BlockStorm: Scalable Block Diffusion for Permissioned Distributed Ledgers

Gregory Chockler
Royal Holloway, University of London

Artem Barger, Yacov Manevich, IBM Research Haifa
Peter Robinson, Royal Holloway, University of London

Scalable Block Diffusion

- Motivation: Hyperledger Fabric
- Approaches to solution
- Adversarial Gossip Problem
- Ongoing and future work

Scalable Block Diffusion

- Motivation: Hyperledger Fabric
- Approaches to solution
- Adversarial Gossip Problem
- Ongoing and future work

Blockchain Scalability

- Three major performance bottlenecks
  - Transaction ordering
  - Transaction validation
  - Transaction execution

Transaction Ordering

- **Permissionless ledgers**
  - Proof-of-work
  - Slow and costly: 7 tx/sec, 10+ min to confirm
- **Permissioned ledgers**
  - Consortium/Committee consensus
    - e.g., BFT
  - Fast: 22.2K tx/sec for 8 nodes
  - But performance declines with the system size

[Croman et al 2016]

Transaction Execution & Validation

- CPU and storage bound
  - Hampers throughput if collocated with ordering
  - Does not require distributed coordination

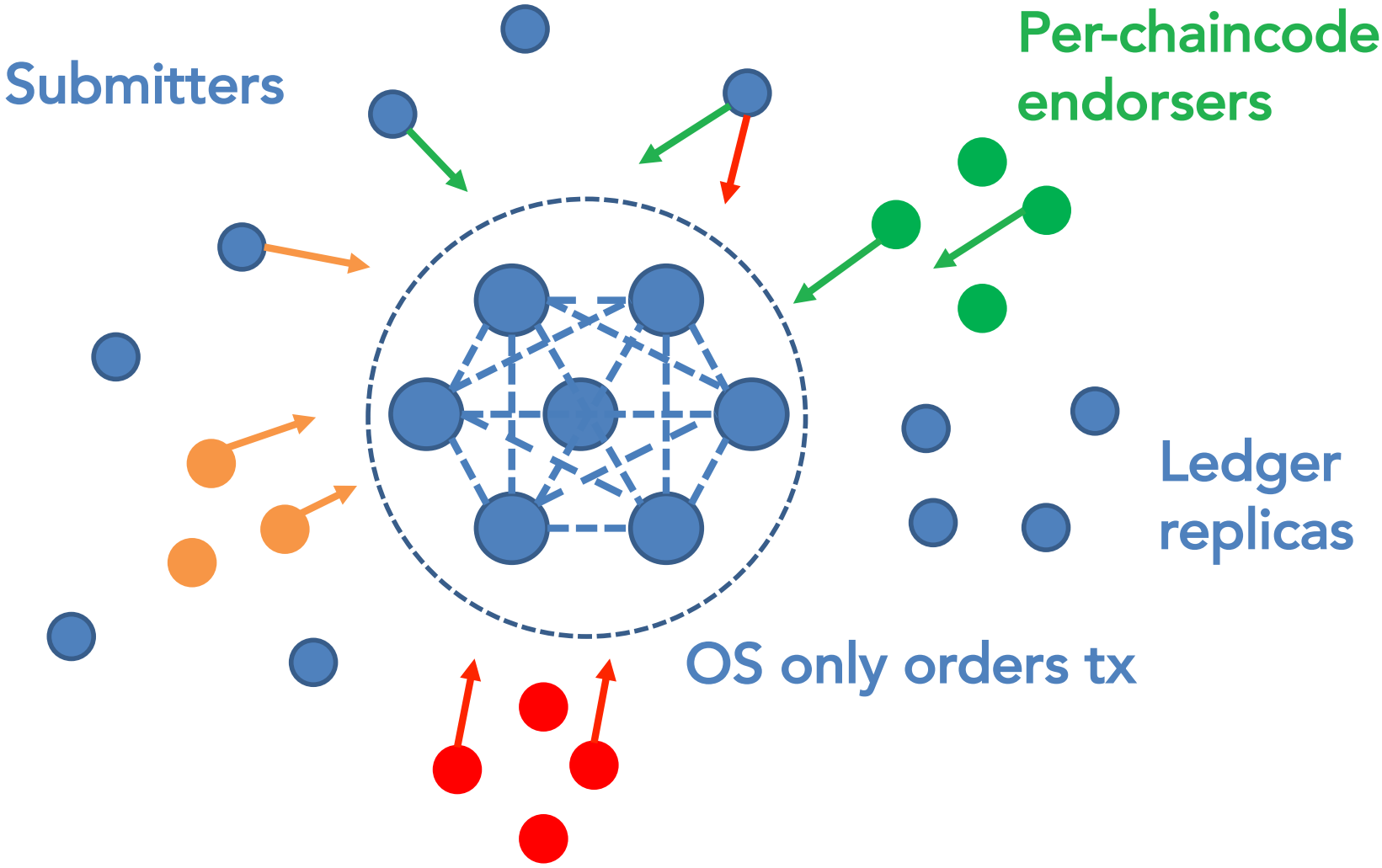
- Hyperledger Fabric: best of both world

