Chair of Connected Mobility
School of Computation, Information and Technology
Technische Universität München

Towards Multicast for Consensus in (Large-Scale) Distributed Systems

David Guzman, Dirk Trossen, Jörg Ott

13 March 2025

TUM Academic Salon

Yes, we can… …but should we?

… run arbitrary applications across the Internet
… using ill-suited protocols that cause lots of overhead

because the Internet is well-engineered and overprovisioned.

“With sufficient thrust, pigs fly just fine.” [RFC 1925]

- Some evidence
  - SOAP and XML-based RPC over HTTP “transport”
  - HTTP-based media streaming (even for live events)
  - Distributed consensus systems

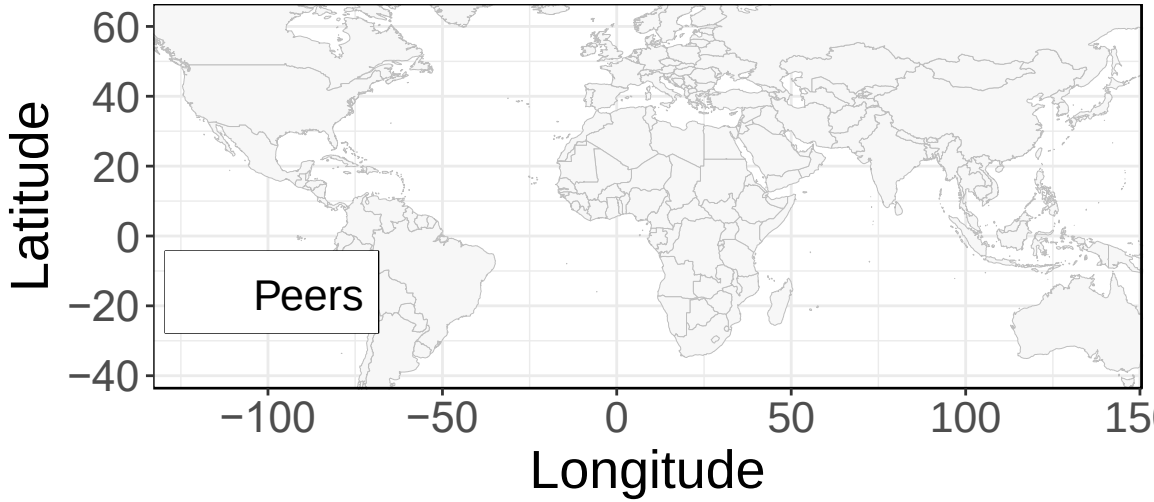
- Case study: Ethereum and learnings for networking

Distributed consensus

- Take a system of N nodes (subject to churn)
- Each node runs a replica of a state machine
  - Independently compute deterministic state transitions as f (state, input)
- Nodes may invoke state transitions S➜S' = f (S, input)
  - Updated state S' broadcast to all nodes
  - S' becomes the new state if accepted by the majority of nodes (> N/2)
- State transitions may be invoked in parallel
  - Above majority rule resolves the conflict to obtain consensus on state
  - Signed chains of blocks to prove that a transaction was based on consensus state
- Random node selection supports fault-tolerant operation
  - Protects against faulty and malicious nodes

Ethereum: a sample consensus system

- Internet scale
- 72K nodes [2022]
  - Found experimentally using one vantage point



