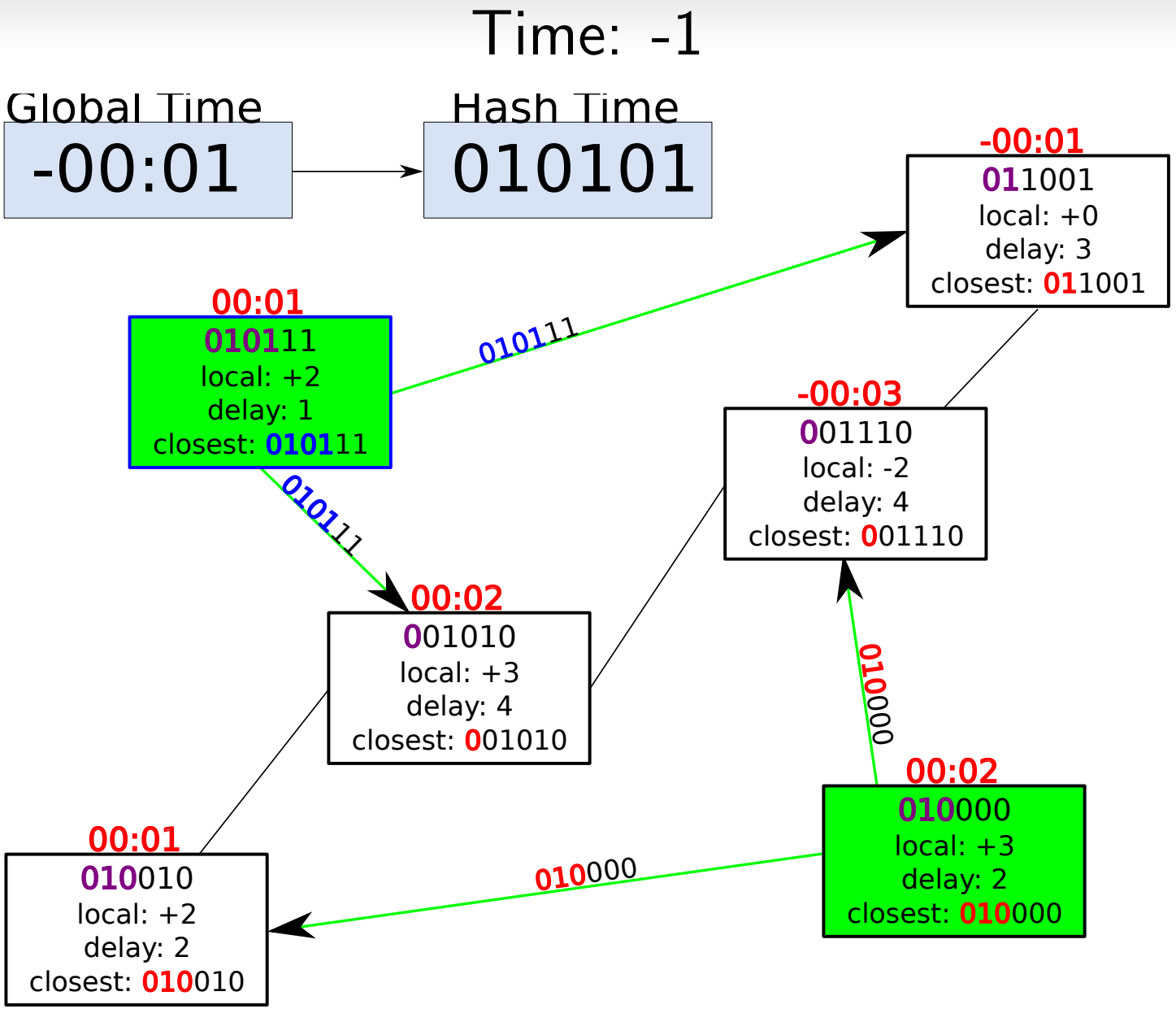

Global Time
00:02
Hash Time
010101
011001
delay: 1
closest: 010111
010111
delay: 1
closest: 010111
010111
010110
001110
delay: 1
closest: 010110
010110
delay: 1
closest: 010110
001010
delay: 1
closest: 010111
010110
010010
010111
010010
delay: 2
closest: 010010
010010
011010
delay: 3
closest: 011010
010110
010000
delay: 1
closest: 010110