Hyperledger Fabric (HLF)

- Dedicated ordering service (OS)
  - Small number of trusted processes
  - Consortium consensus (e.g., BFT)

- Endorsing and validation offloaded to a separate tier of processes (peers)
  - Can be scaled independently
  - Does not need a strong trust model

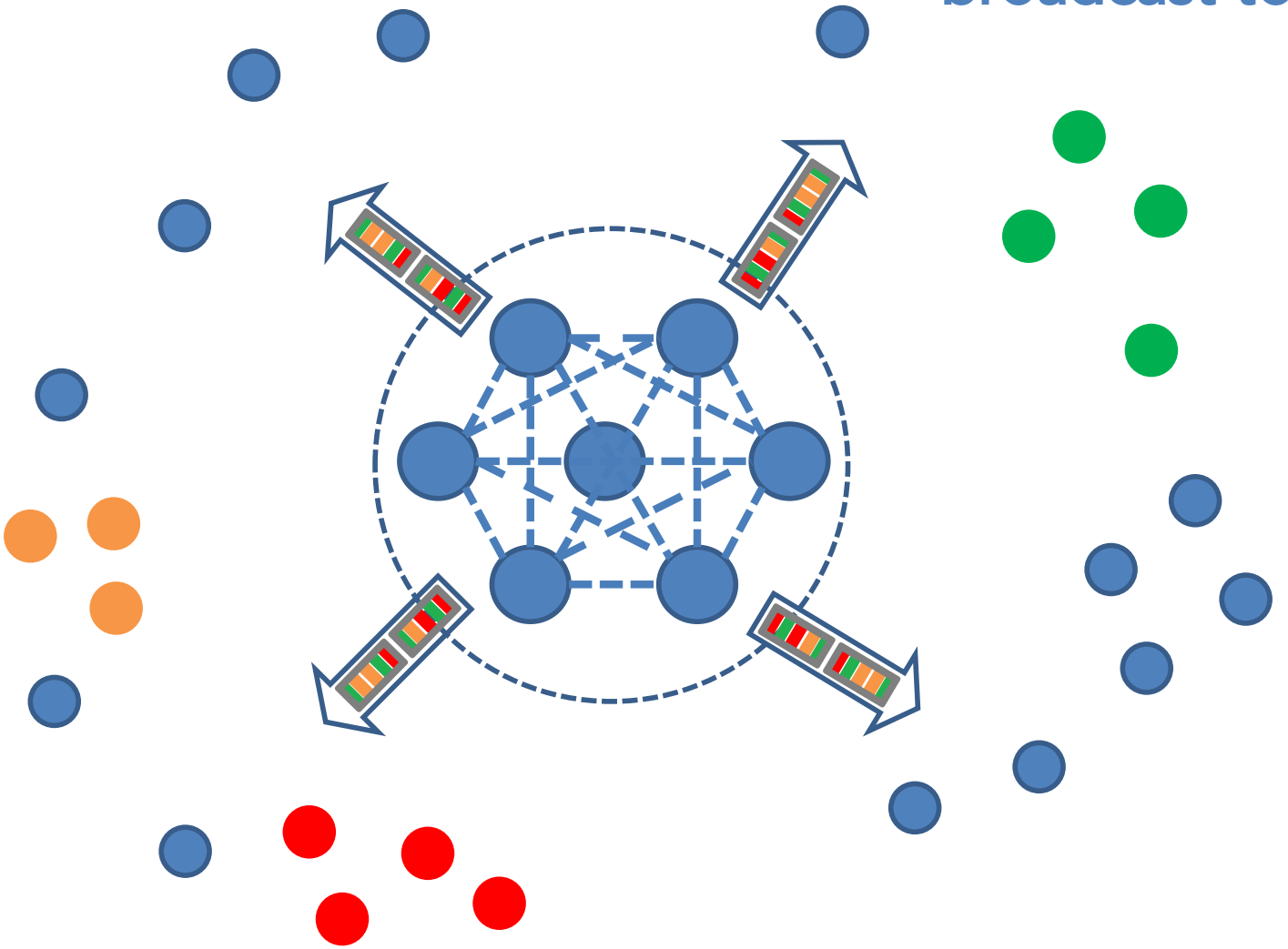
Hyperledger Fabric (HLF)



