Impact of DLTs on Provider Networks

Dirk Trossen (Huawei)
David Guzman (Huawei)

Starting Point of Our Work

Perspective of the DLT Application:

- DLTs do not typically care about the underlying IP network

- DLTs use a fixed size ***pool*** of transport connections, constantly changing over time, to maintain application communication

- They focus on securing their application and do not worry about the network

Perspective of the Network: What is the impact of choices made by the application design on the network, e.g., in terms of costs, traffic generated etc.?

Goal for Our Work

Our work aims to understand the impact of DLTs on provider networks and the possible opportunities to improve on those impacts

Extend our discussion initiated originally in IIC Whitepaper, extended in current RTG WG draft, with same title!

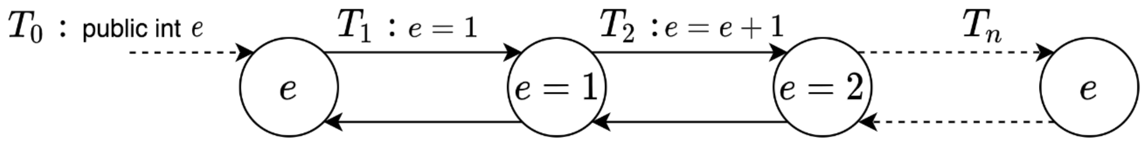
Interactions in a DLT over IP

Example: Ethereum

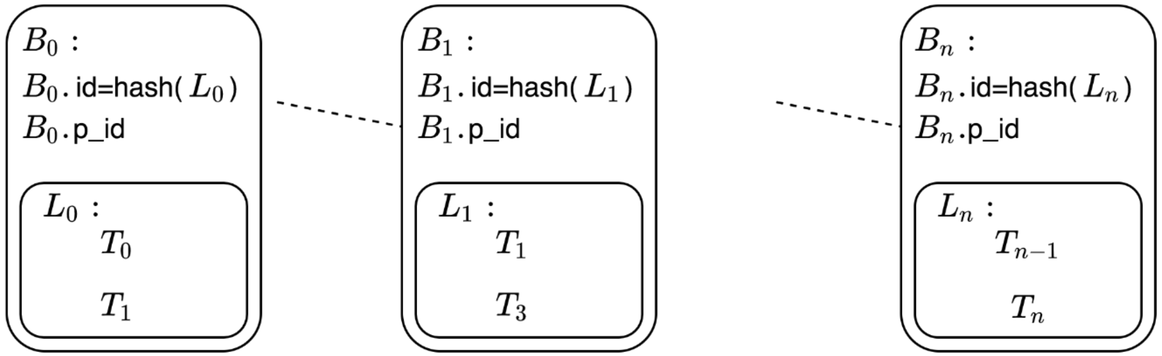
Distributed Consensus System

Main Concepts: Transactions over a Distributed Ledger

- **Transaction:** A cryptographically-signed instruction or set of instructions to modify the state machine. It can be a structure (BTC, Monero), a script (ETH) or a chunk of instructions



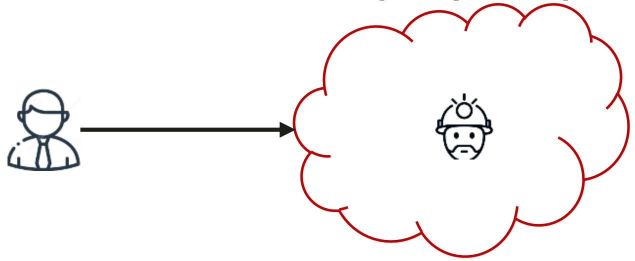
- **Ledger:** A set of valid transactions
- **Block**: a hashed ledger
- A set of concatenated blocks is a **blockchain**
- To find a hash with a specific pattern (leading 0s) is called **PoW a.k.a. mining**



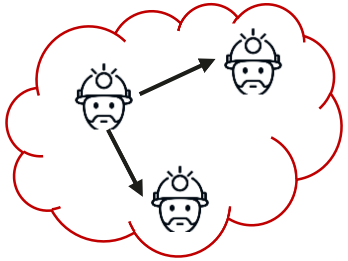
Distributed Consensus System

Interactions

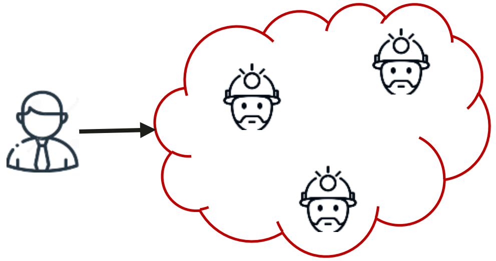
A client **commits** a transaction (request) to the DCS