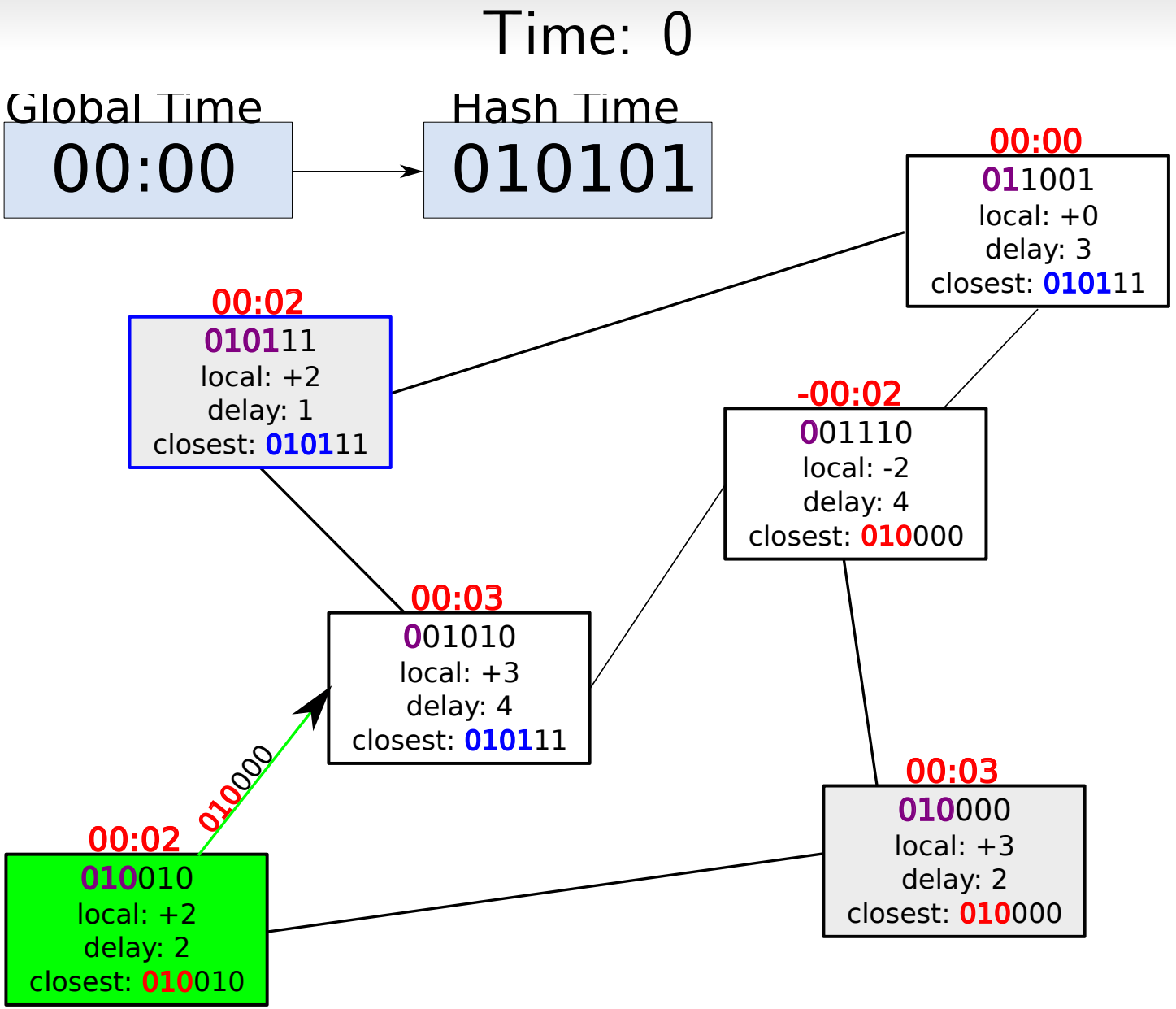
Time: 3

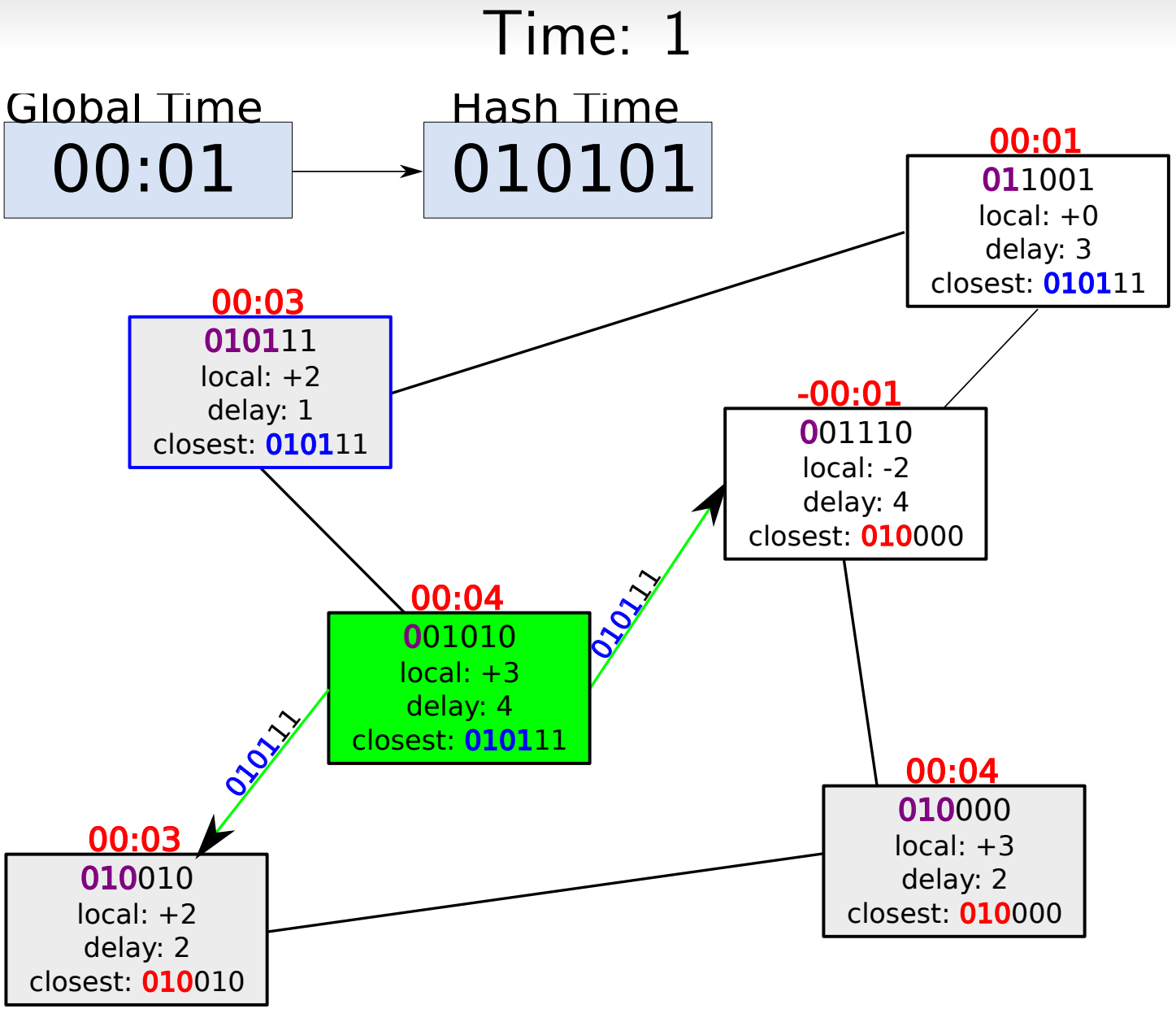

Global Time
00:03
Hash Time
010101
011001
delay: 1
closest: 010111
010111
delay: 1
closest: 010111
001110
delay: 1
closest: 010110
010110
delay: 1
closest: 010110
001010
delay: 1
closest: 010111
010010
delay: 1
closest: 010111
010111
010110
011010
delay: 1
closest: 010110
010000
delay: 1
closest: 010110

Time: 4

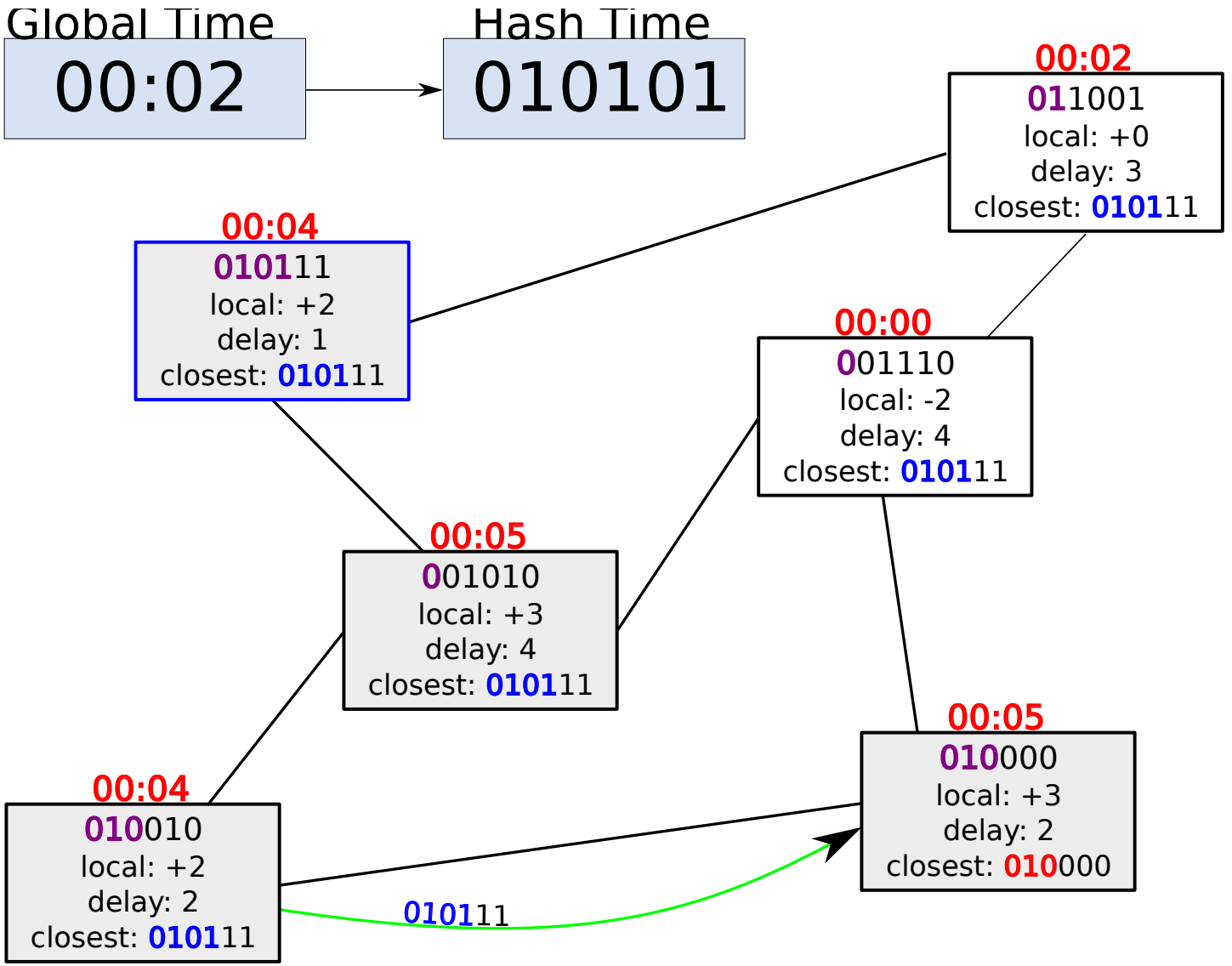

Global Time
00:04
Hash Time
010101
011001
delay: 1
closest: 010111
010111
delay: 1
closest: 010111
001110
delay: 1
closest: 010110
010110
delay: 1
closest: 010110
Estimate:
2^4 - 0.3327747
~12 peers
001010
delay: 1
closest: 010111
010010
delay: 1
closest: 010111
011010
delay: 1
closest: 010110
010000
delay: 1
closest: 010110