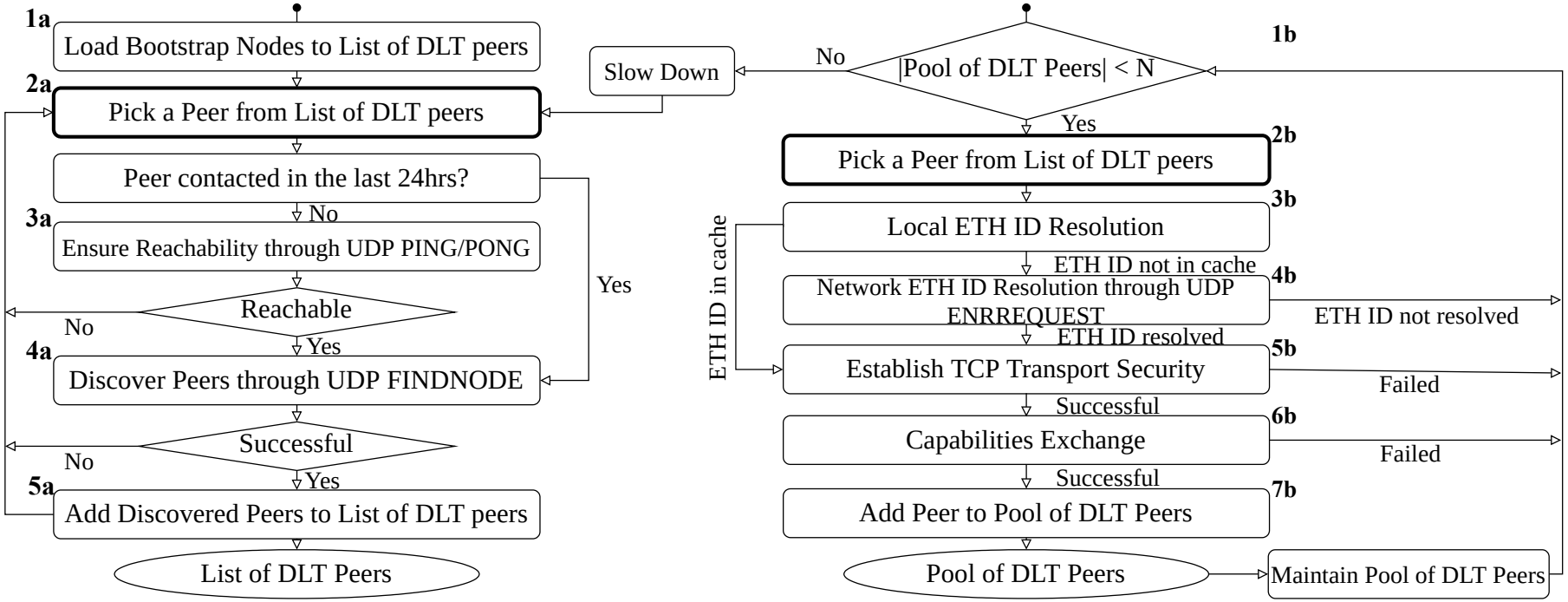
- Random topology among nodes
  - 50 peers per node
    - Outgoing or incoming connection per peer
  - continuously updated
    - Dropping existing peers, establishing connections to new ones

- Operation in different phases
  1. Topology maintenance (“control plane”)
  2. Transaction diffusion (“data plane”)

