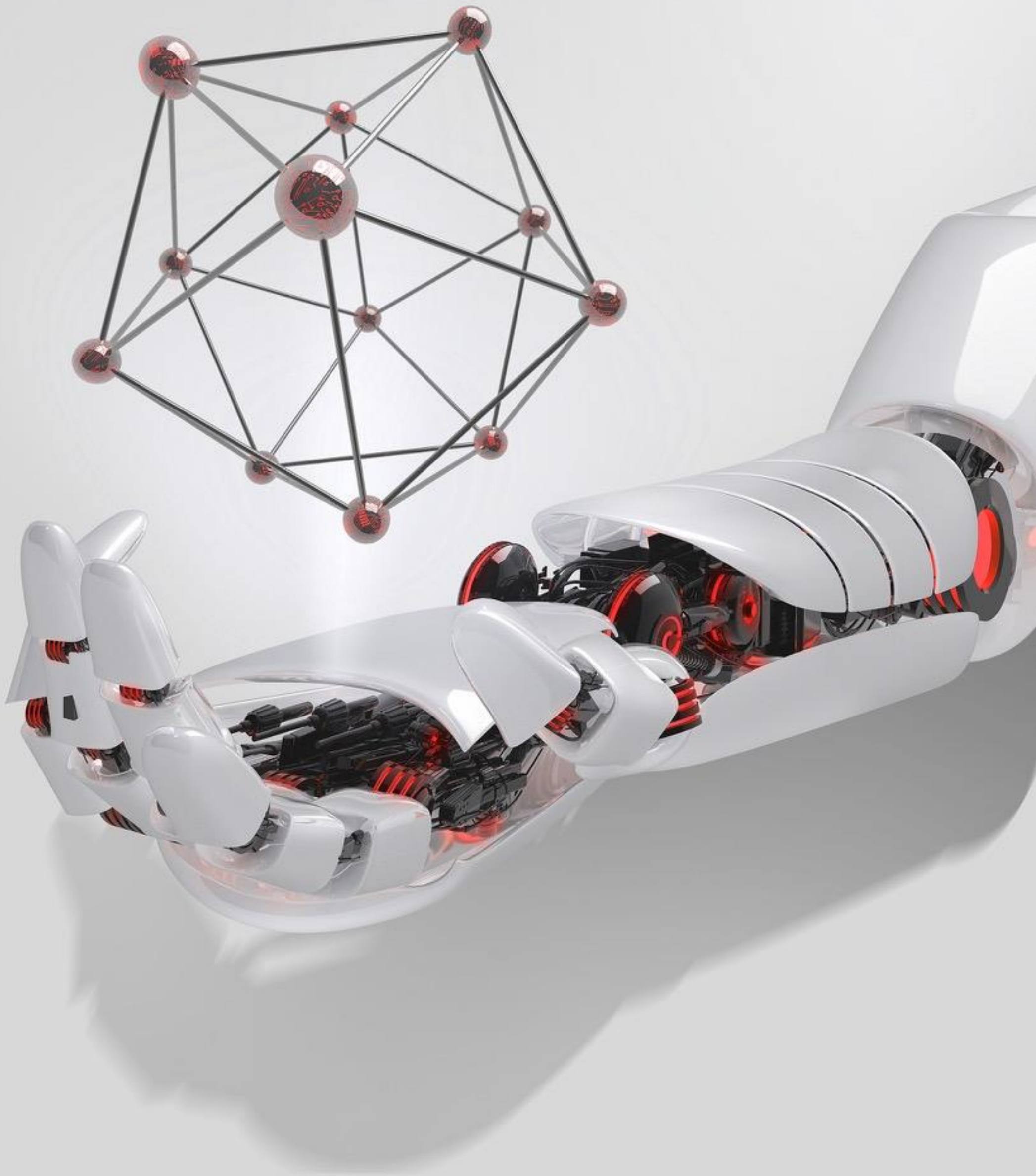
Decentralized Distributed Consensus at Expenses of Diffusion



David Guzman, Dirk Trossen, Jörg Ott

Munich Internet Research Retreat
Raitenhaslach

October 2024

TUM HUAWEI

Framing our Discussion Today

- **Distributed consensus systems (DCSes)** foresee participating peers to achieve consensus over a shared state in a continuous computation
  - **Majority rule** [VonNewman1956]
- We discuss a randomized consensus **approach**: why and how
- We present **empirical** insights
- We then discuss its implications for these **D**CSes: are they really **D**?
- We ask for further solutions to current approaches

