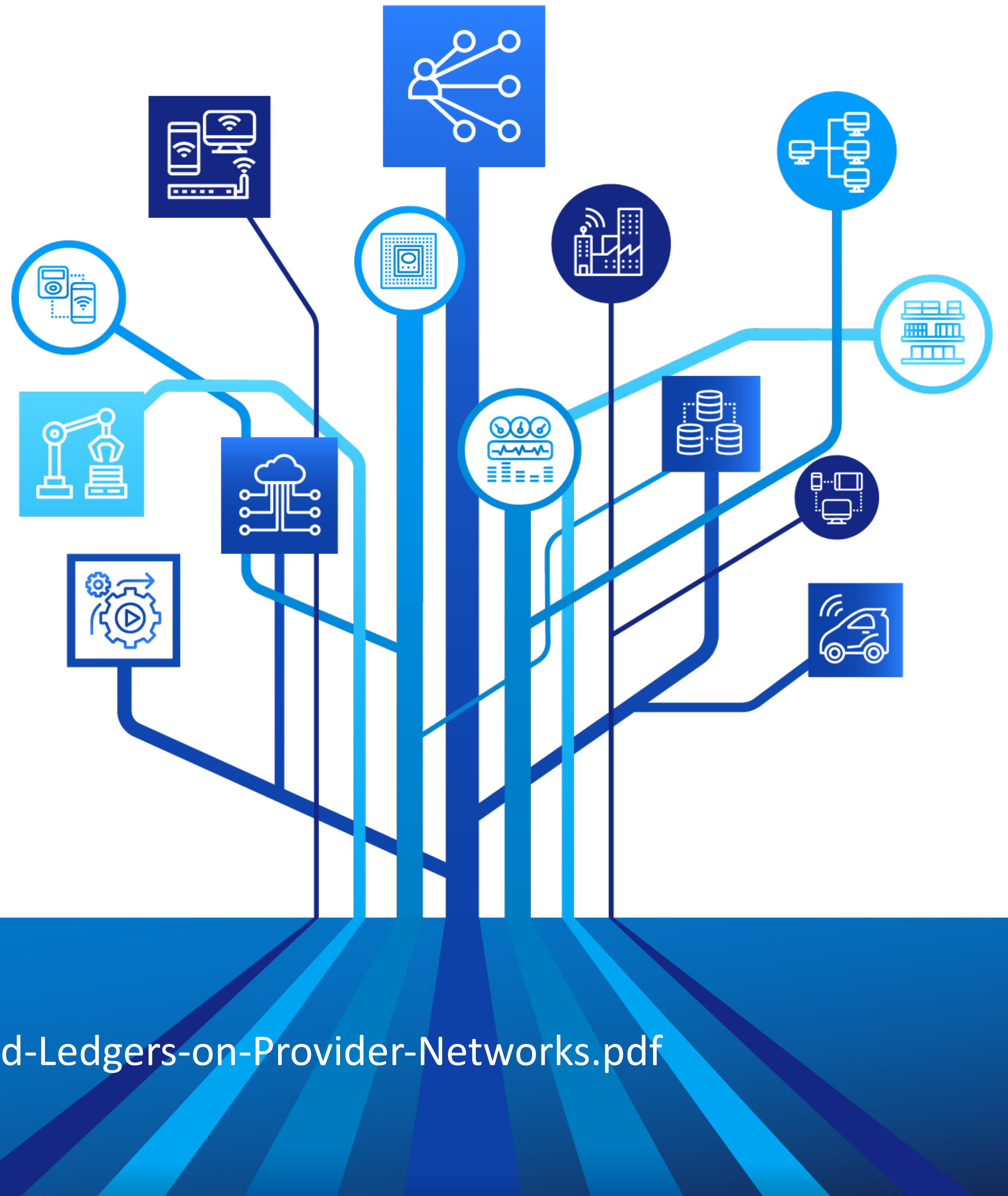
Industry IoT Consortium

Impact of DLTs on Provider Networks

Dr. Dirk Trossen, Chief Network Architecture Research, Huawei
Dr. Xinxin Fan, Head of Cryptography, IoTeX
Mike McBride, Sr Director of Innovation and Strategy, Futurewei



https://www.iiconsortium.org/pdf/2022-01-10-Impact-of-Distributed-Ledgers-on-Provider-Networks.pdf