A miner (peer) **commits** a found block to the DCS



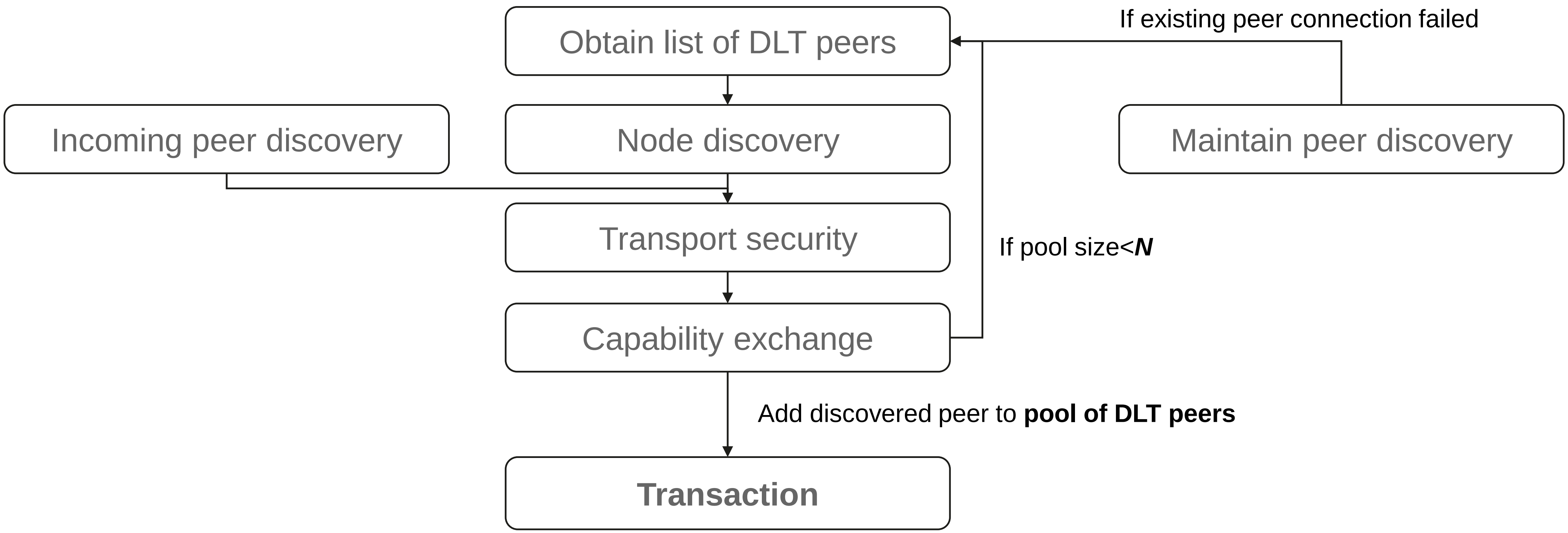
Any client or miner can **read** the blockchain in the DCS



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

Resulting Communication Patterns

(reverse engineered from the Ethereum code base)



HUAWEI

Challenges When Realizing Patterns over IP

Defining Our Opportunity Space

Problem 1: Information is required to reach other miners

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

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

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

Problem 3: Clients 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 (& Peers)

- Sending one's IP address as part of signing up to DLT may lead to ***privacy*** (e.g., exposing topological location of peer) and/or ***security*** issues
- Difficult to handle ***IP address changes*** or ***supporting mobility***