1. Ethereum topology maintenance

Peer discovery

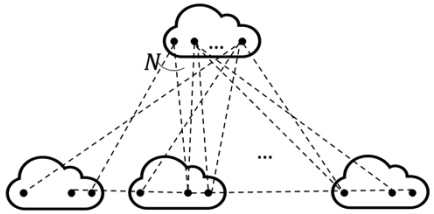
Peer establishment



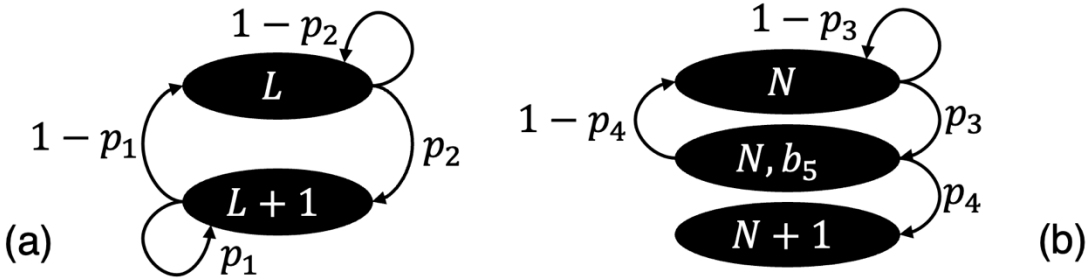
Independent processes

1. Ethereum topology maintenance

- Communication cost upon success: some 3 KB per peer



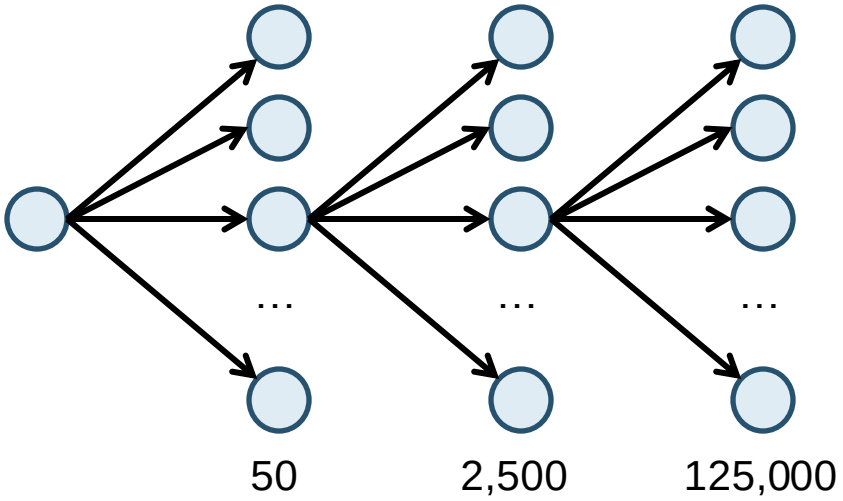
Subproc.	Var.	B	Subprocess	Var.	B
Reach.	a_2	347	TCP	b_3	120
Lookup	a_4	$a_2 + 1284$	Auth. (min)	b_5	120+409
			Hand. (min)	b_8	529+402
			Cap. (min)	b_{10}	931+150



1. Ethereum topology maintenance

- But not all connections are successful
- 1.4% non-reachable peers
  - Failed PING/PONG or ENRREQUEST
  - Routing, NATs, …?
- 85.6% don’t complete signaling
  - ETH-ID resolution, TCP handshake, TLS establishment, …
- 0.6% connect but fail at data exchanges
  - Information not found, incomplete transfers, …
- Only some 13% complete all stages and make it to the pool
- Measurements showed a mean of ~20min for a pool of 50

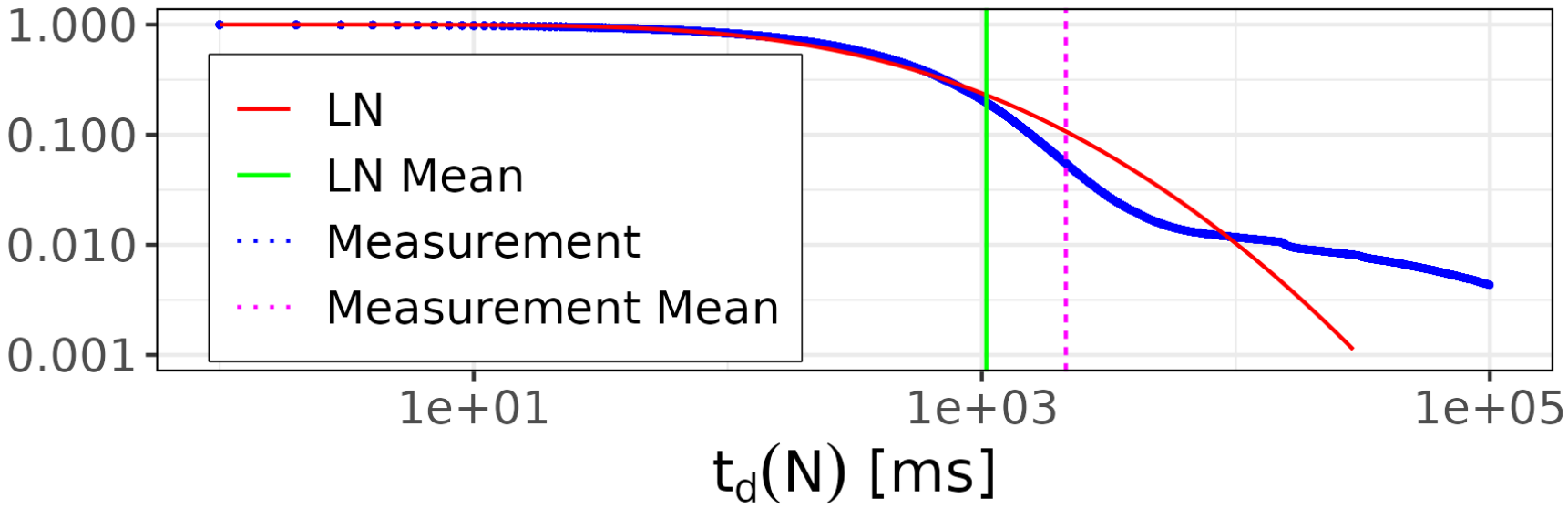
2. Ethereum operation: iterative diffusion