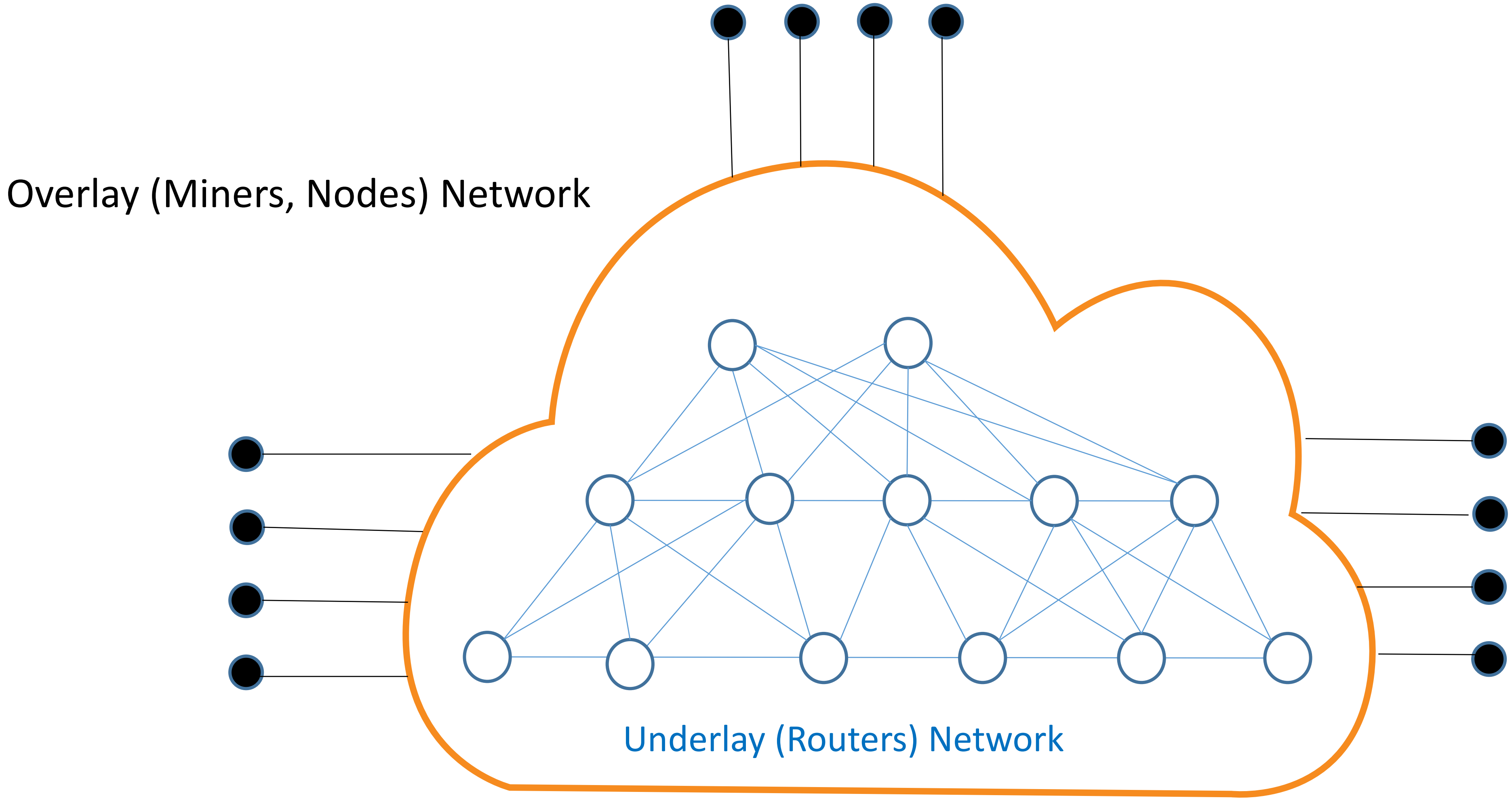
Agenda

- Opportunities
- System Overview
- Impact on the Network
- Relevance to Standardization