Time: -1
Global Time
-00:01
Hash Time
010101
-00:01
011001
local: +0
delay: 3
closest: 011001
00:01
010111
local: +2
delay: 1
closest: 010111
010111
-00:03
001110
local: -2
delay: 4
closest: 001110
010111
00:02
001010
local: +3
delay: 4
closest: 001010
010000
00:02
010000
local: +3
delay: 2
closest: 010000
00:01
010010
local: +2
delay: 2
closest: 010010
010000

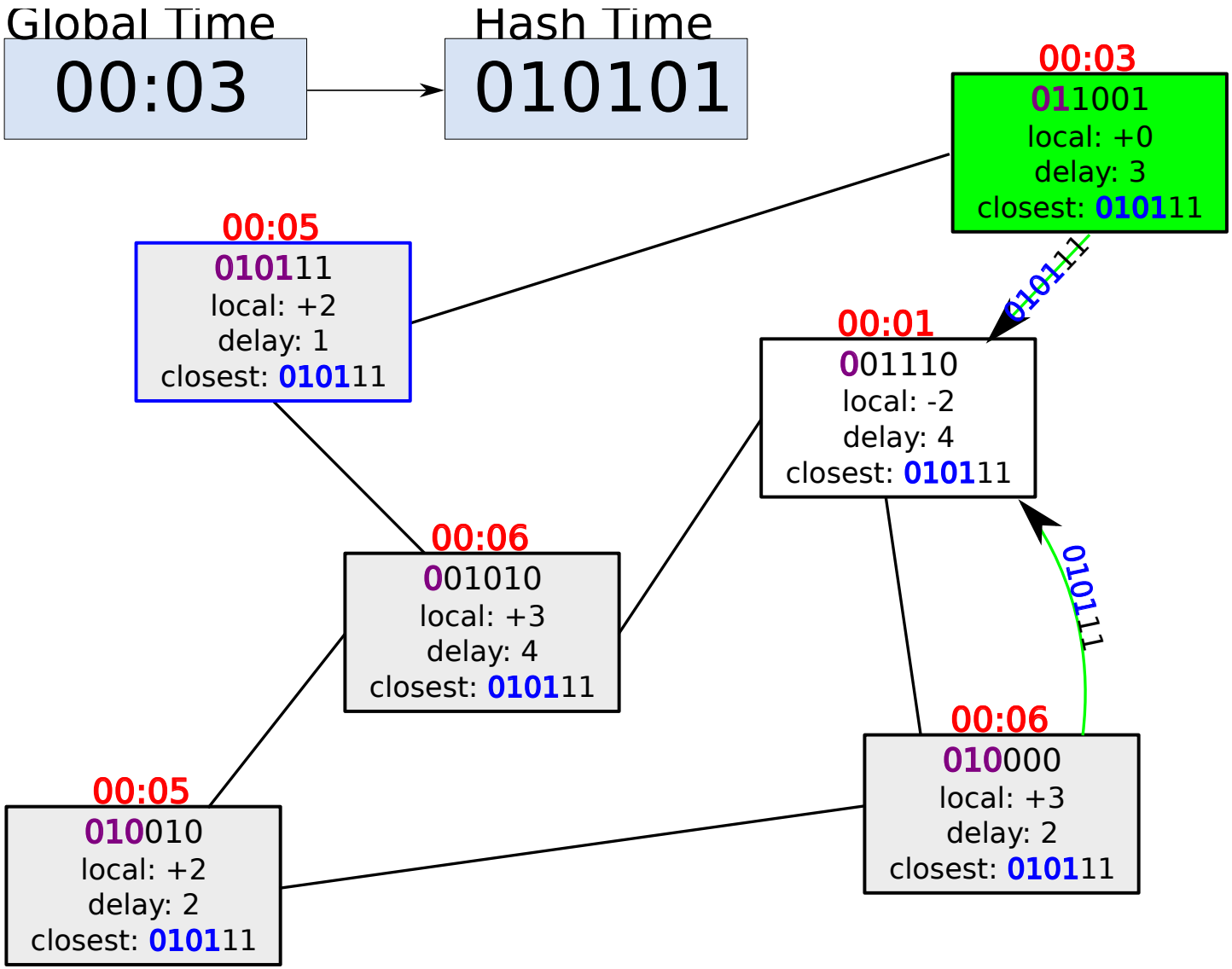

Time: 0
Global Time
00:00
Hash Time
010101
00:00
011001
local: +0
delay: 3
closest: 010111
00:02
010111
local: +2
delay: 1
closest: 010111
-00:02
001110
local: -2
delay: 4
closest: 010000
00:03
001010
local: +3
delay: 4
closest: 010111
010000
00:03
010000
local: +3
delay: 2
closest: 010000
00:02
010010
local: +2
delay: 2
closest: 010010


Time: 1
Global Time
00:01
Hash Time
010101
00:01
011001
local: +0
delay: 3
closest: 010111
00:03
010111
local: +2
delay: 1
closest: 010111
-00:01
001110
local: -2
delay: 4
closest: 010000
00:04
001010
local: +3
delay: 4
closest: 010111
010111
010111
00:04
010000
local: +3
delay: 2
closest: 010000
00:03
010010
local: +2
delay: 2
closest: 010010