The Consensus

- Distributed consensus arises in **replicated** state machines
  - State machines in a collection of **peers** compute **identical** copies of the **distributed** state and can **continue** operating even if some of the peers are untrusted (permissionless) or are down **[Ongaro2014-USENIX]**
- DCSes, e.g., DLTs:
  - Peers **replicate** information, i.e., transactions or blocks
  - The system picks a **random** replica (peer). **Decentralized**
  - The random peer **decides** on the distributed consensus.
  - This peer **replicates** (~ broadcasts) the consensus
- Information **replication** at the Internet scale is **key**
  - Current realization: randomized iterative diffusion

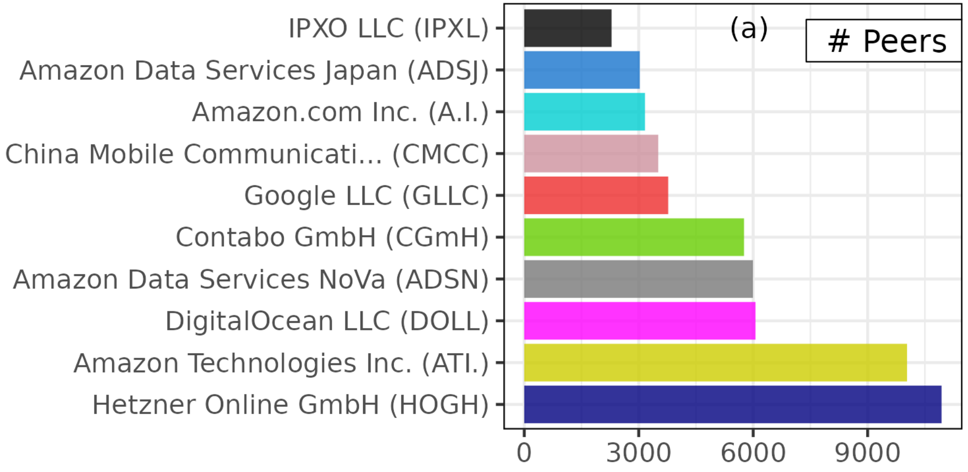
The Fault Tolerant Consensus

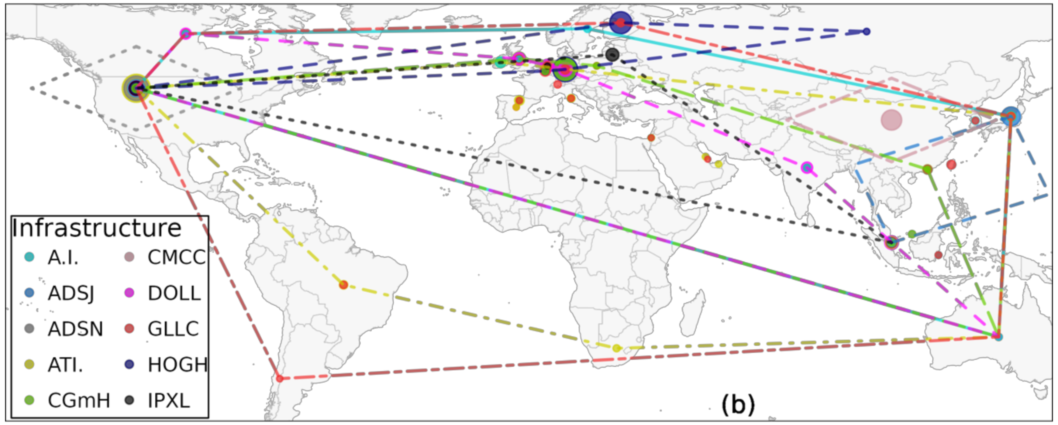
- Byzantine peers **[Castro1999]**
  - Consensus adversaries
  - Message adversaries
- The <u>logs</u> in replicas, i.e., ledgers, cope with **consensus adversaries**
  - The hashed-ordered chain of blocks avoids the double-spending **[Nakamoto2008]**
- Randomized iterative <u>diffusion</u> copes with oblivious **message adversaries**
  - Randomization and replenishing copes with churn **[Maymounkov2002]**
  - **~System resilience**

Measurement Insights

The majority of the system relies on few well-known cloud providers

- **Infrastructure <u>concentration</u> [Guzman2024-ICBC]**
  - **Low churn** (low frequency for peers going offline),
  - <u>High availability</u> and <u>reliability</u>
  - Internet-scale systems like IPFS, Bitcoin, TON, and XRP Ledger
  - (Monetary) incentives motivate users to deploy in highly reliable and available infrastructures at the expense of **system resilience**





Measurement Insights

Internet relations from peer-to-peer to customer-provider

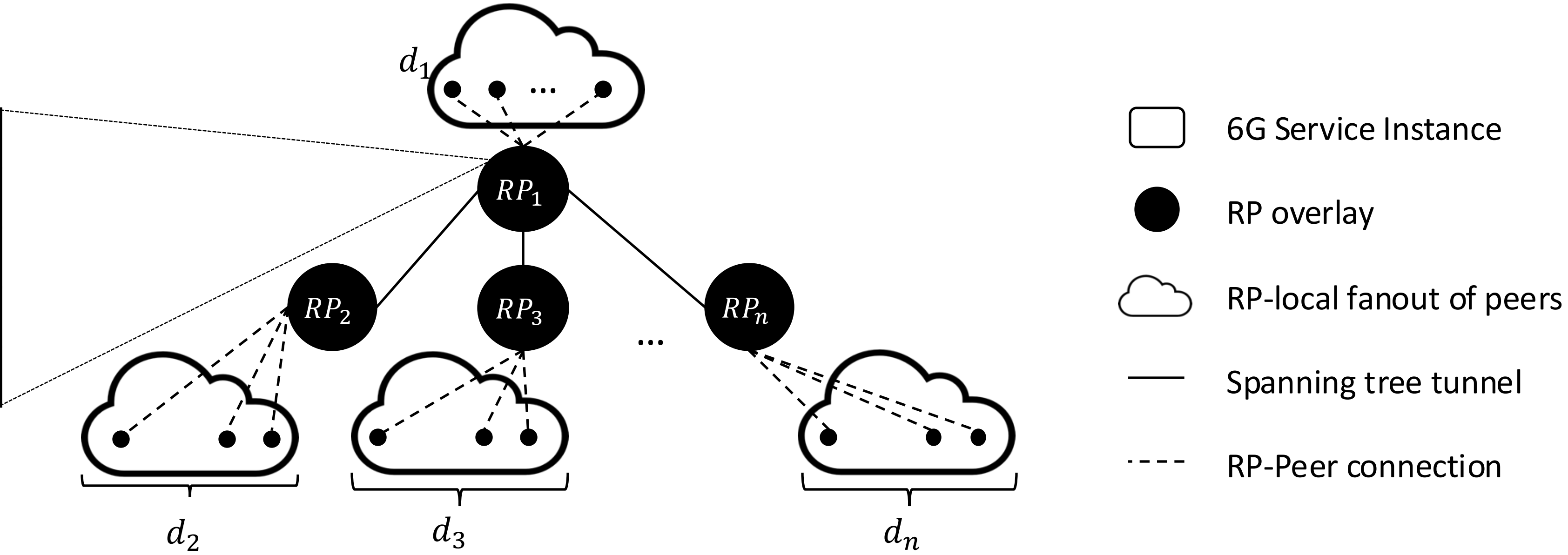
- The **delay** for opening a **communication relation follows a tail distribution** scale higher than the Internet delay, **~10x**, even though these distributed systems run in highly reliable infrastructures with few hops **[Guzman2022-ICBC]**
- Randomized iterative diffusion **cannot evaluate consensus finality**
  - Defining contention times to evaluate the consensus
  - Contention times are part of the proofs (e.g., PoX)
- **Expensive** communication costs for data and control planes due to the distributed nature of the consensus
  - DP: Distributed state, while randomly being flooded, gets **duplicated, 22 MB per 256 B transaction; on a day, 41 TB**
  - CP: **3 GB is** required only to discover and establish relations; this grows to a maintenance regime at a rate of **249 MB/s** [Guzman2024-DIN CoNEXT]

Is there Something better?
A multicasted approach

- Multicast-based diffusion
  - Places replication points (RPs) in infrastructure clusters
  - Replicates distributed state
  - RPs are not part of the consensus; consensus is still (randomized) **decentralized and fault tolerant** (oblivious message adversaries)

Node	IFace	# peers
RP_2	1	d_2
RP_3	2	d_3
…		
RP_n	F	d_n

$S = d_1 + d_2 + \ldots + d_n$



Is there Something better?

Multicast Insights

- There is no **delay** in opening a **communication relation**
  - **The diffusion can be enhanced by 5x** [Guzman2024-IFIP]
- Multicast **can evaluate consensus finality**
  - Informed contention times without expensive proofs (e.g., PoX)
- **Reduced** communication costs for data and control planes
  - DP: Distributed state, replicated with negligible extra cost
  - CP: Only **4 MB is** needed **(instead of 3 GB) to** establish relations with a maintenance regime 30x better than unicast [Guzman2024-DIN CoNEXT]