The DLT Network

- Crypto currencies and DLTs don’t typically care about the underlying TCP/IP network.
- They have a P2P overlay network (TCP, UDP based) and that is their focus.
- They have done a good job securing their application and don’t worry about the network.

The Network

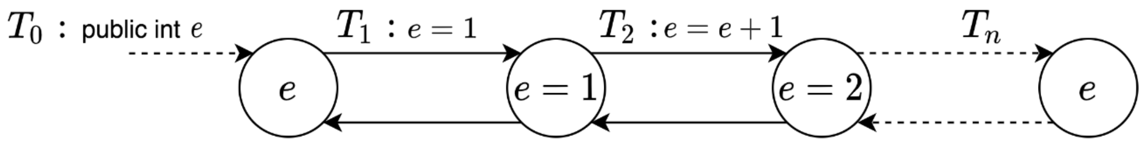


Opportunities

- Trust packet capture data
- Network mgmt moves to a decentralized, smart contract-based system
- Web 3.0, decentralized Internet
- BGP/RPKI blockchain
- Overlays such as LISP

Main DLT Concepts

- **Transaction:** A cryptographically-signed instruction or set of instructions to modify the state machine.



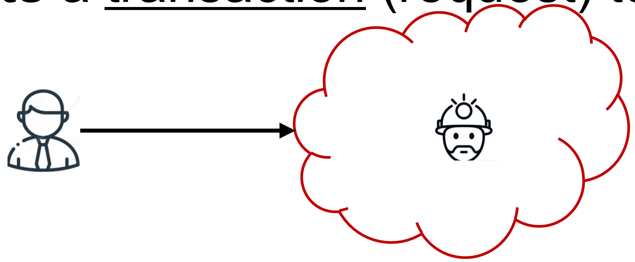
- **Ledger:** A set of valid transactions
- **Block**: a hashed ledger
- A set of concatenated blocks is a **blockchain**
- To find a hash with a specific pattern (leading 0s) is called **PoW (proof-of-work) a.k.a. mining**

DLT Layering Architecture

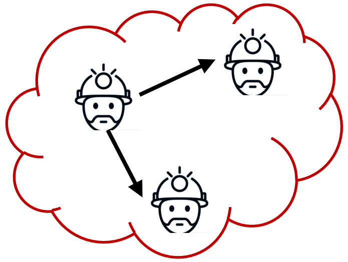
Application Layer	User Interface	DLT Wallet	DLT Explorer	DLT Analytics	Decentralized Finance	...
Application Protocol Layer	Token Management	Identity Management	Storage Management	Decentralized Governance	DLT Oracle	...
Contract Layer	Transaction Engine	Smart Contract				
Consensus Layer	PoW/PoS/DPoS/PBFT/Raft/etc.					
Session Layer	Transaction	Block	Account			
Transport Layer	TCP	QUIC	TLS			
Network Layer	DNS+IP	Overlay	Service Routing	Pub/sub		
Resource Layer	CPU	Storage	Transport Network			

DLT Interactions

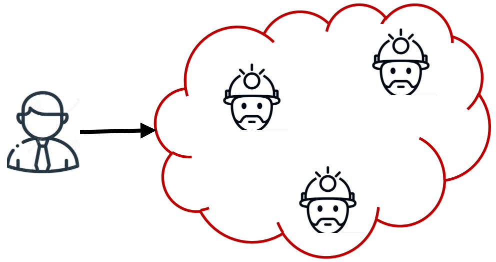
A client **commits** a transaction (request) to the DLT