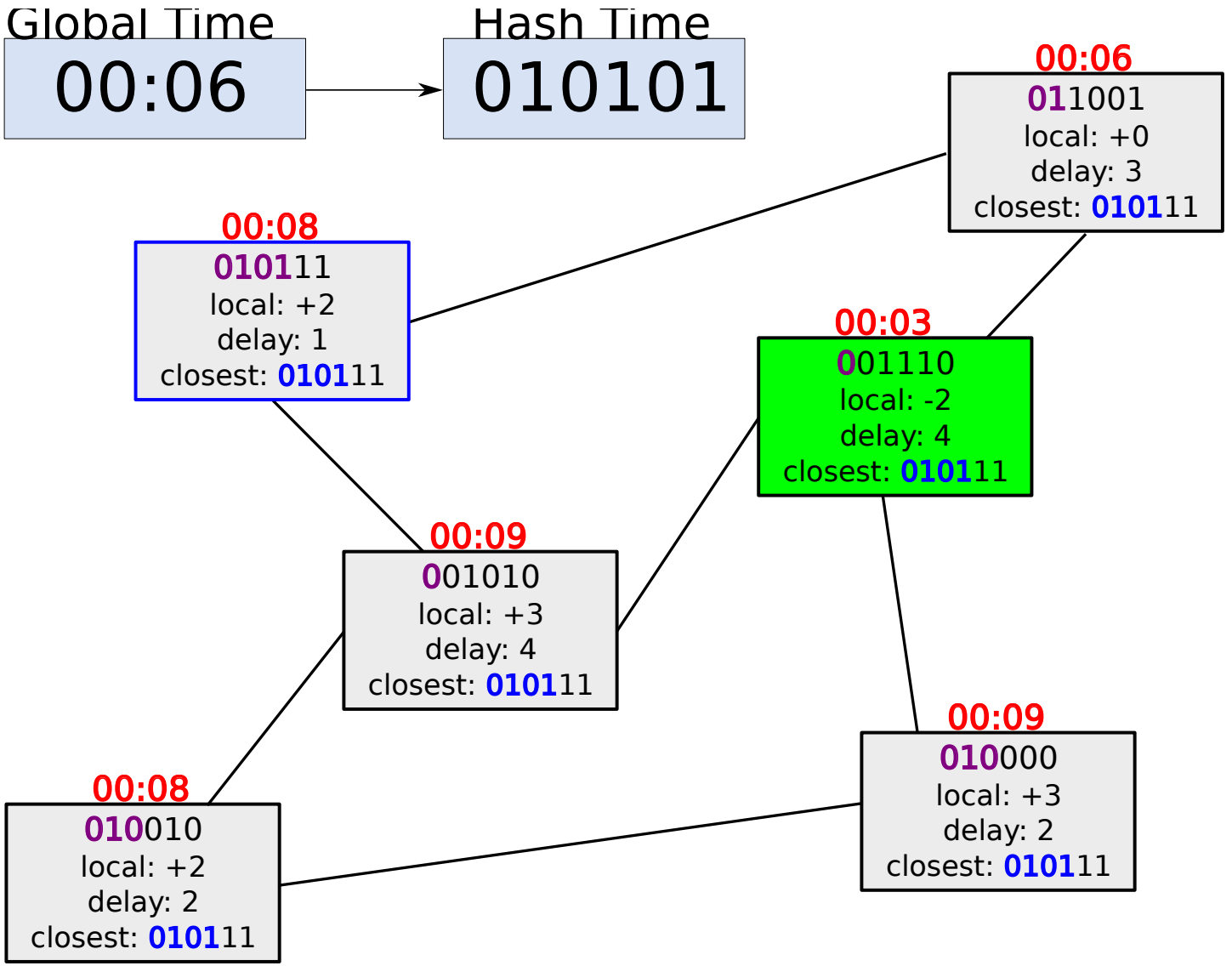
Time: 2


Global Time
00:02
Hash Time
010101
00:02
011001
local: +0
delay: 3
closest: 010111
00:04
010111
local: +2
delay: 1
closest: 010111
00:00
001110
local: -2
delay: 4
closest: 010111
00:05
001010
local: +3
delay: 4
closest: 010111
00:05
010000
local: +3
delay: 2
closest: 010000
00:04
010010
local: +2
delay: 2
closest: 010111
010111

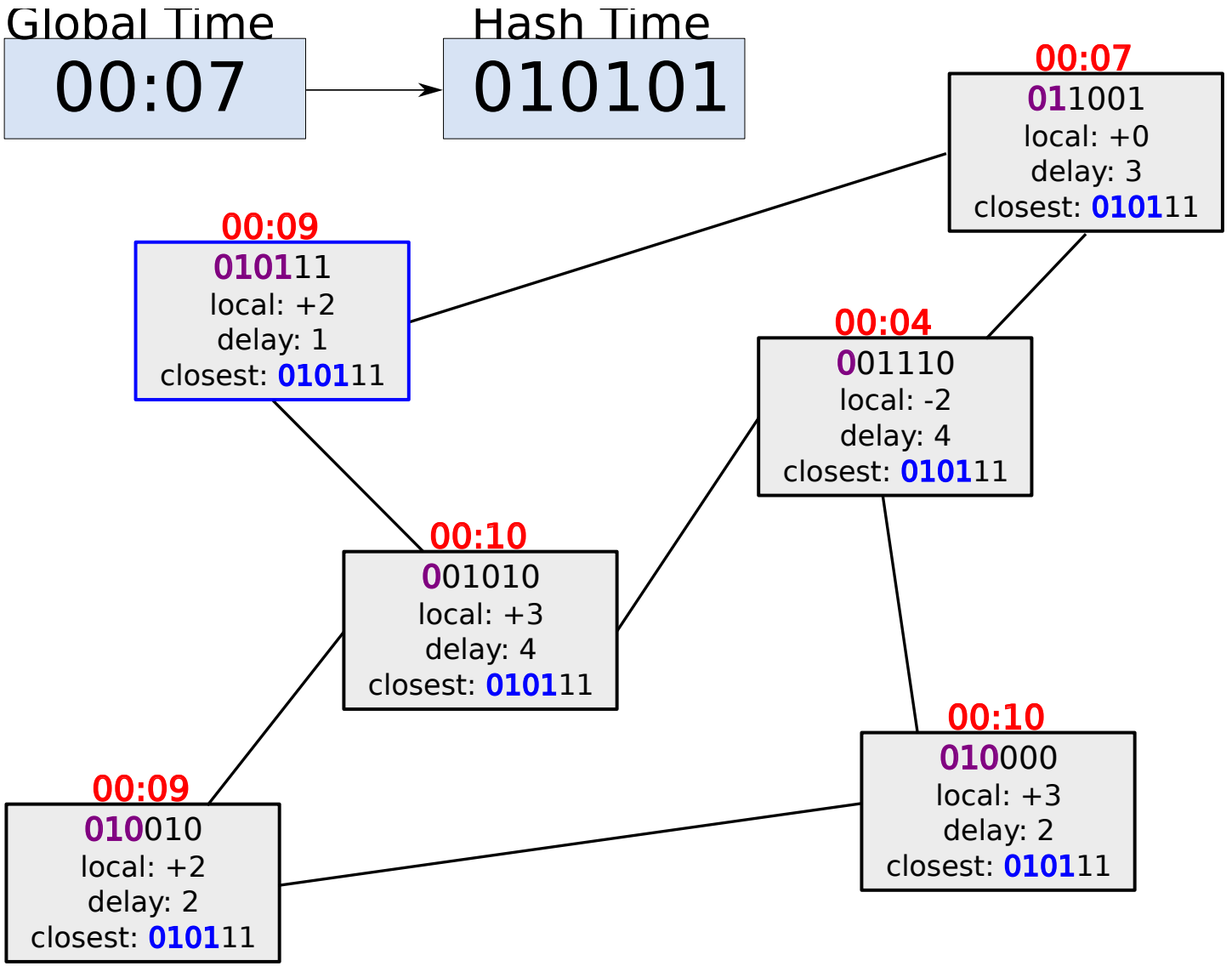
Time: 3


Global Time
00:03
Hash Time
010101
00:03
011001
local: +0
delay: 3
closest: 010111
00:05
010111
local: +2
delay: 1
closest: 010111
010111
00:01
001110
local: -2
delay: 4
closest: 010111
00:06
001010
local: +3
delay: 4
closest: 010111
010111
00:06
010000
local: +3
delay: 2
closest: 010111
00:05
010010
local: +2
delay: 2
closest: 010111

Time: 6


Global Time
00:06
Hash Time
010101
00:06
011001
local: +0
delay: 3
closest: 010111
00:08
010111
local: +2
delay: 1
closest: 010111
00:03
001110
local: -2
delay: 4
closest: 010111
00:09
001010
local: +3
delay: 4
closest: 010111
00:09
010000
local: +3
delay: 2
closest: 010111
00:08
010010
local: +2
delay: 2
closest: 010111

Time: 7


Global Time
00:07
Hash Time
010101
00:07
011001
local: +0
delay: 3
closest: 010111
00:09
010111
local: +2
delay: 1
closest: 010111
00:04
001110
local: -2
delay: 4
closest: 010111
00:10
001010
local: +3
delay: 4
closest: 010111
00:10
010000
local: +3
delay: 2
closest: 010111
00:09
010010
local: +2
delay: 2
closest: 010111

Our Approach: Key Points

- Use the current time to generate a random number
- More overlapping bits $\Rightarrow$ gossip earlier
- Also delay gossip randomly to avoid traffic spikes
- Proof-of-Work to make Sybil attacks harder