- Nodes spread state updates to their peers
  - which recursively continue spreading
- Incurs independent peer pool processes at each node
  - arbitrary distances with naturally varying latencies (different RTTs)
  - random processes lead to duplicates (increasing p at each stage)

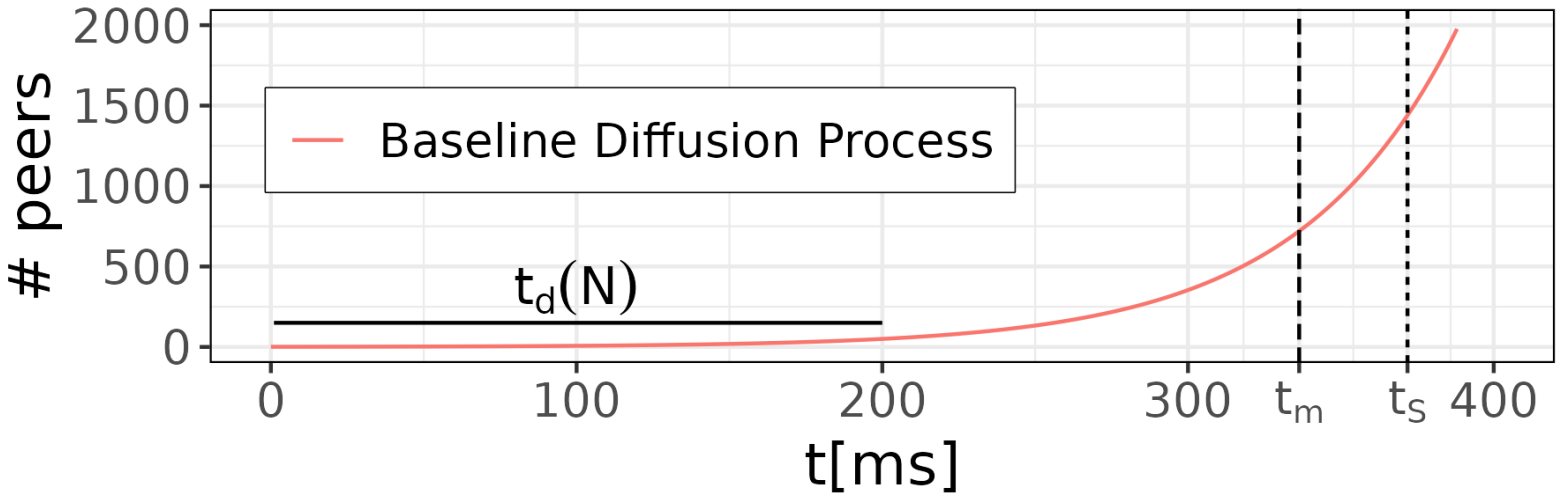
2. Ethereum operation: iterative diffusion

- Long tail in the distribution of diffusion to 50 peers in one iteration



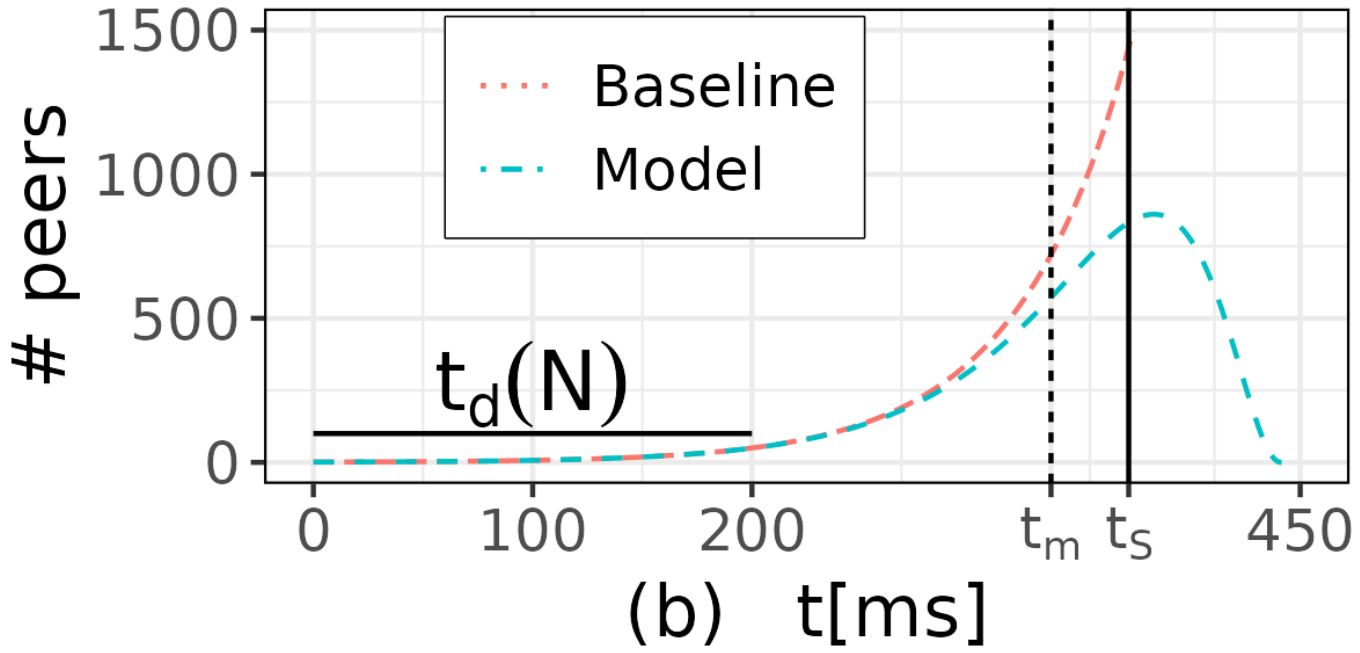
2. Ethereum operation: iterative diffusion