A miner **commits** a found block to the DLT

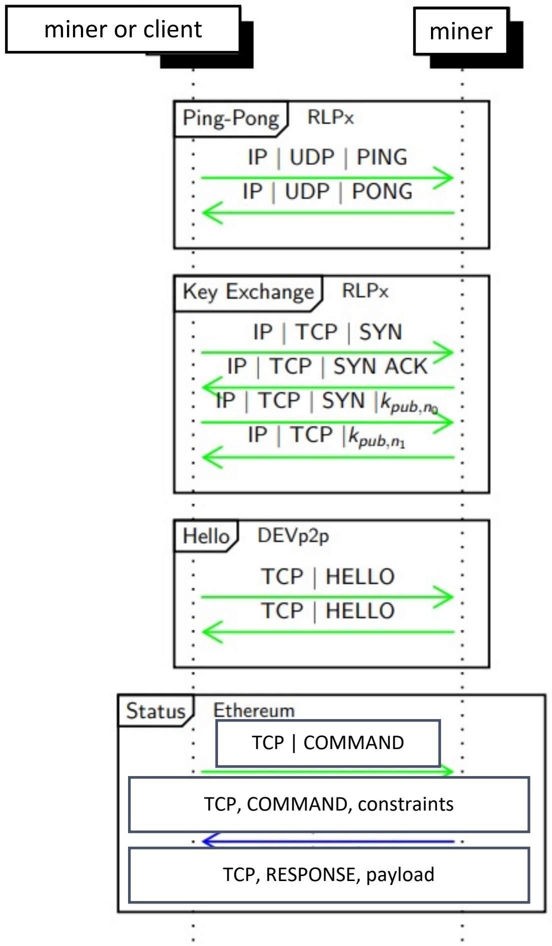


Any client or miner can **read** the blockchain in the DLT



All of those interactions are between originator and *N* peers, i.e. inherently **multipoint** in nature

Resulting Communication Patterns



Node discovery

Transport security

Keep alive, used to maintain a given sized (about 1500) **pool of peers** to communicate with for the transactions

Transactions to be executed at miners with RESPONSE specific to transaction type with miners selected from pool of peers

This may lead to disconnects with ***reachable miners*** if constraints do not match

Challenges Realizing DLT over Provider Networks

Problem 1: Information is required to reach other peers

- Bootstrap nodes maintain IP addresses of all peers (plus port information)
- New DLT members ***need to download routing information*** upon joining and for regular update

Problem 2: Clients know nothing about peers' capability to serve requests

- Approach is to (1) contact potential peer, (2) wait for connection, (3) inquire capabilities, (4) disconnect if not matching
- Peers ***may never reply*** to connection establishment (step 2)

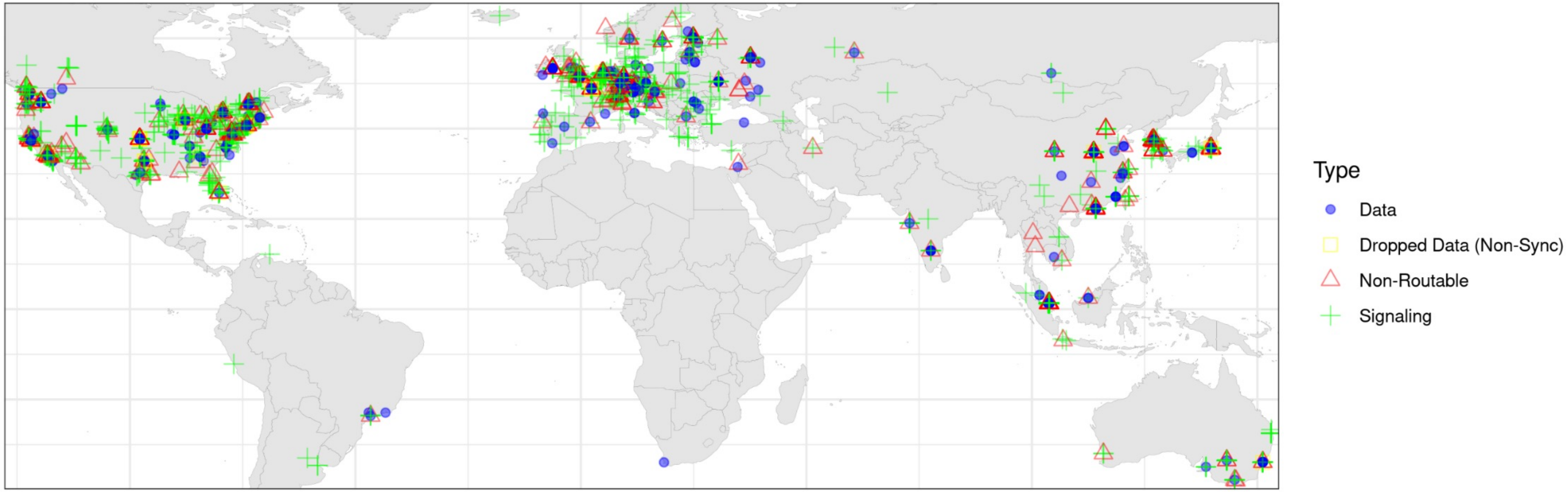
Problem 3: Peers map sending of transactions onto unicast communication