Message Format

Offset	Contents
0	Message header magic code
4	Hop-Count (updated at each peer)
8	Signed data header magic code
16	Time S of the round
24	Proximity p in bits
28	Public key (2048 bit RSA)
288	Proof-of-work
296	Signature (signing bytes 8–295)

Security

Security Properties

- No trusted third parties
- Reliable
- Resistant to malicious participants

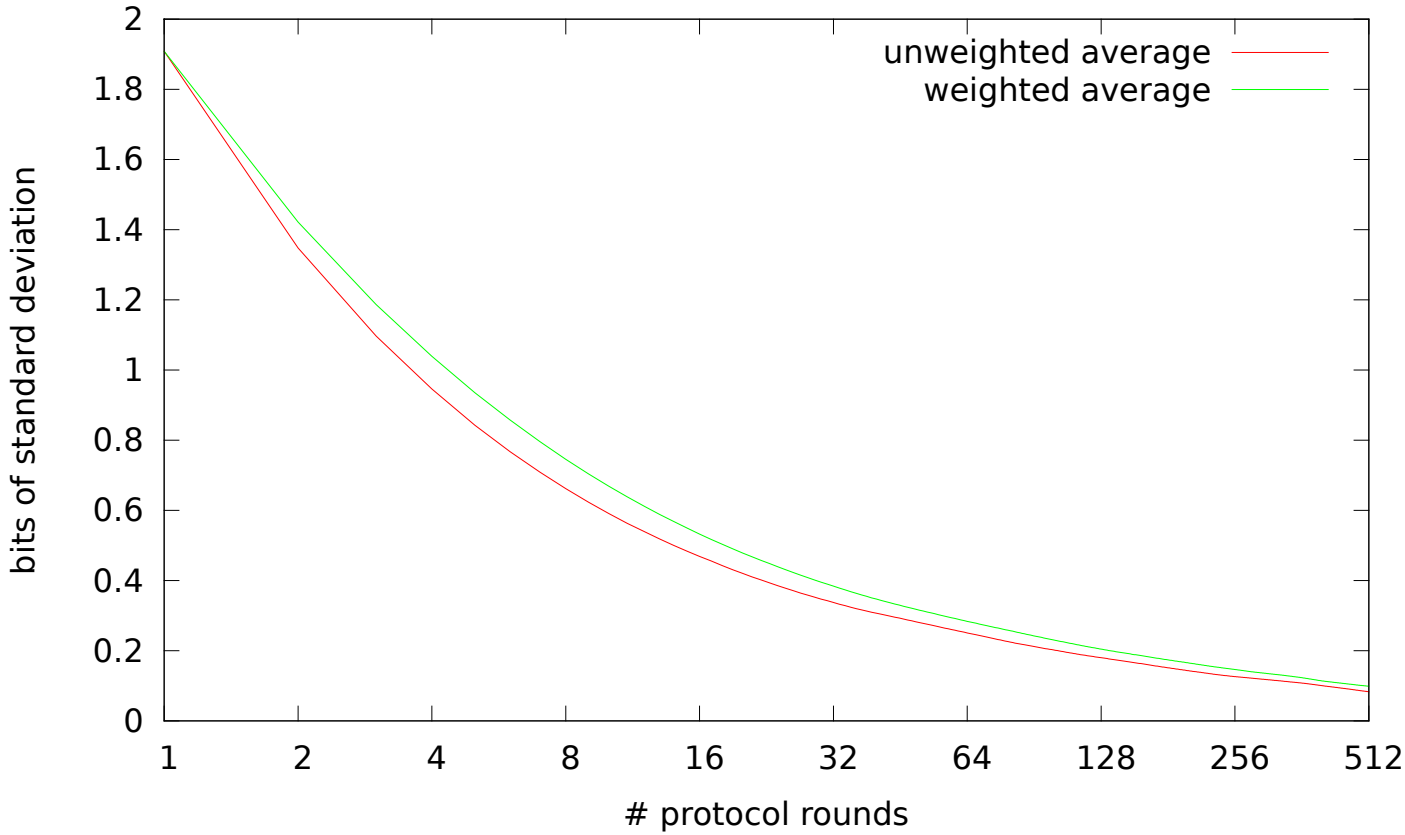
Attacker Model

- Freely participate
- Multiple identities
- May alter, drop, send/receive data
- Same resources as “normal” peers

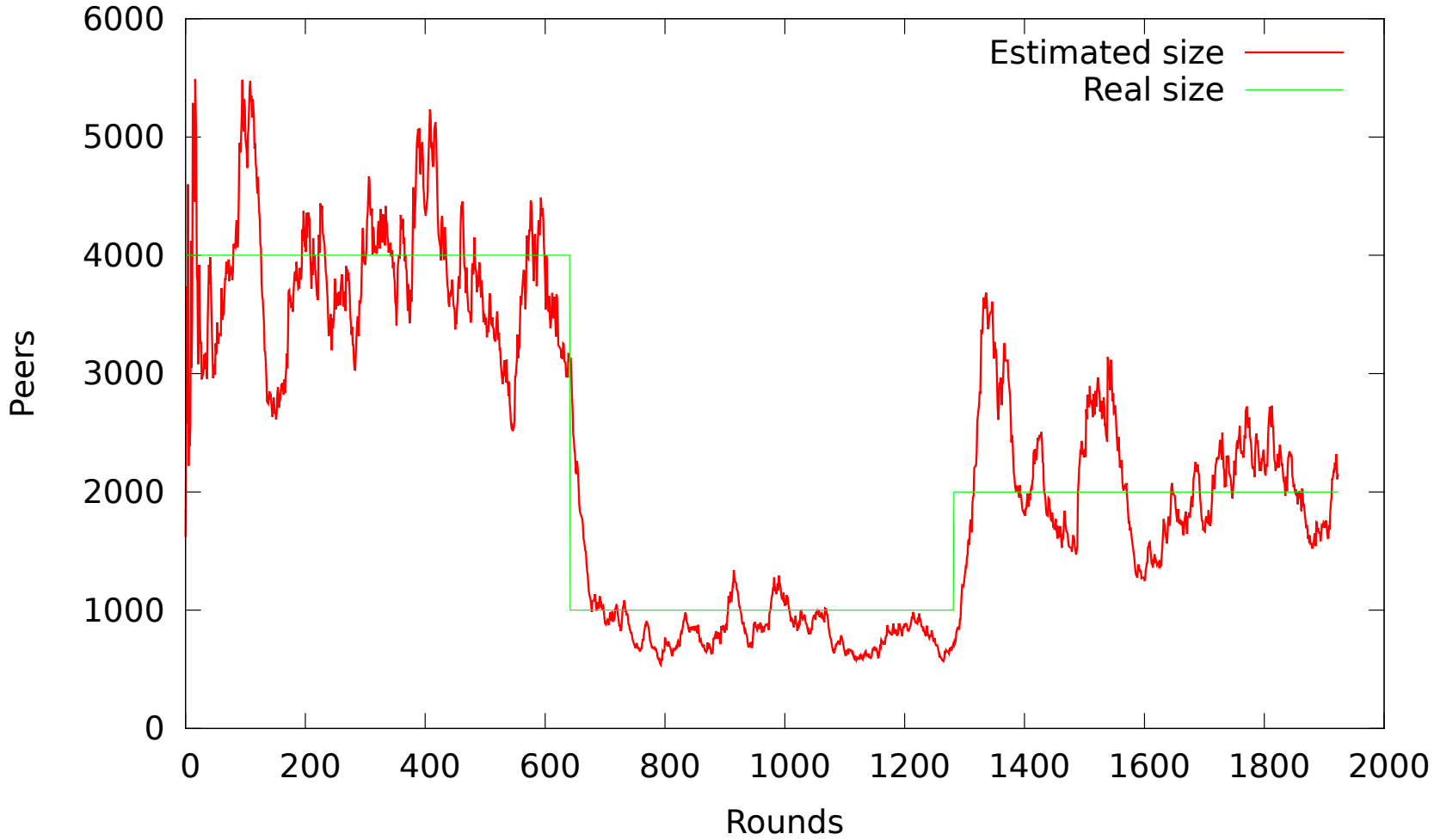
Processing results

- Final agreed value fluctuates around the actual size
- Average and std dev over last i protocol rounds is provided

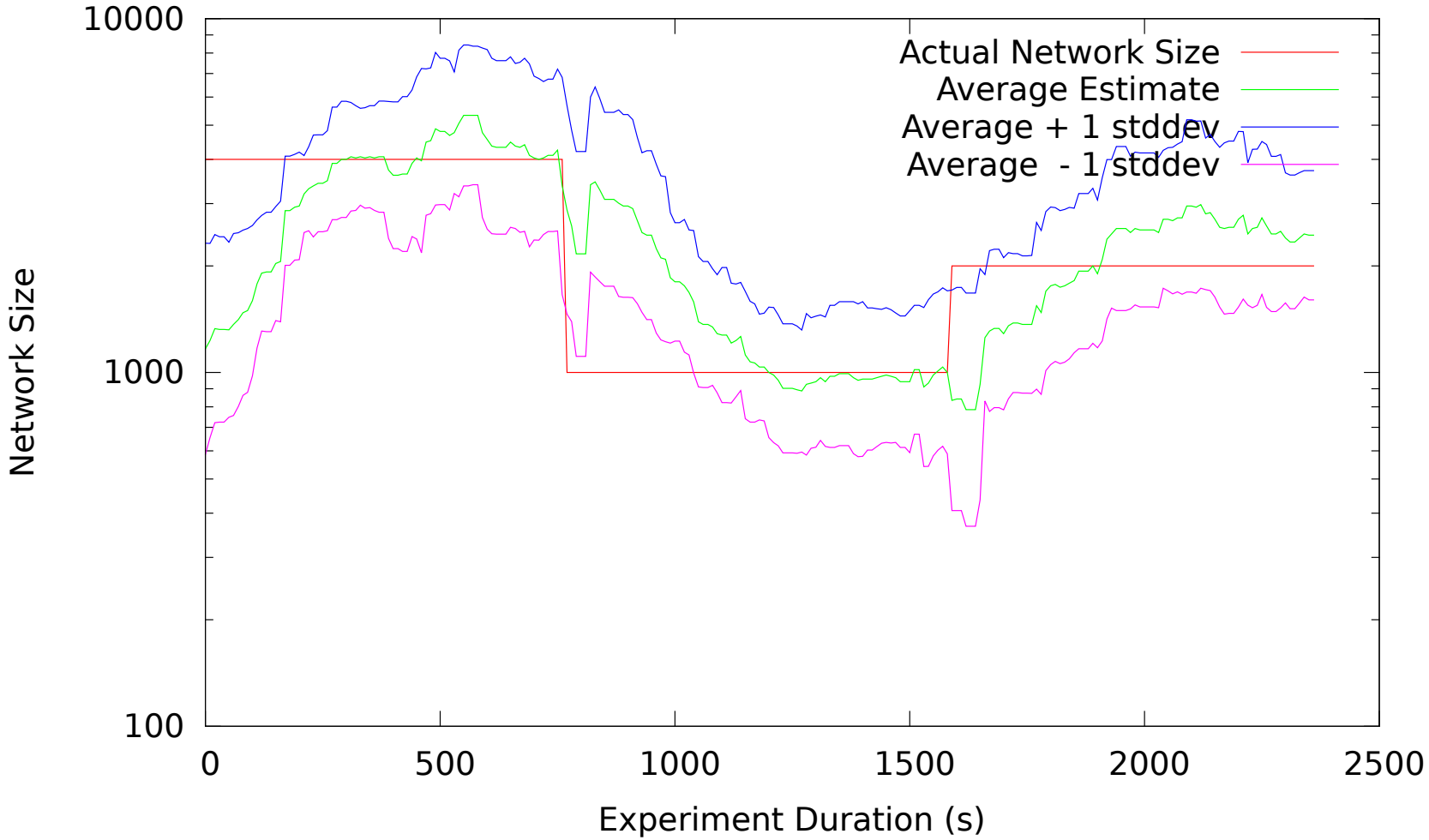
Precision vs. Rounds of Measurement



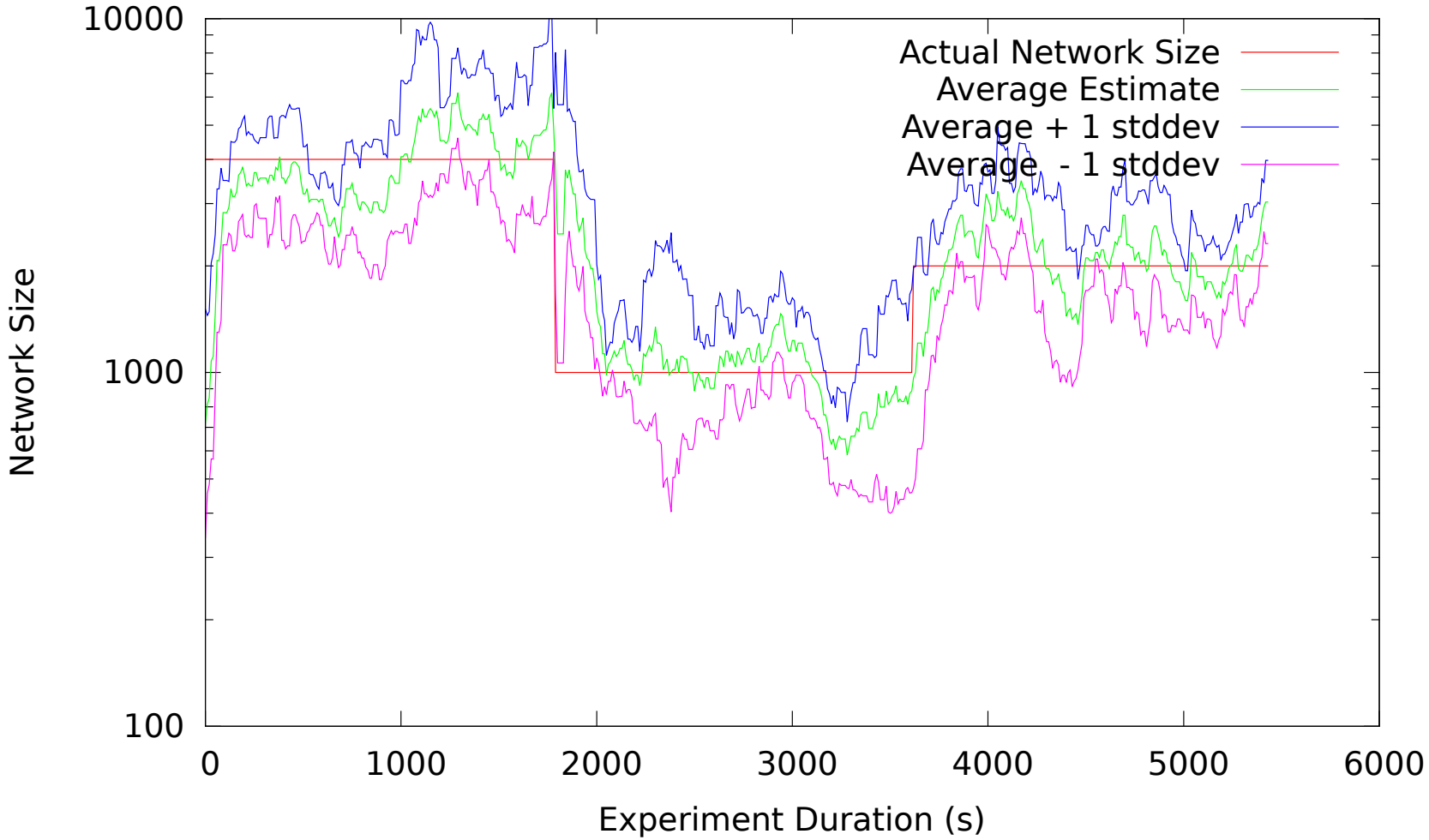
Network Size Estimate (64 rounds) under Churn



