BLS cosigning via a gossip protocol

Semester project (master)

Lukas Gelbmann

DEDIS / EPFL

Responsible: Prof. Bryan Ford

Supervisors: Cristina Basescu and Gaylor Bosson

Motivation

- for **cosigning**: ensure that a message has been seen and verified by many peers
- for cosigning **over gossip protocol**: more robust than our current implementation

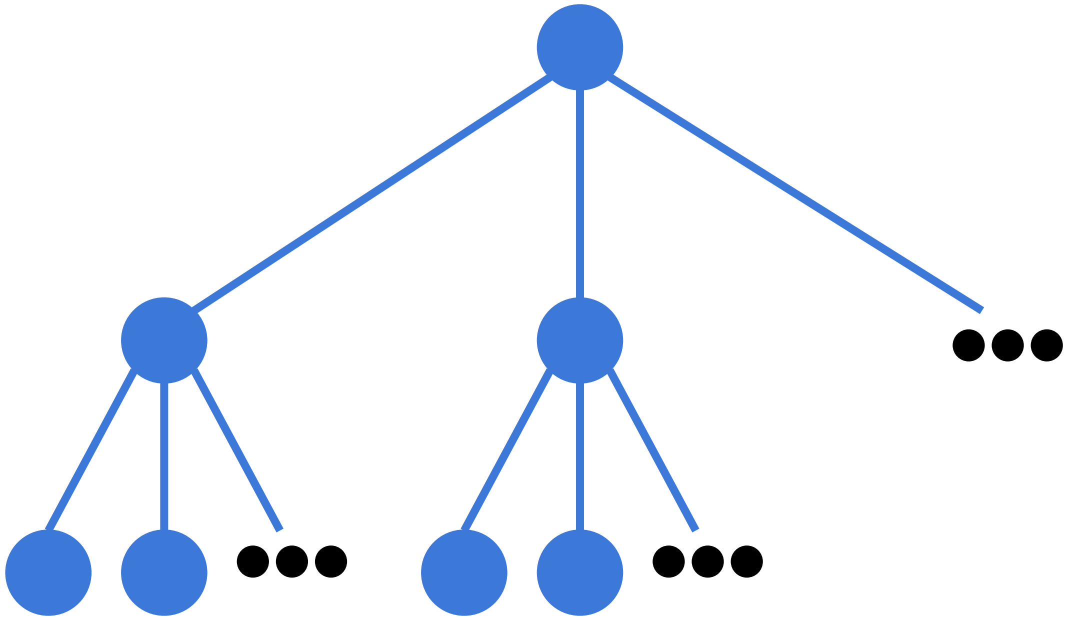
Roadmap

1. **Problem statement**
2. Old protocol
3. Gossip
4. New protocol
5. Crypto: BLS signatures
6. Results
7. Future work

Problem statement

- Build a cosigning protocol:
  - **Fault tolerant** (Byzantine failure model)
  - **Fast** (seconds or less)
  - **Don’t overload** any nodes
- Secondary goal: efficiency

Old tree-based protocol



1. Problem statement
2. Old protocol
3. **Gossip**
4. New protocol
5. Crypto: BLS signatures
6. Results
7. Future work

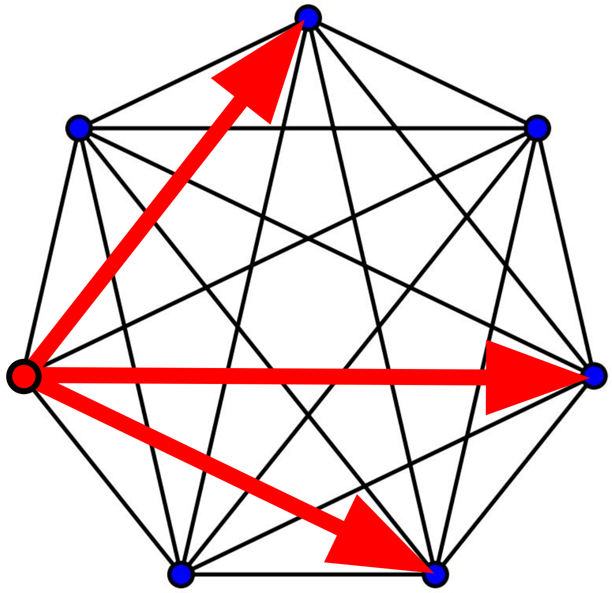
Gossip: our use case

- **Short-lived** protocol
- Each node knows every other node
- Each node has information to be gossiped: its signature
- At the **start**, the goal is to spread the message quickly
- At the **end**, the goal is to get the signatures back to the root node without overloading the node