- Negatively impacts ***efficiency*** (bandwidth usage) and ***completion time***

Problem 4**:** Need to expose IP address to Bootstrapping Node

- Sending IP address during DLT sign-up may lead to ***privacy*** and/or ***security*** issues

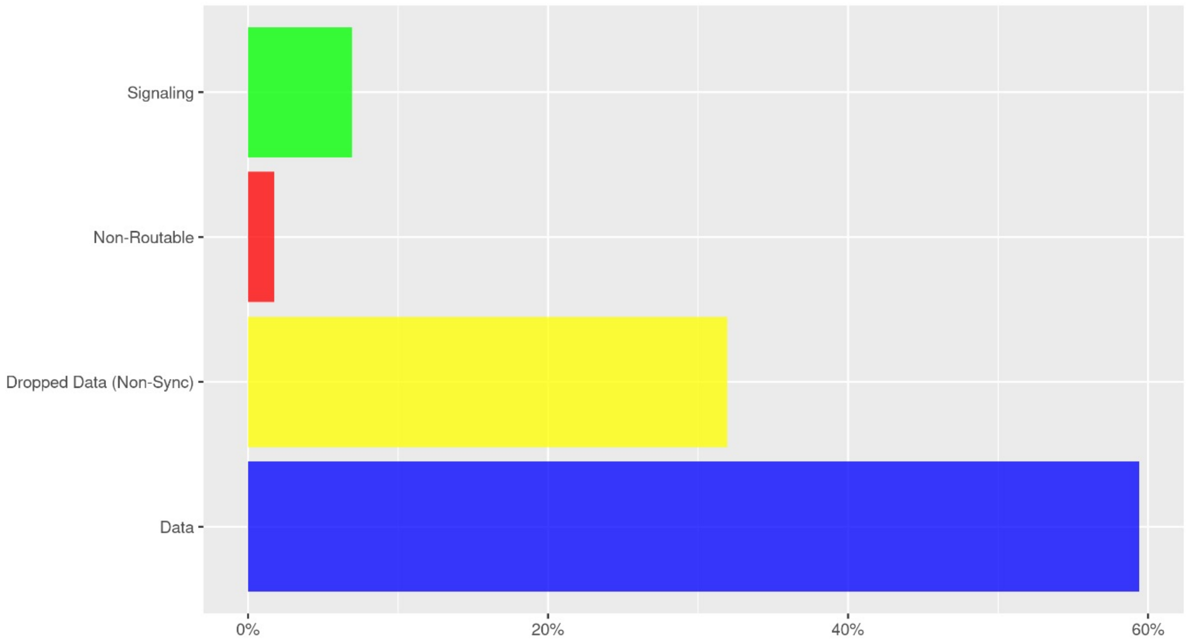
DLT Experiment to Find Good Peers

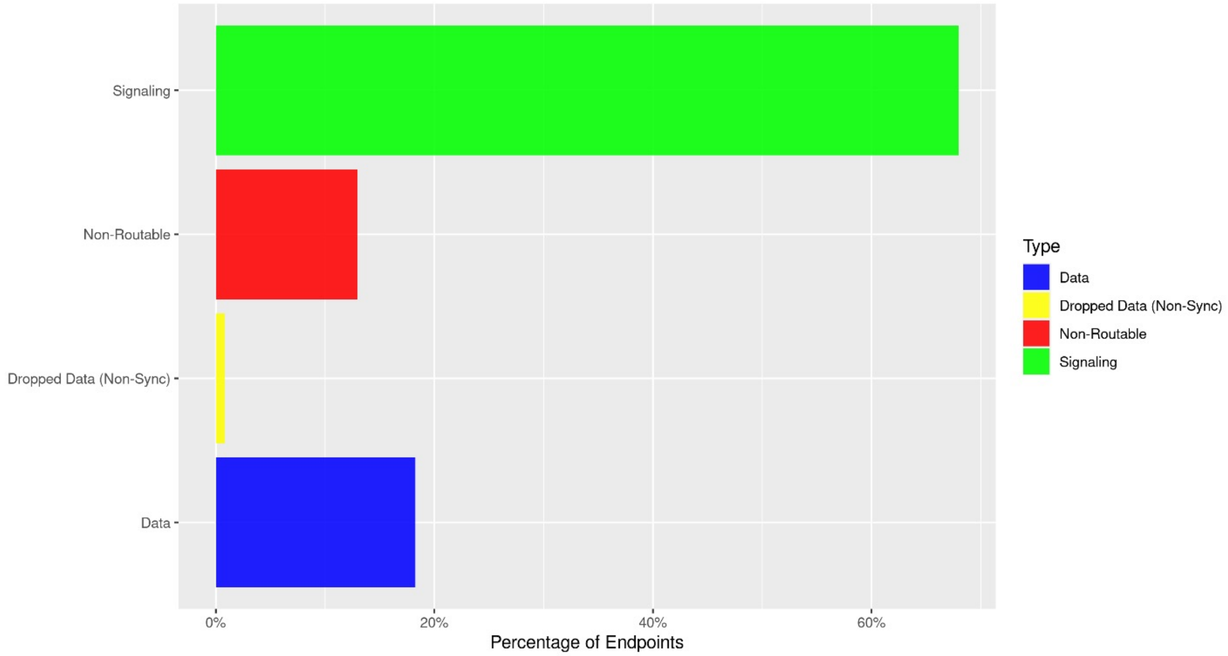


Conducted experiment with Ethereum to identify ***good*** nodes (blue) vs ***bad*** nodes (other colors)

- ***Good nodes*** are those responding with actual data transactions
- ***Bad nodes*** are those wasting communication due to disconnects, non-routability, purely signaling, …

Determining Wasted Bandwidth





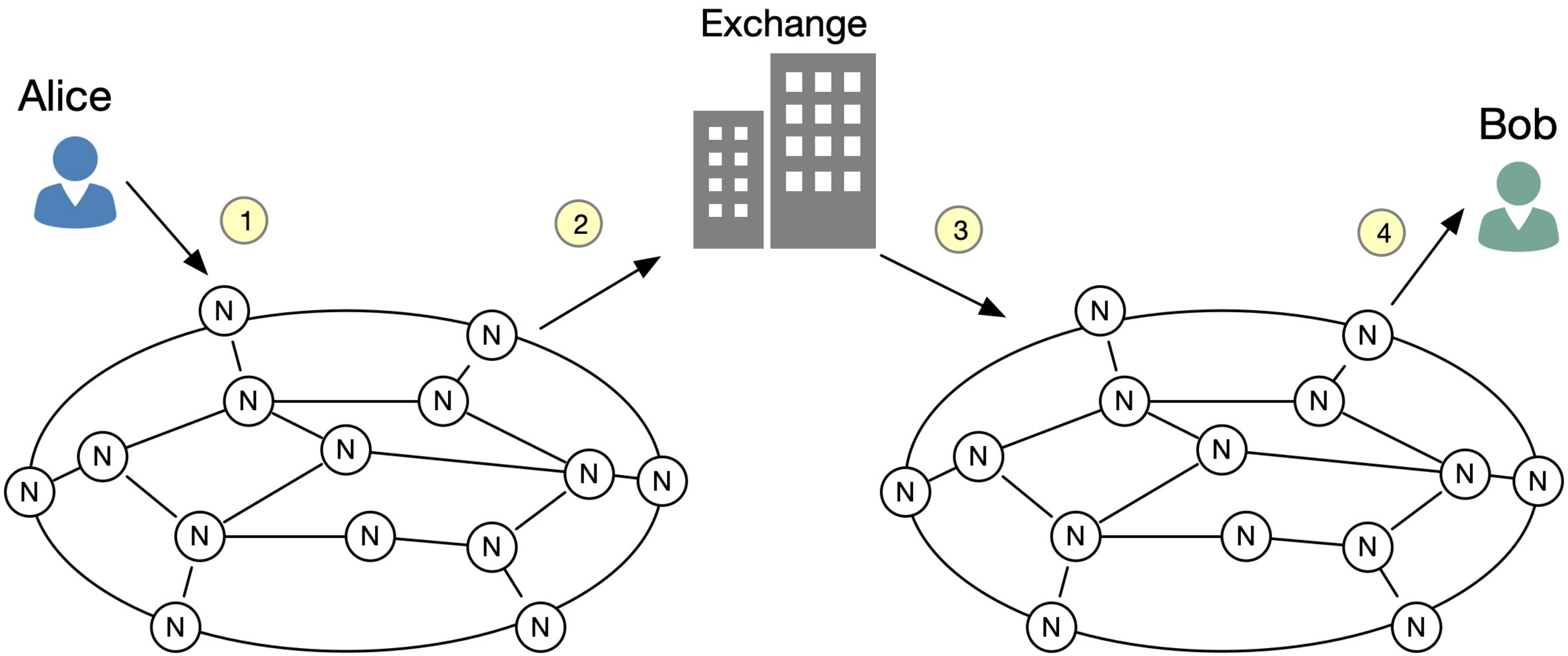
- Good nodes account for **only ~16%** of all nodes (with active node discovery in ETH)
- Bad nodes account for **~42% of wasted traffic**