- Takes time: sample for 72k peers
- Fresh peers per step without duplicates



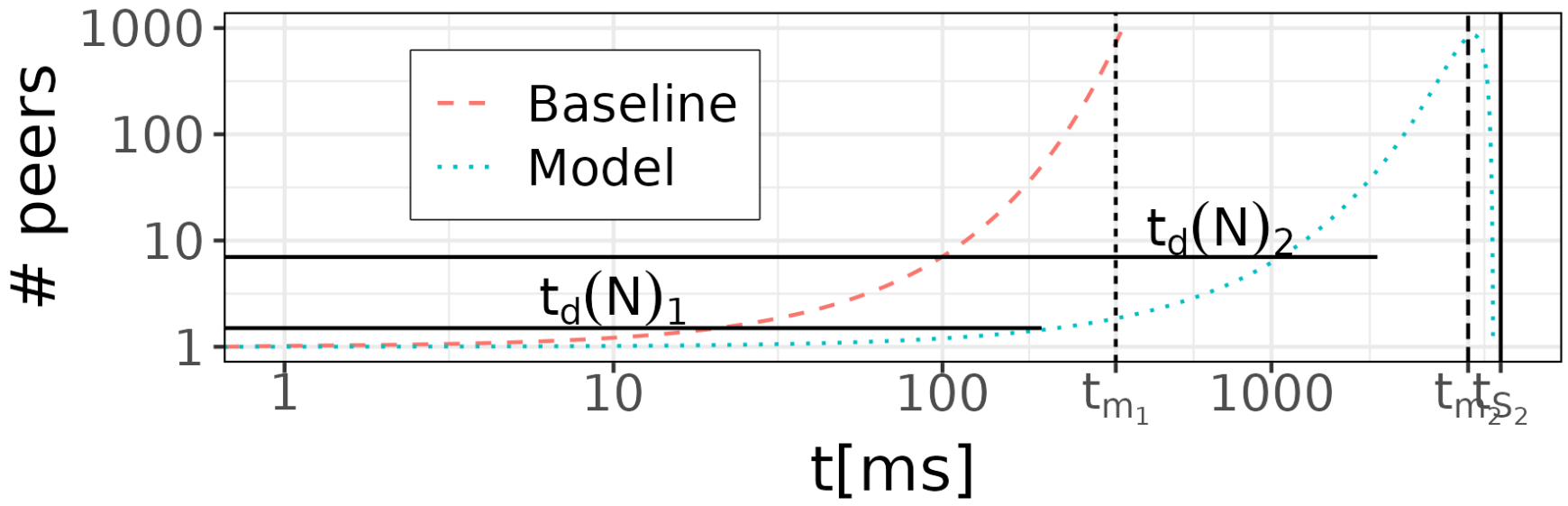
2. Ethereum operation: iterative diffusion

- Takes time: sample for 72k peers
- Duplicates in peer pools lead to diminishing returns



2. Ethereum operation: iterative diffusion

- Takes time: sample for 72k peers
- In comparison: avoiding duplicates can improve ~12x