Slide credit: Christian Cachin

Hyperledger Fabric (HLF)

Ordered tx blocks are broadcast to all peers



Challenges

- High incoming transaction rates
- Large numbers of peers
  - Grows w/number of installed smart contracts
- Dynamic peers
- Compromised peers

Design a high throughput broadcast layer which is scalable, secure and robust under attacks

Assumptions

- **Blocks are injected by the OS**
  - Reliable source
  - Blocks are signed by the OS
    - Group or threshold sig
  - All peers can verify the OS signature
- **Peers can be compromised**
  - Can fail to propagate blocks (send/receive)
  - Cannot fake the blocks
  - Adversary can be adaptive

Scalable Block Diffusion

- Motivation: Hyperledger Fabric
- Approaches to solution
- Adversarial Gossip Problem
- Ongoing and future work

Fixed Diffusion Topologies

- **Star: peers connect directly to the OS**
  - ✓ Efficient block authentication
  - ✗ OS is the throughput bottleneck
- **Spanning tree rooted at the OS**
  - ✓ Low communication overhead
  - ✗ Low resiliency

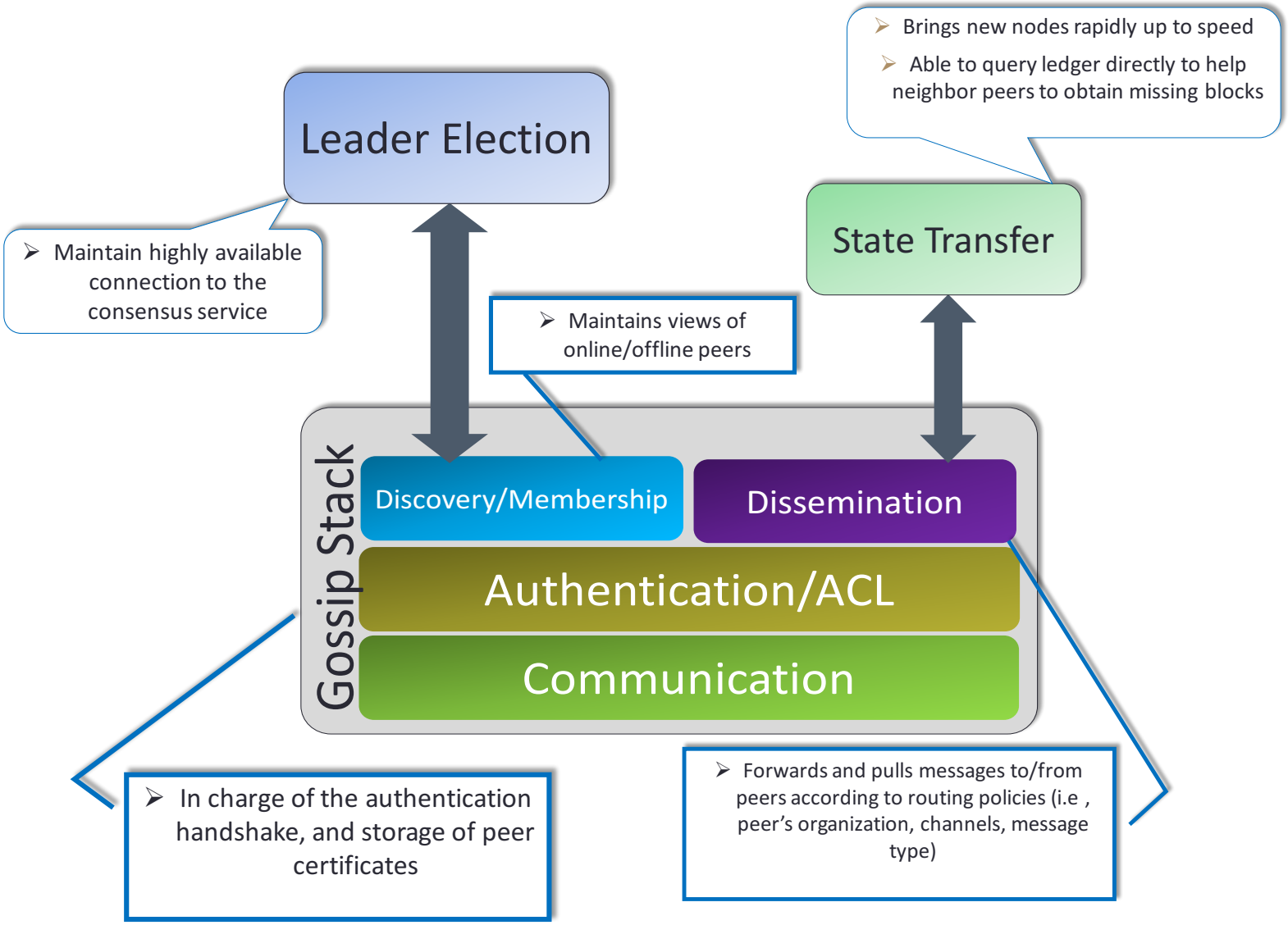
Overlay Networks

- **Long-lived topologies with well-defined guarantees**
  - Connectivity, diameter, degree
- **Highly complex to create and maintain in adversarial settings**
  - Bitcoin overlay is known to have problems
  - No solutions with provable guarantees
  - Future work…

Epidemic Diffusion

- **Proceeds in rounds**
- **Every round:**
  - Choose random partners
  - Push/Pull blocks
  - Discard old blocks from the buffer (if needed)
- **Benefits:**
  - Simple to implement
  - Inherently robust
    - Never gets stuck with "bad" partners
  - Decentralized
  - Scalable

BlockStorm: High-Level Architecture



Preliminary version: SYSTOR'17

Scalable Block Diffusion

- Motivation: Hyperledger Fabric
- Approaches to solution
- Adversarial Gossip Problem
- Ongoing and future work

Adversarial Gossip Problem

- **Late ε-bounded Adaptive Adversary**
  1. Can observe the past system state up to round t – 1 for all t ≥ 2
  2. ≤ εN peers can be attacked in each round

- **Analyse performance for a single block M**
  - Latency and message complexity as a function of
  - Propagation strategy: push/pull/fanout

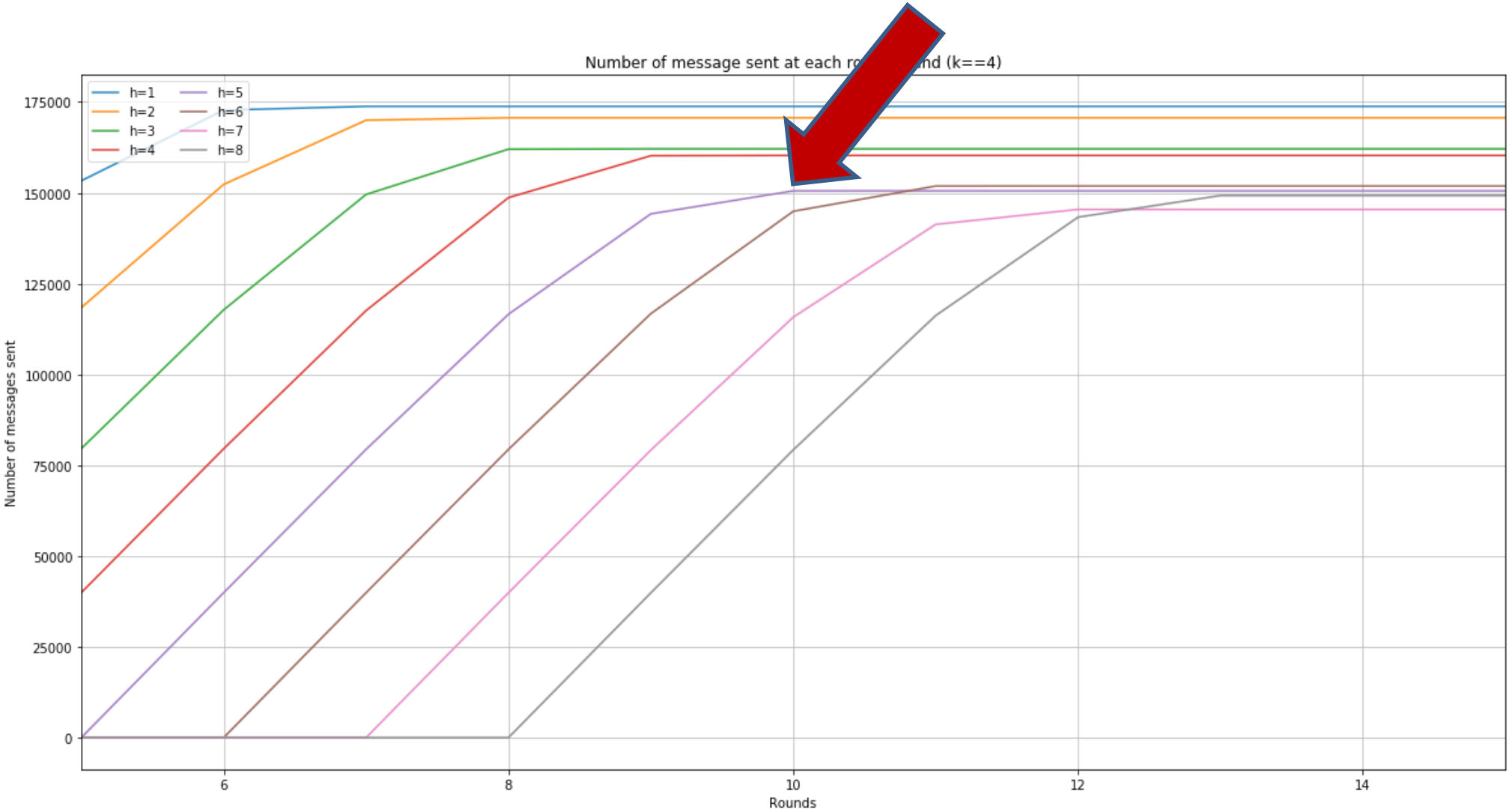
Results so far

- **Best propagation strategy:**
  1. Forward M once when first received until
  2. ≥N/2 peers are infected, then switch to pull

  Forward fanout: $\Theta(\log(N))$, t=1; $\Theta(1)$, t > 1
- **Best adversarial strategy:**

  Every round t ≥ 2: attack all peers that are known to not have received M as of the round t-1 with probability ε

Propagation Horizon



Scalable Block Diffusion

- Motivation: Hyperledger Fabric
- Approaches to solution
- Adversarial Gossip Problem
- Ongoing and future work

Ongoing work

- Simulations to verify the theoretical analysis and clarify the constants
- Incorporating the optimal strategy into BlockStorm
- Large-scale performance study on a cloud testbed

Future Work

- **Boosting efficiency of block verification**
  - Combine signatures with hash chains/trees
- **Beyond Hyperledger**
  - Block propagation in permissionless ledgers and hybrid blockchains [PS16], ByzCoin [Kogias et al.]
- **Adversarial overlay networks**
- **Multicast for sharded ledgers**

Thank You!

Assumptions

- **Self-verifiable block content**
  - Signed by the OS signature
- **Synchronous computation**
  - Can be loosely synchronized in practice
- **Late ε-bounded Adaptive Adversary**
  - Can observe the past system state up to round t – 1 for all t ≥ 2
  - ≤ εN peers can be attacked in each round

Blockchain Scalability

- **Proof-of-Work:**
  - Strong guarantees vs. performance tradeoff
  - Unacceptable for enterprise/mission critical settings
- **BFT:**
  - Well-suited for permissioned ledgers
  - High throughput, but cannot scale to large groups

- **Solution: separation of concerns ➔**