Opportunities through Network Innovations

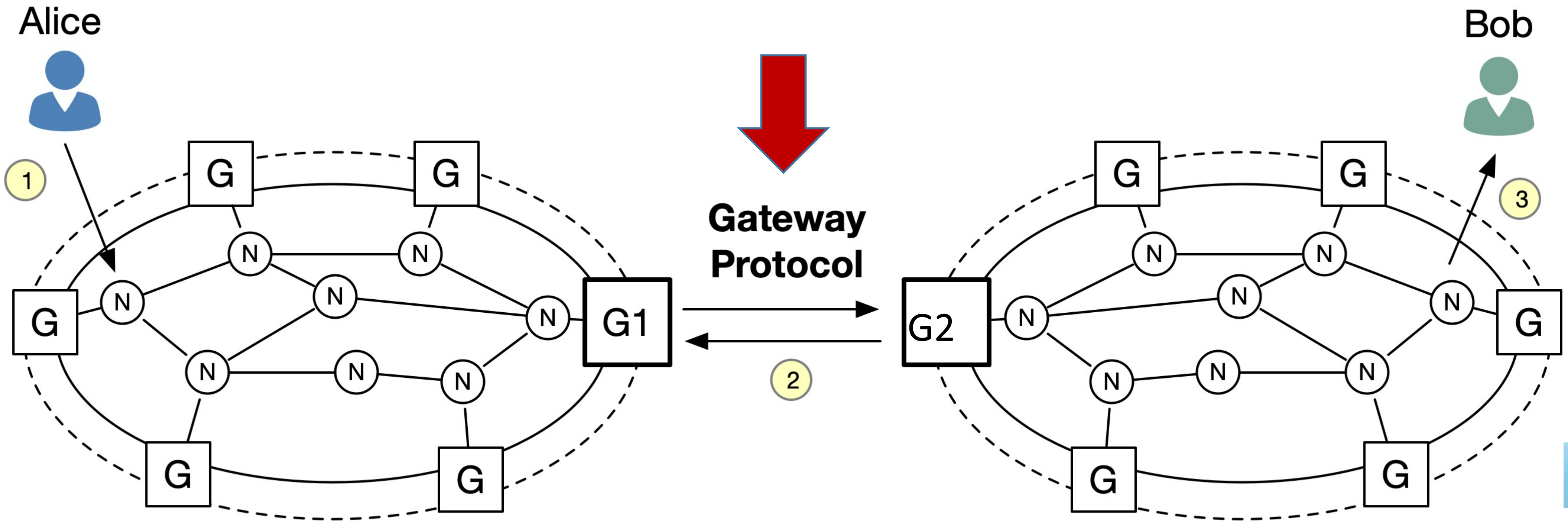
Interpret miners as ***service instances*** to ***mining*** service, e.g., miner.mydlt.org	Use of service announcement removes need to expose connectivity information to DLT and removes need for dedicated overlay infrastructure (***innovation 1***)
Utilize routing over service addresses (instead of IP addresses) to send *transactions* to ***ALL service instances***	Forward network-level multicast replaces overlay multipoint replication (***innovation 2***) Use entire pool of DLT peers, not a limited size as in current DLT (***innovation 3***)
Formulate ***constraints*** on <u>*capabilities*</u> (e.g., used hash), <u>*conditions*</u> (e.g., network diameter), and <u>*events*</u> (e.g., block computation, smart contract creation)	Forwarding service requests is dynamically constrained by those aspects that makes receivers accept the request, avoiding unnecessary disconnects (***innovation 4***)

Key here is to use (distributed) routing over services addresses (not overlay nodes), where constraints ensure the success of the intended communication

Transfers mediated by Third Party (Exchange)



IETF Proposal



draft-hardjono-blockchain-interop-arch-03

IETF Proposal

Phase 1: Pre-transfer Verification of Asset and Identities
Phase 2: Evidence of asset locking or escrow
Phase 3: Transfer commitment

```
    Originator
        |
+--------------+
|    Client    |
|(Application)|
+--------------+
        |
        |                 Phases
        V
+--------------+       |<-----(1)----->|       +--------------+
| Blockchain   |  +----+                +----+  | Blockchain   |
| System B1    |  |Gate|                |Gate|  | System B2    |
|              |--|way |<-----(2)----->|way |--|              |
| +----------+ |  | G1 |                | G2 |  | +----------+ |
| |Ledger L1|  |  +----+                +----+  | |Ledger L2|  |
| +----------+ |       |<-----(3)----->|       | +----------+ |
+--------------+                               +--------------+
```

draft-hargreaves-odap-03

Standardization – IEEE

Published Standards: 6

- 2140.1-2020 - IEEE Standard for General Requirements for Cryptocurrency Exchanges
- 2140.5-2020 - IEEE Standard for a Custodian Framework of Cryptocurrency
- 2142.1-2021 - IEEE Approved Draft Recommended Practice for E-Invoice Business Using Blockchain Technology
- 2143.1-2020 - IEEE Standard for General Process of Cryptocurrency Payment
- 2144.1-2020 - IEEE Standard for Framework of Blockchain-based Internet of Things (IoT) Data Management
- 2418.2-2020 - IEEE Approved Draft Standard Data Format for Blockchain Systems

Standards Under Development: 52

Standardization – IEEE

- P2418.1 Blockchain Use in IoT
- P2958 Decentralized IAM
- P2145 Blockchain Governance


Q&A