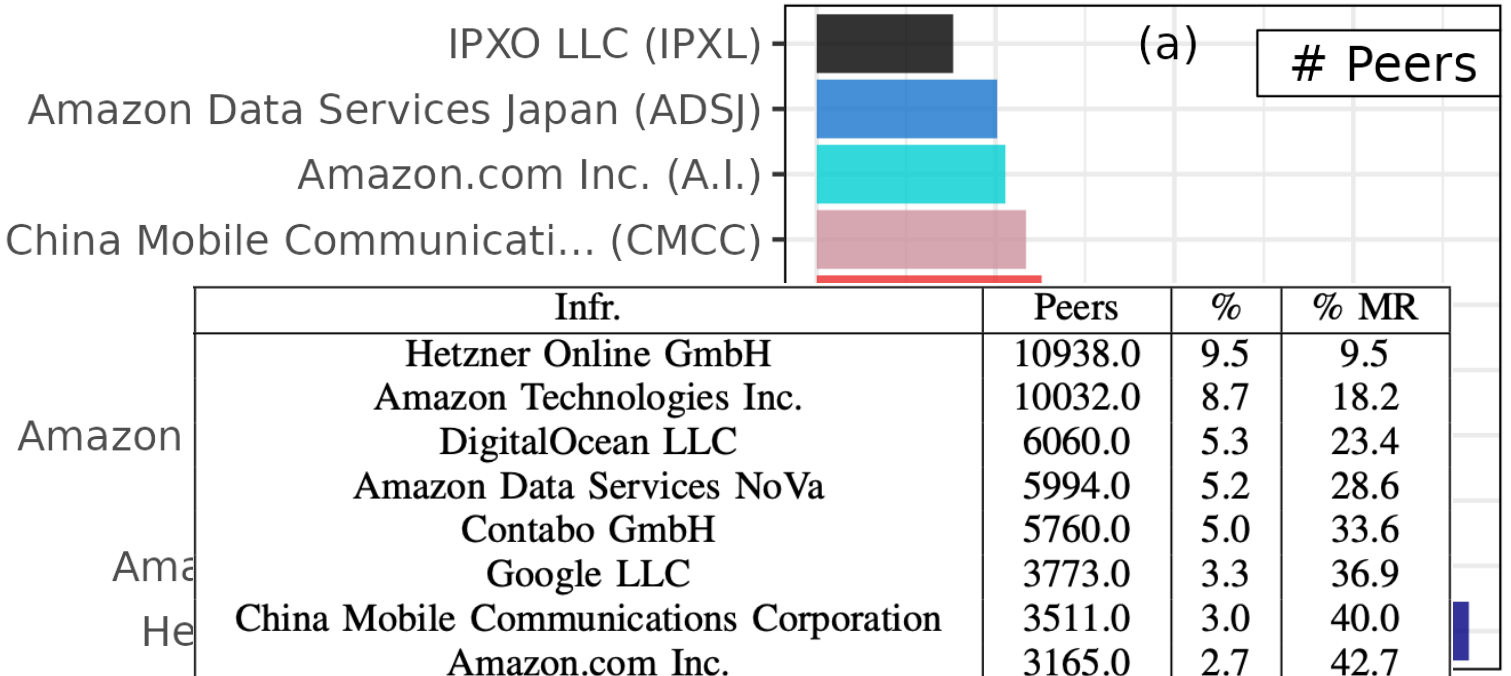


Ethereum: a decentralized system?

- Distributed consensus systems should avoid central control
- Designed to support trust in untrusted environments
- Decentralized operation across many nodes yields independence
- Does it?
- Well, DCs are too convenient…

Ethereum: a decentralized system?

- Majority of peers run in the infrastructure of 10 providers



Infr.	Peers	%	% MR
Hetzner Online GmbH	10938.0	9.5	9.5
Amazon Technologies Inc.	10032.0	8.7	18.2
DigitalOcean LLC	6060.0	5.3	23.4
Amazon Data Services NoVa	5994.0	5.2	28.6
Contabo GmbH	5760.0	5.0	33.6
Google LLC	3773.0	3.3	36.9
China Mobile Communications Corporation	3511.0	3.0	40.0
Amazon.com Inc.	3165.0	2.7	42.7
Amazon Data Services Japan	3027.0	2.6	45.3
IPXO LLC	2289.0	2.0	47.3
Amazon Data Services Ireland Limited	2070.0	1.8	49.1
Amazon Data Services Singapore	1640.0	1.4	50.5

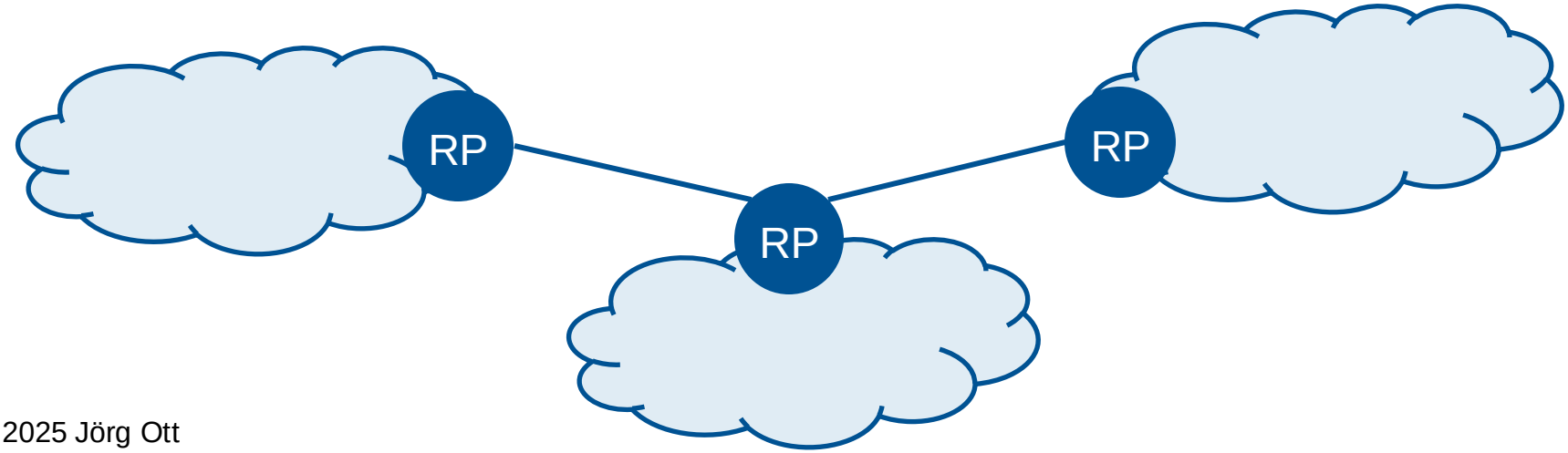
Ethereum: a decentralized system?

- Can we turn this bug into a feature?

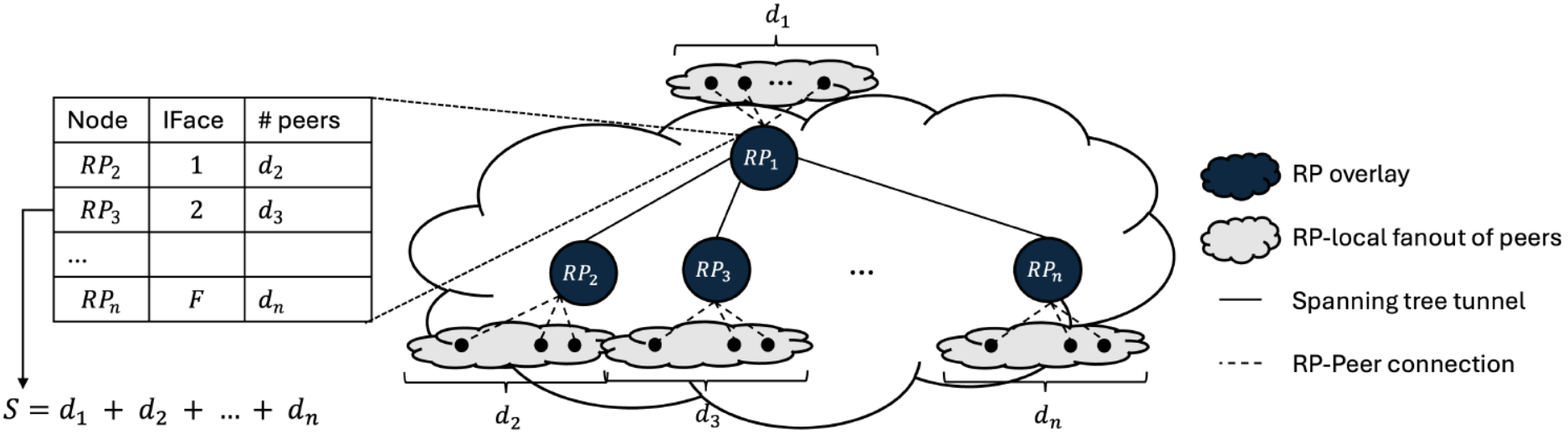
- Especially to support efficient diffusion
  - with low latency
  - with minimal network capacity utilization

Multicast to the rescue?

- Well, native multicast not turned on at scale
  - This is why people built overlay multicast in the first place!
- But would we need a general multicast?
  - One could build a unicast a spanning tree across ISP domains
  - And use source-specific multicast within ISPs
- Introducing replication points as part of the infrastructure

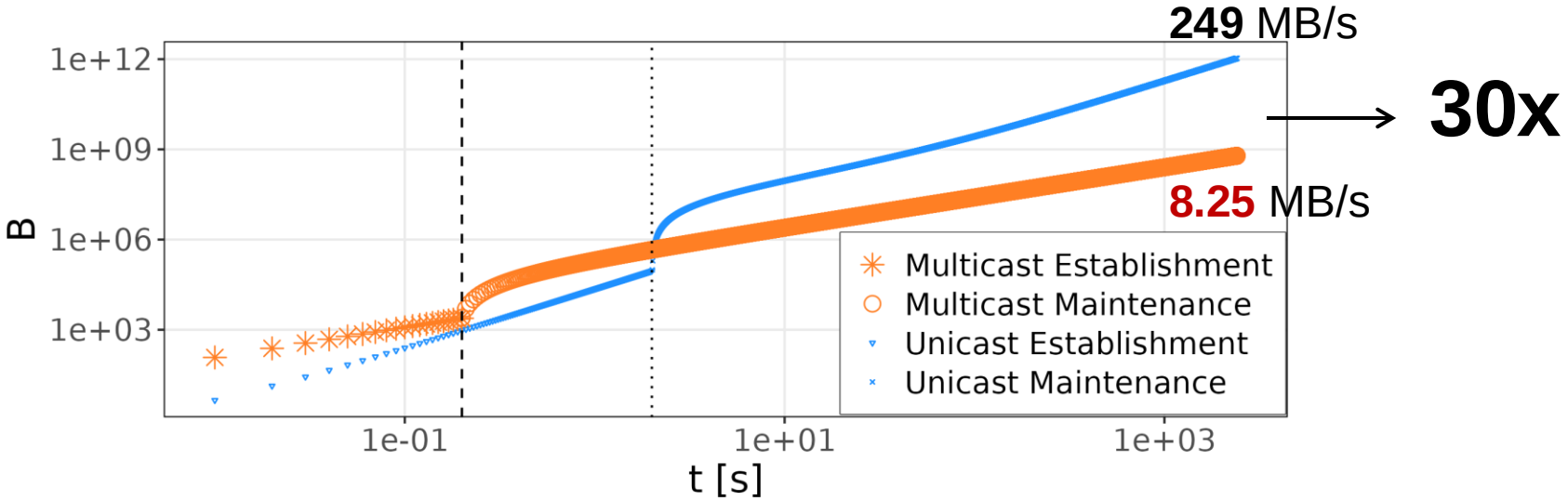


Multicast to the rescue?



- Nodes register to their RPs: think IGMP + PIM
- RPs maintain liveness and node count
- RPs form a backbone for diffusion
- Use efficient per-provider mechanisms for local diffusion
- Localized randomization and failure handling
- RP failover mechanisms inspired from routing

Potential performance gains



Data plane: model-based computations yielded 20MB data exchange for a 256 byte transaction.

And high-speed networks?

- DC environments could benefit but at different time scales
- Could be in a better position
  - Controlled RP deployment
  - L2/L3 multicast
- Maybe not for permissionless p2p consensus protocols
  - Could have better options
- Consensus protocols such as Paxos shown to benefit from in-network support (cf. NetPaxos)

Concluding

- Building large-scale distributed systems is hard
- Building them well from a networking perspective is harder

- Dissecting and understanding protocol interactions
- Paired with operational system measurements
- Derived analytical traffic and performance models

- Let to an in-network assisted design
- I-D for the PIM WG @ IETF 122 next week

- Real numbers?
- Threat models?
- Who is to run RPs? How to find, assign, scale them?