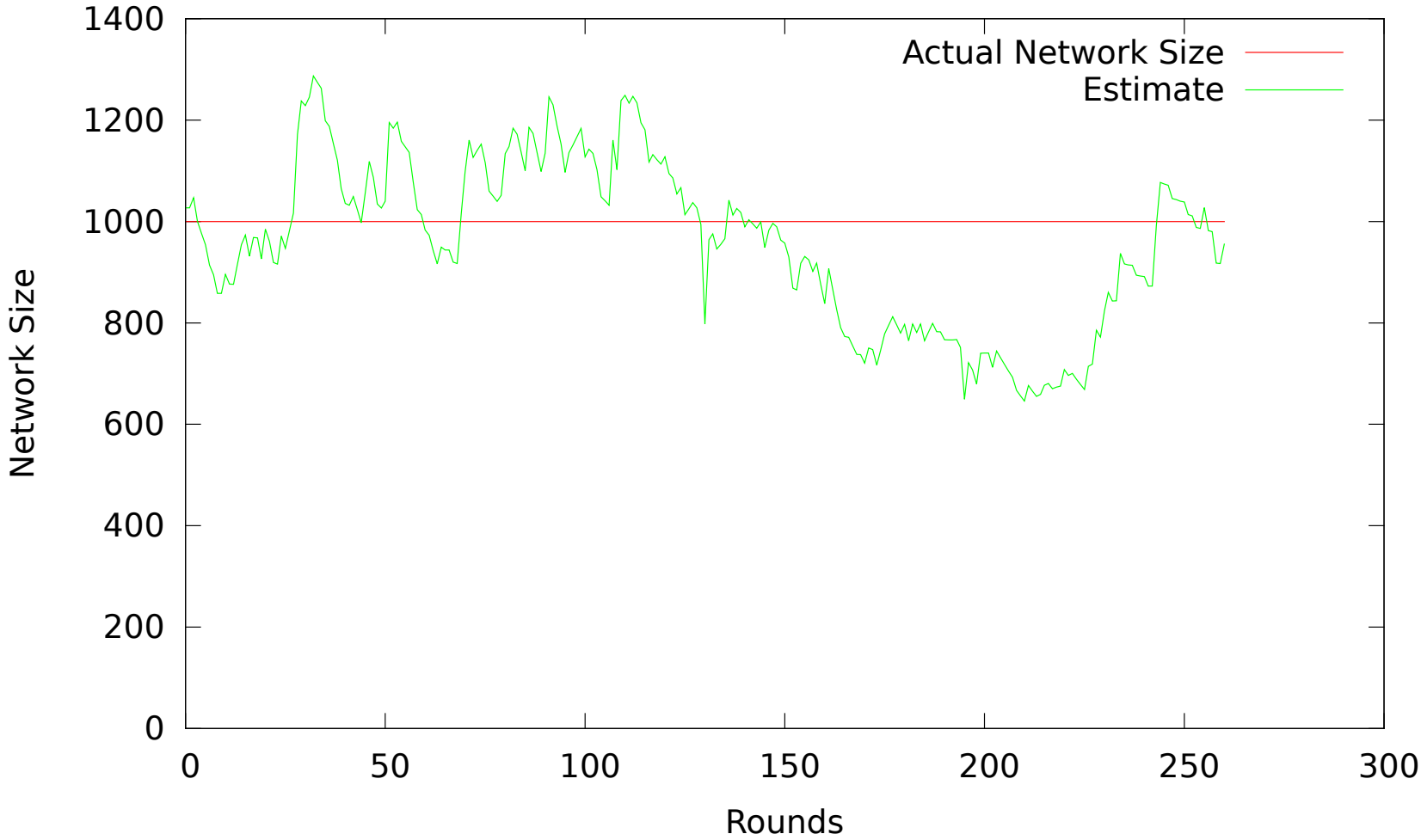
Small-World Topology



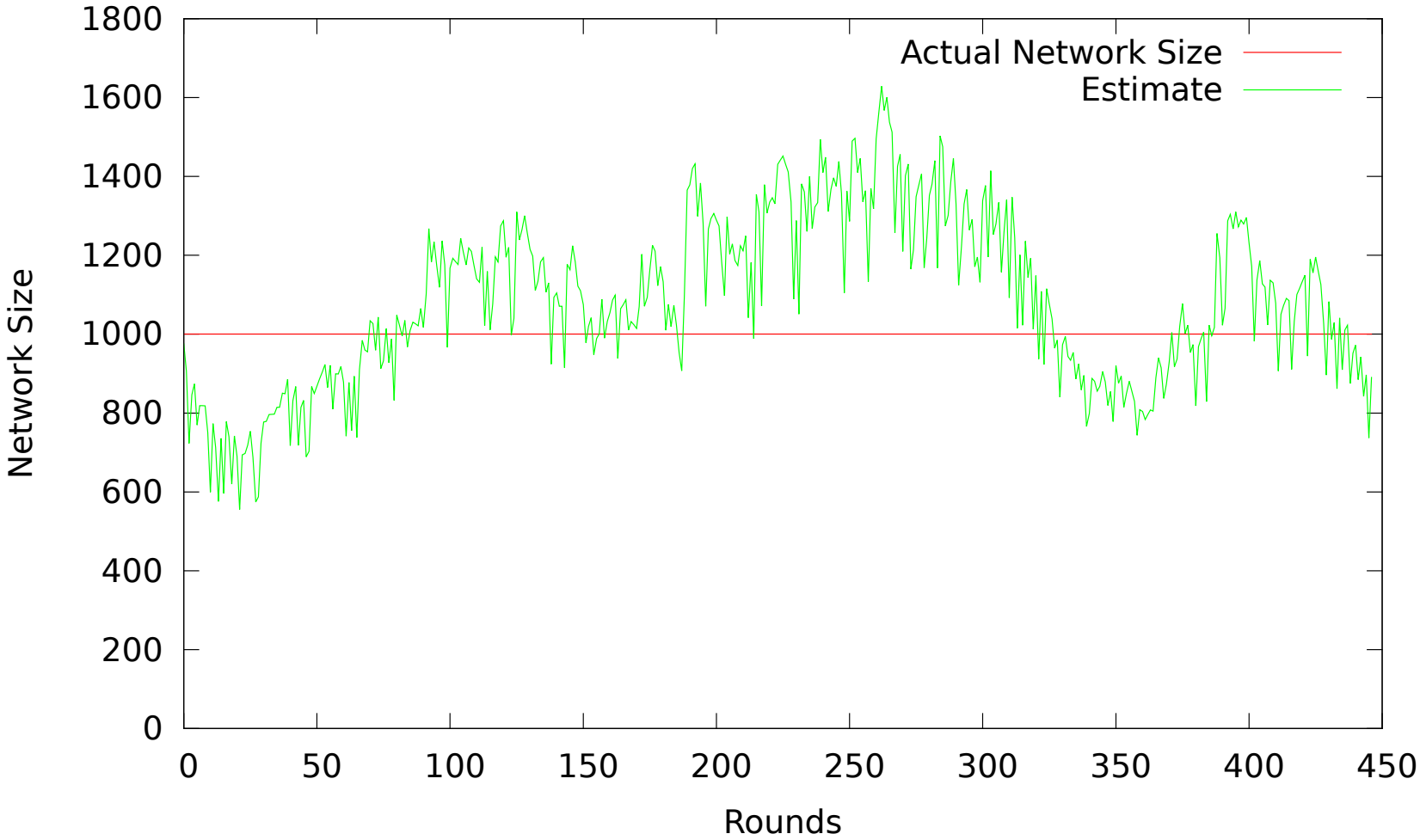
Random Graph Topology



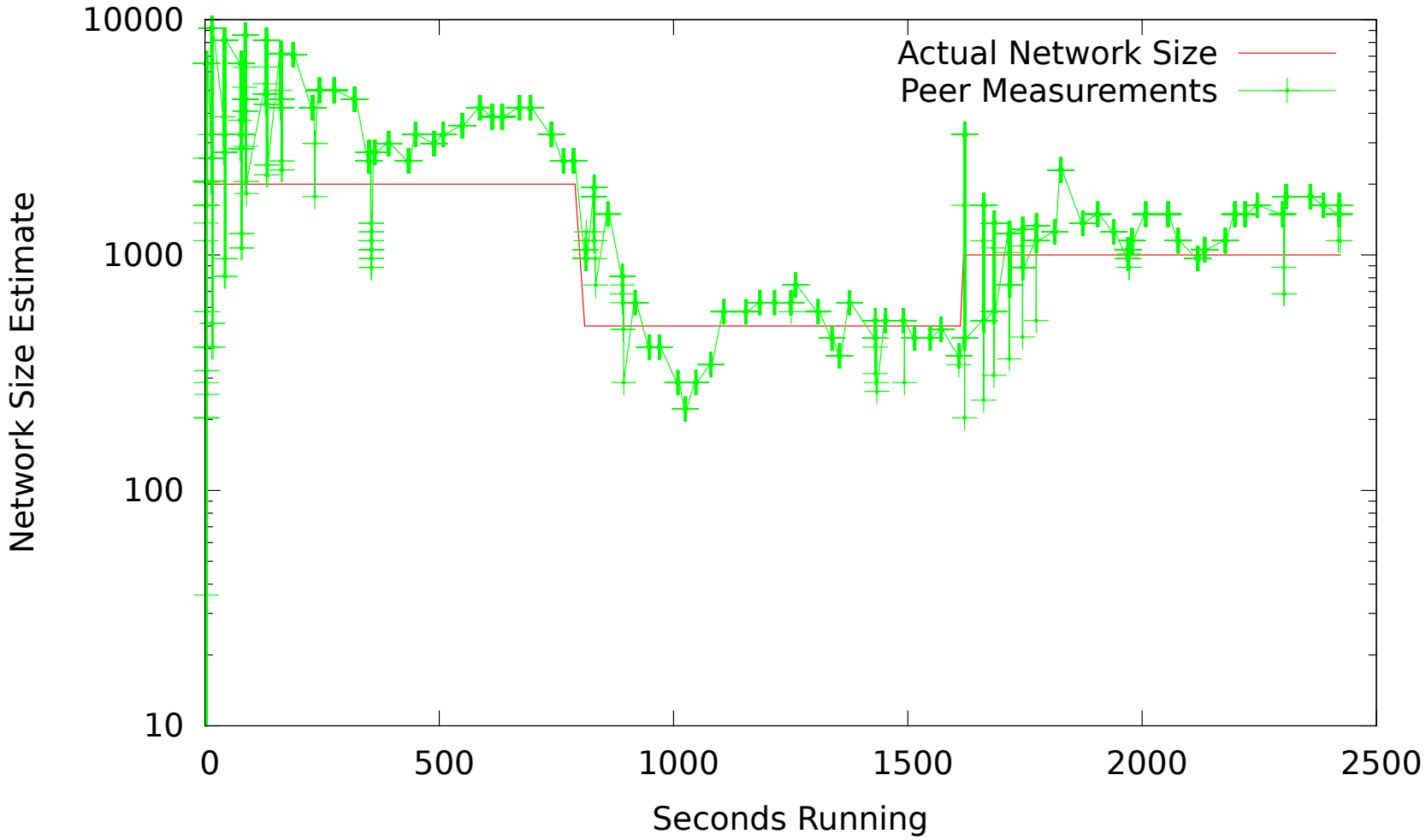
Without Clock Skew



With Clock Skew



Agreement between peers



Compare

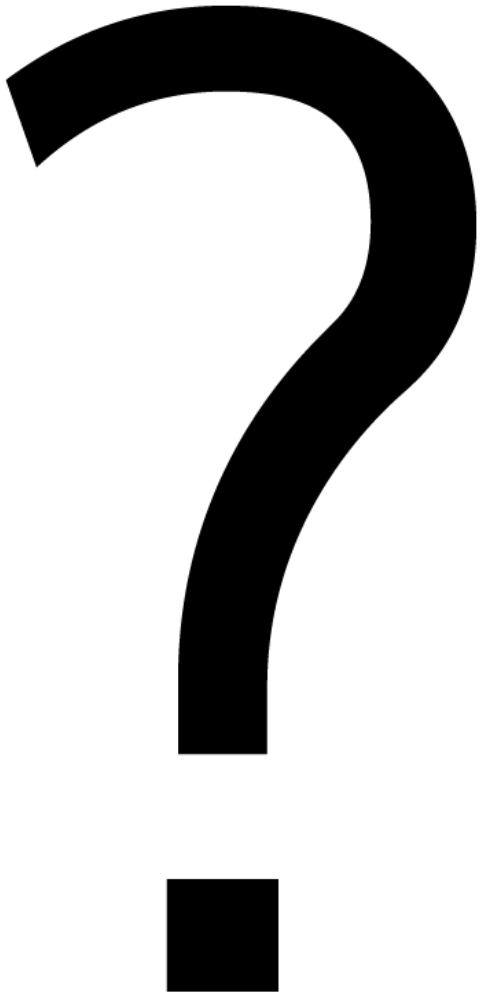
	S. & Coll.	Gossip	H. Sampling	This work
MEM	$O(\sqrt{N})$	$O(1)$	$O(\|N\|)$	$O(1)$
CPU	$O(\sqrt{N})$	$O(\|N\|)$	$O(\|E\|)$	$O(\|E\|/\|N\|)$
NET	$O(\|N\|\sqrt{N})$	$O(\|N\|^2)$	$O(\|N\| \cdot \|E\|)$	$O(\|E\|)$
SEC	DoS, BE	DoS, BE	DoS, BE	Pr.-of-Work
IMP	Simulation	Simulation	Simulation	Yes

(BE = Bad Estimates)

Conclusion

- Mathematical foundation applicable broadly for group size estimates
- Secure & Efficient Network Size Estimation Protocol
- Arbitrary Topologies, Clock Skew harmless, DoS resistant
- Simple to implement, free software implementation in GNUnet
- Trade-off between precision (Gossipico) and security (GNUnet)

Questions?



References

Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu.

Gossip-based aggregation in large dynamic networks.

ACM Trans. Comput. Syst., 23:219–252, August 2005.

Dionysios Kostoulas, Dimitrios Psaltoulis, Indranil Gupta, Ken Birman, and Al Demers.

Decentralized schemes for size estimation in large and dynamic groups.

In *Proceedings of the Fourth IEEE International Symposium on Network Computing and Applications*, pages 41–48, Washington, DC, USA, 2005. IEEE Computer Society.

Laurent Massoulié, Erwan Le Merrer, Anne-Marie Kermarrec, and Ayalvadi Ganesh.

Peer counting and sampling in overlay networks: random walk methods.

In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, PODC '06, pages 123–132, New York, NY, USA, 2006. ACM.

Tallat Shafaat, Ali Ghodsi, and Seif Haridi.

A practical approach to network size estimation for structured overlays.

In Karin Hummel and James Sterbenz, editors, *Self-Organizing Systems*, volume 5343 of *Lecture Notes in Computer Science*, pages 71–83. Springer Berlin / Heidelberg, 2008.

Ruud van de Bovenkamp, Fernando Kuipers, and Piet Van Mieghem.

Gossip-based counting in dynamic networks.

In *IFIP International Conferences on Networking (Networking 2012)*, pages 404–419, Prague, CZ, 05/2012 2012. Springer Verlag, Springer Verlag.