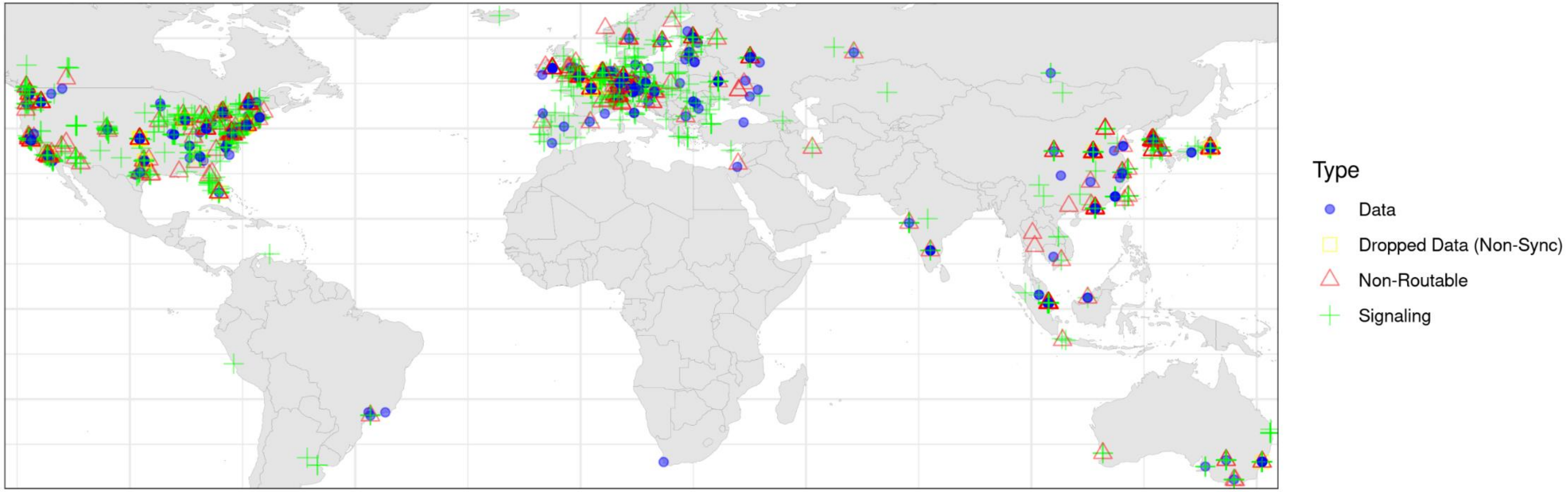
On Using Multicast for DLTs…

Challenges identified from our Ethereum Insights

- **Highly individualized** operations
  - seeding from bootstrap nodes, discovery of other peers, constantly changing pool of transport connections
    -> either no or **frequently changing multicast relations**
- **Highly distributed** DLT network
  - Discovered nodes, i.e. potential members of DLT pool, may reside ANYWHERE within the geo spread of the DLT, i.e. likely across ASs
    -> **inter-domain support** for multicast poses a problem, possibly requiring hybrid approaches (e.g., replay nodes)
- **Highly dynamic** pools per peer
  - Pools are constantly refreshed to randomize membership
    -> meaning the pool of peers undergoes **constant changes**, possibly incurring **high membership signaling**
- **Highly diverse peers** in overall DLT network
  - May range from individual at home over hosted VM in cloud to entire private clouds
    -> **multicast may or may not be supported** in local domain or enabled for peer

Of course, one could look deeper into the communication patterns and achieve same purpose but through more multicast-friendly patterns (this was not our angle of investigation though)

DLT Experiment to Find *Good* vs *Bad* Peers



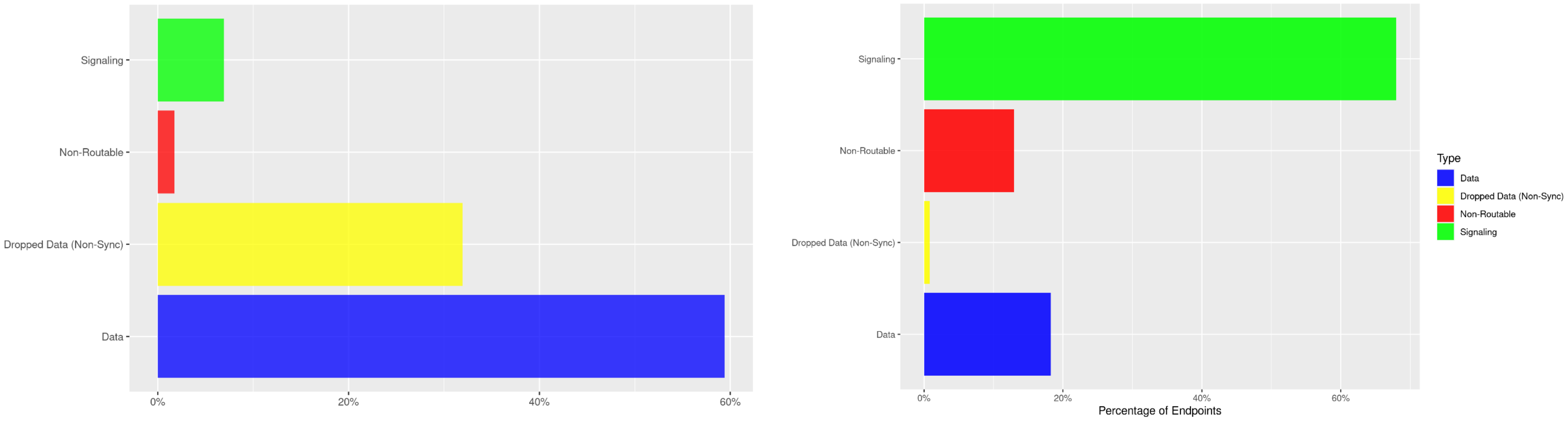
Conducted experiment with Ethereum to identify ***good*** nodes (blue) vs ***bad*** nodes (other colors)