1. Problem statement
2. Old protocol
3. Gossip
4. **New protocol**
5. Crypto: BLS signatures
6. Results
7. Future work

New protocol

- **Push rumor messages** to random peers in regular interval
- After the root has enough signatures
  - return cosignature
  - spread shutdown messages



1. Problem statement
2. Old protocol
3. Gossip
4. New protocol
5. **Crypto: BLS signatures**
6. Results
7. Future work

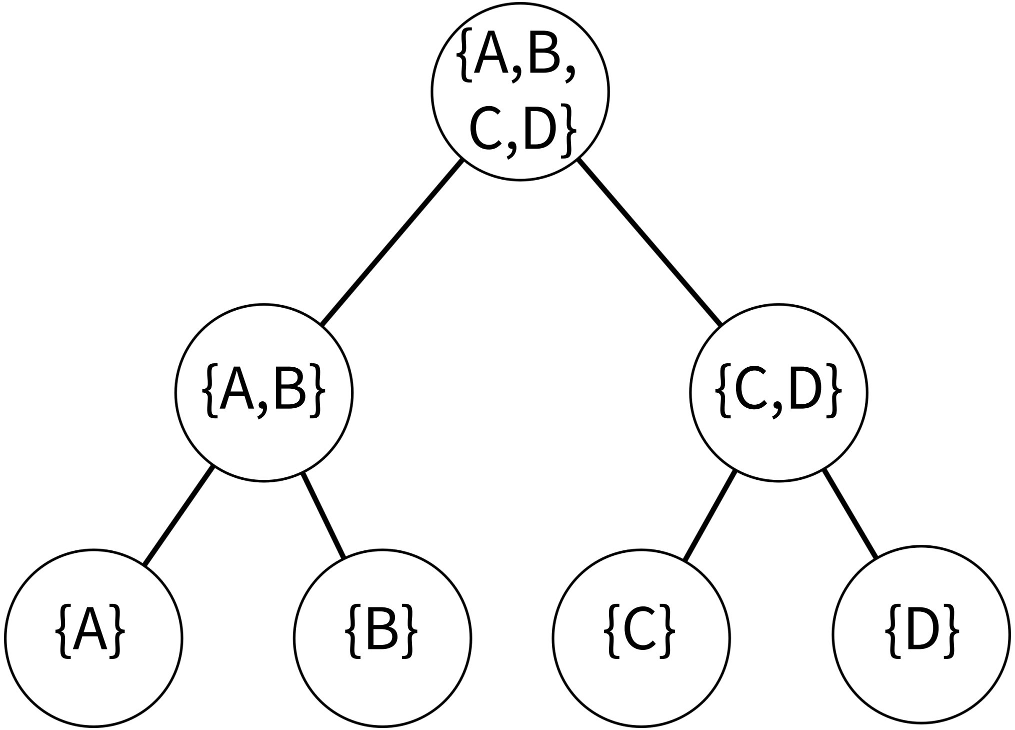
Crypto: BLS signature aggregation

- Easy to aggregate: signature from **{A, B}** and one from **{C}** into a signature **{A, B, C}**
- **Overlap** is hard to deal with:
  signatures from **{A, B}** and **{B, C}**

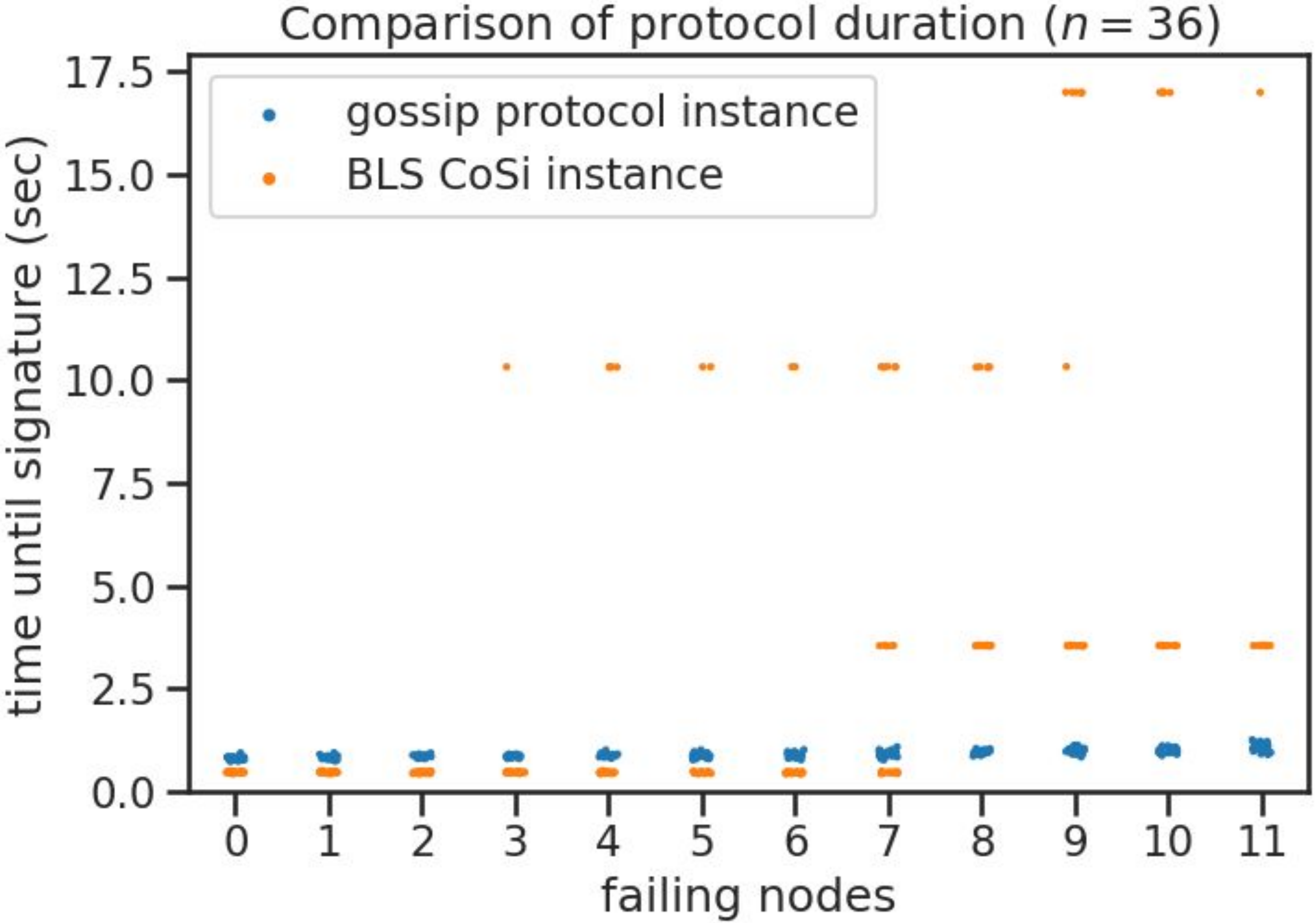
Crypto: BLS signature aggregation

- **Simple solution**: aggregate signatures only at the very end
- **Better solution**: binary tree

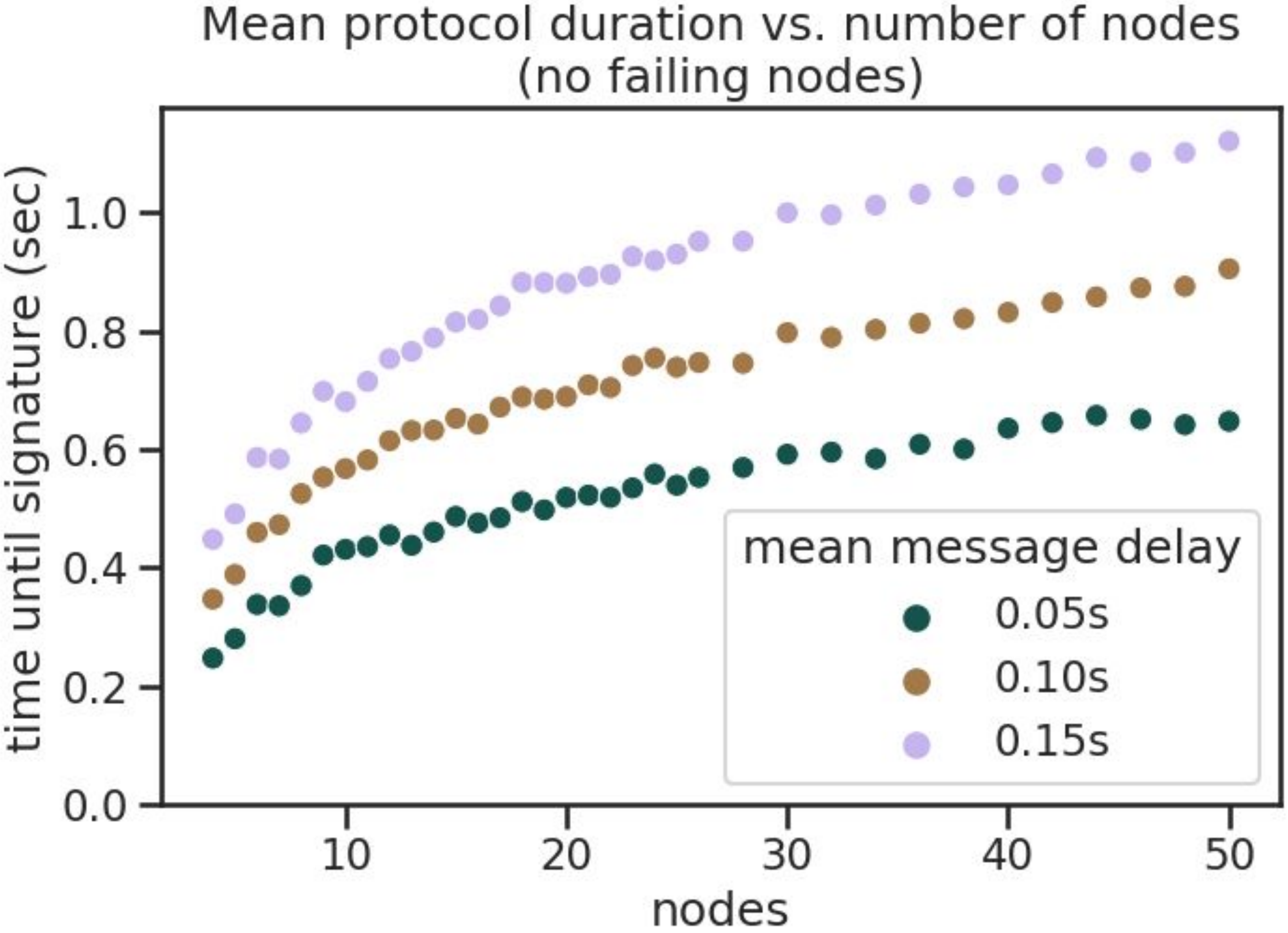
Signature aggregation rule

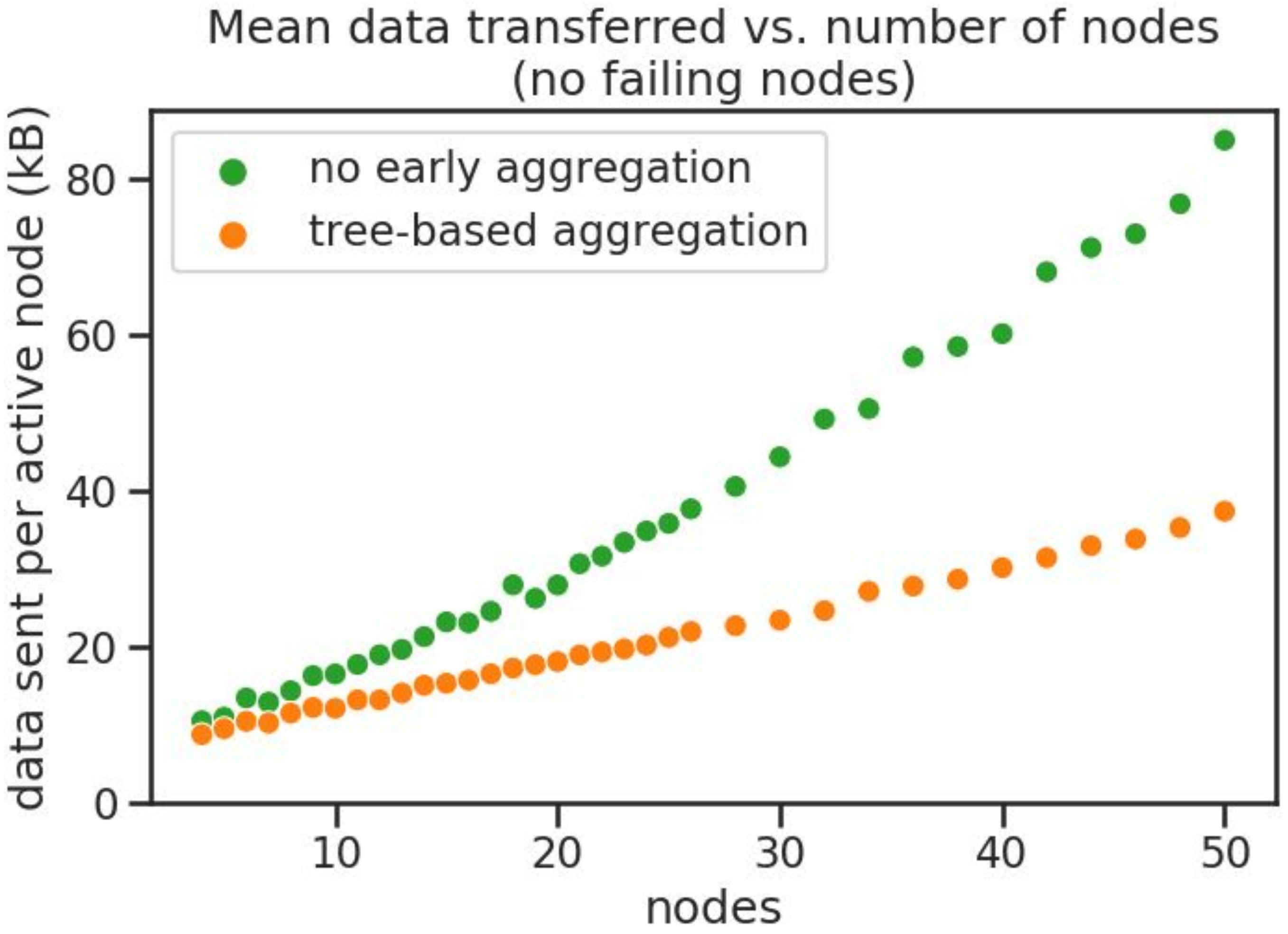


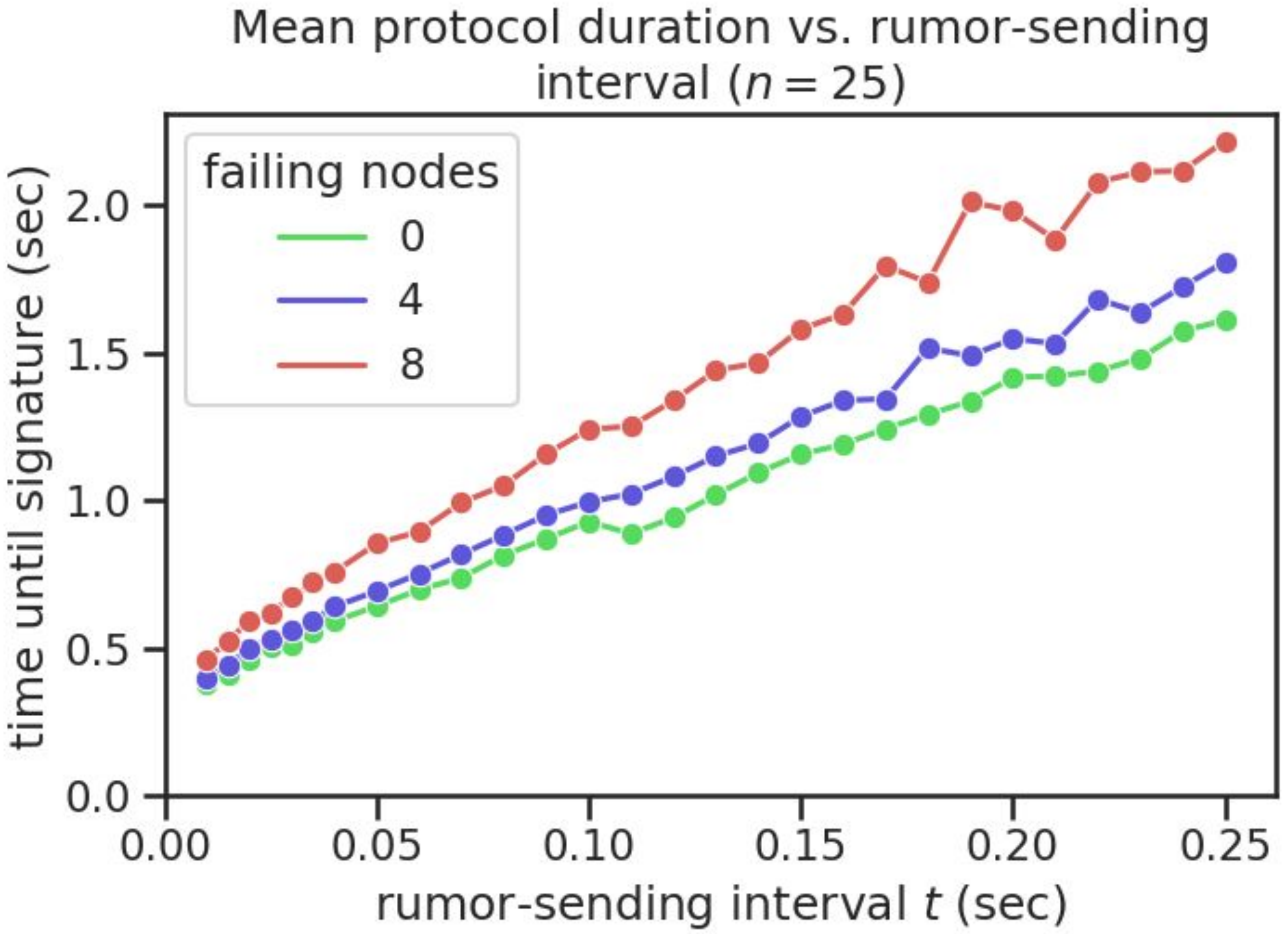
1. Problem statement
2. Old protocol
3. Gossip
4. New protocol
5. Crypto: BLS signatures
6. **Results**
7. Future work

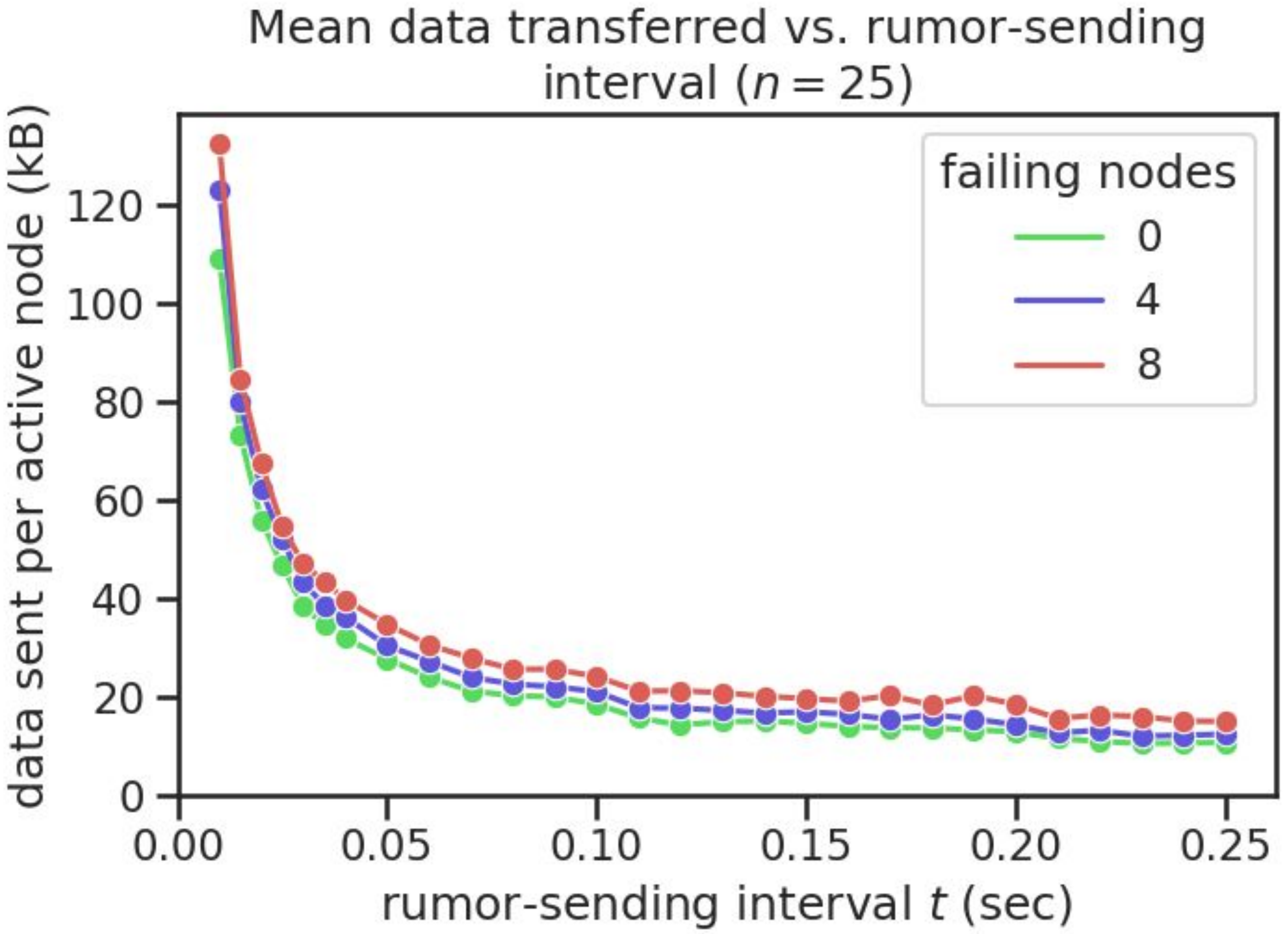


Comparison of protocol duration ($n = 36$)
gossip protocol instance
BLS CoSi instance
time until signature (sec)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
failing nodes
0
1
2
3
4
5
6
7
8
9
10
11


Mean protocol duration vs. number of nodes
(no failing nodes)
time until signature (sec)
1.0
0.8
0.6
0.4
0.2
0.0
10
20
30
40
50
nodes
mean message delay
0.05s
0.10s
0.15s


Mean data transferred vs. number of nodes
(no failing nodes)
no early aggregation
tree-based aggregation
data sent per active node (kB)
0
20
40
60
80
10
20
30
40
50
nodes


Mean protocol duration vs. rumor-sending interval ($n = 25$)
failing nodes
0
4
8
time until signature (sec)
0.0
0.5
1.0
1.5
2.0
0.00
0.05
0.10
0.15
0.20
0.25
rumor-sending interval t (sec)


Mean data transferred vs. rumor-sending interval ($n = 25$)
failing nodes
0
4
8
data sent per active node (kB)
0
20
40
60
80
100
120
0.00
0.05
0.10
0.15
0.20
0.25
rumor-sending interval t (sec)

1. Problem statement
2. Old protocol
3. Gossip
4. New protocol
5. Crypto: BLS signatures
6. Results
7. **Future work**

Future work

- **Possible optimizations**
    - Pull messages
    - Strategic peer selection
    - Better aggregation

Thank you for your attention

Your turn