Good nodes are those responding with actual data transactions

Bad nodes are those wasting communication due to disconnects, non-routability, purely signaling, not sending complete blockchain despite positive signaling, …

HUAWEI

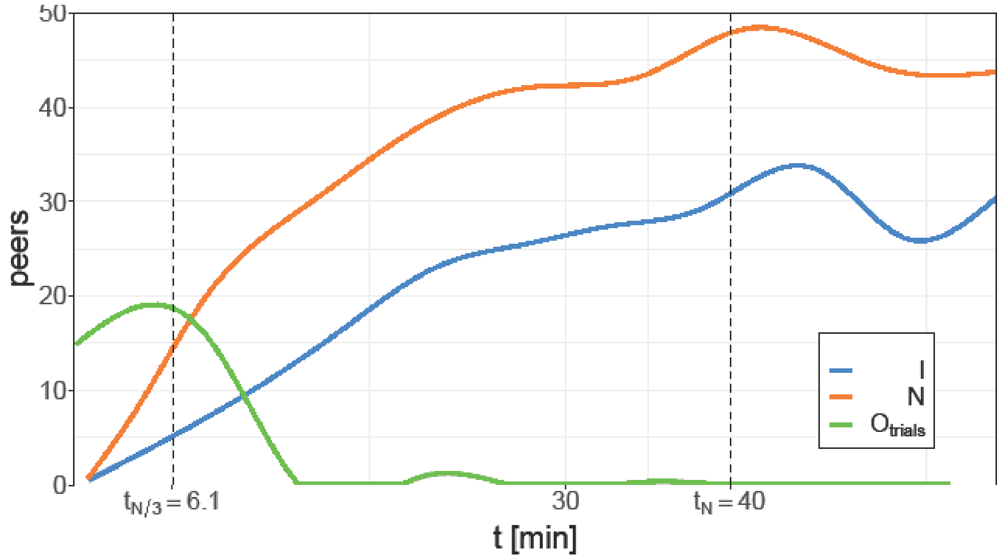
Determining Wasted Bandwidth



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

Other KPIs

Ongoing Work (under submission)



- Pool Establishment Time
  - > How does it take to create a suitable overlay network structure?

  -> on average ~36mins!

- Pool Establishment Cost
  - > How many peers need to be contacted overall to sustain the required pool size N?

  -> on average ~5k peers for a pool size of 50 until the end of the pool establishment time!
  - > How much traffic is being generated?

- Effective Data Consumption
  - > The amount of useful data retrieved from DLT (i.e. a valid blockchain) vs total downloaded data

  -> only about 32% of downloaded BC data is useful!

Opportunities through Network Innovations

Example: Routing on Service Addresses

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

Utilize routing over service addresses (instead of IP addresses) to send *transactions* to ***ALL service instances***

→ Network-level multicast with in-network diffusion support, i.e., **diffused sending to N out of M>>N** service instances, using entire pool of DLT peers, not limited size as in current DLT

Formulate ***constraints*** on *capabilities* (e.g., used hash), *conditions* (e.g., network diameter), and *events* (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

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

What Next?

- Extend our work to other DLTs to understand differences in overlay management
  - Other PoW systems
  - PoS and its impact on overlay management
  - …
- Extend our insights into Ethereum as PoW example
  - More KPIs
- Look into other network innovations
  - LISP, …
- Look into impact of network innovations
  - Improvement over IP overlay

More importantly, we are looking for **collaborators** to deepen our insights

Conclusions

- As an overlay, DLTs have a clear impact on provider networks in terms of traffic generated
  - Understanding this impact is important for many reasons

- We attempt to formulate a methodology to think about the evaluation of DLTs and their possible improvement through network innovations

- We aim to identify networking innovations that may improve on the impact and performance of DLTs overall

This work has only started and we are working towards more insights and collaborations alike

Thank you.

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI