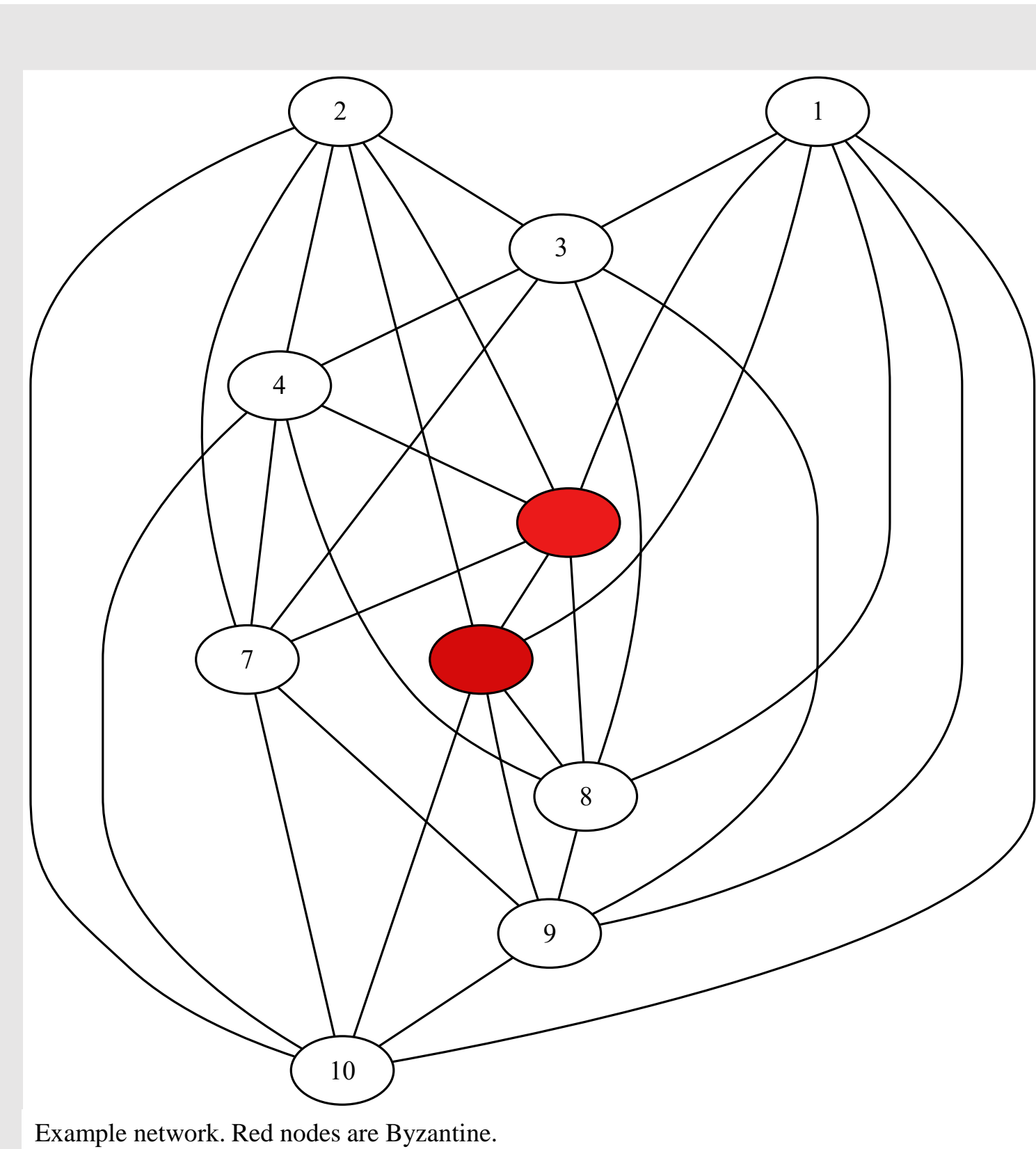


Byzantine Reliable Broadcast with signatures

(1) Background

- **BRB** – Broadcast to entire network in the presence of Byzantine nodes, where the broadcaster can also be Byzantine
- **Reliable Communication** - Broadcast to entire network in the presence of Byzantine nodes, where the broadcaster cannot be Byzantine.
- **Bracha's algorithm** – BRB in fully connected networks
- **Dolev's algorithm** – RC in $2f+1$ - connected networks
- **Bracha-Dolev** – BRB in $2f+1$ connected networks

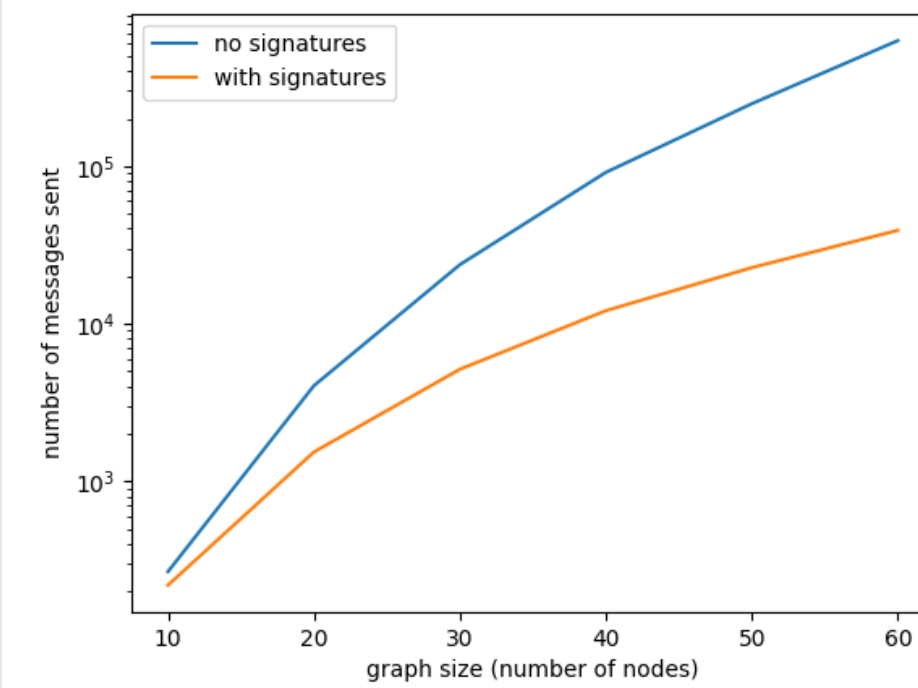


Example network. Red nodes are Byzantine.

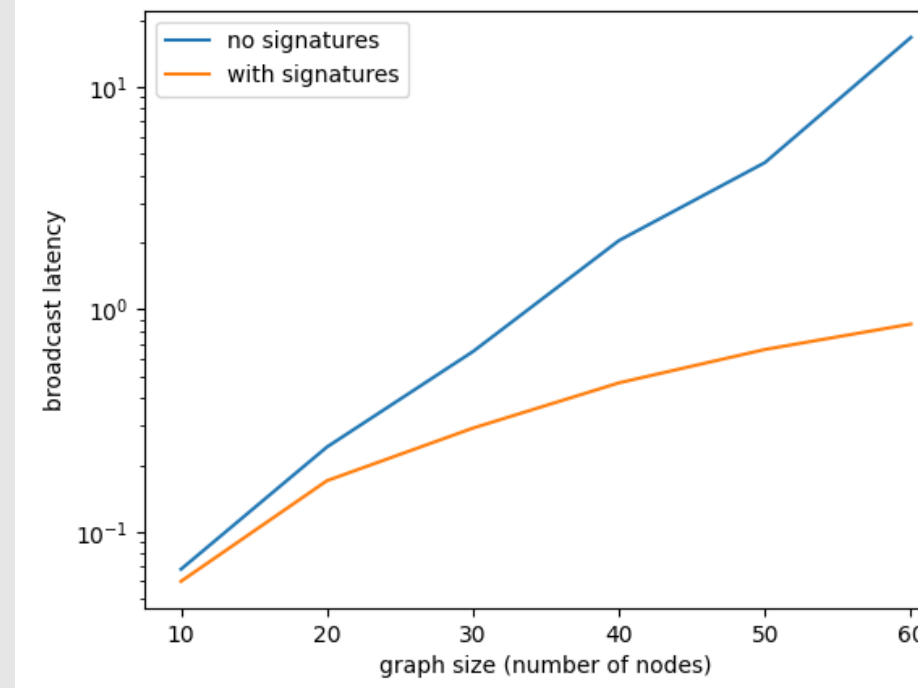
(2) Objective

Use Digital Signatures to reduce the number of messages sent by Bracha-Dolev

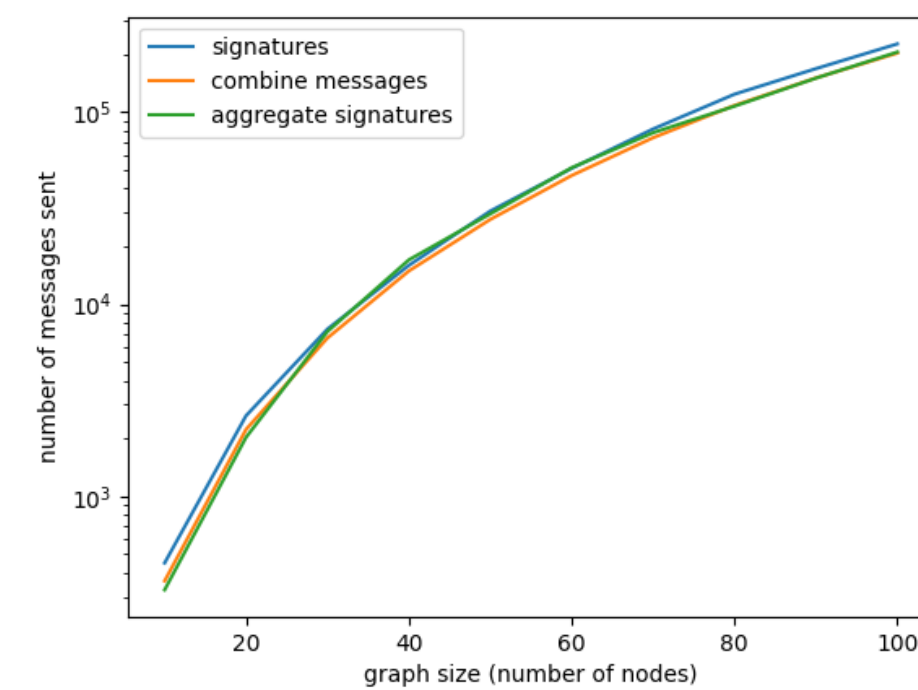
(5) Results



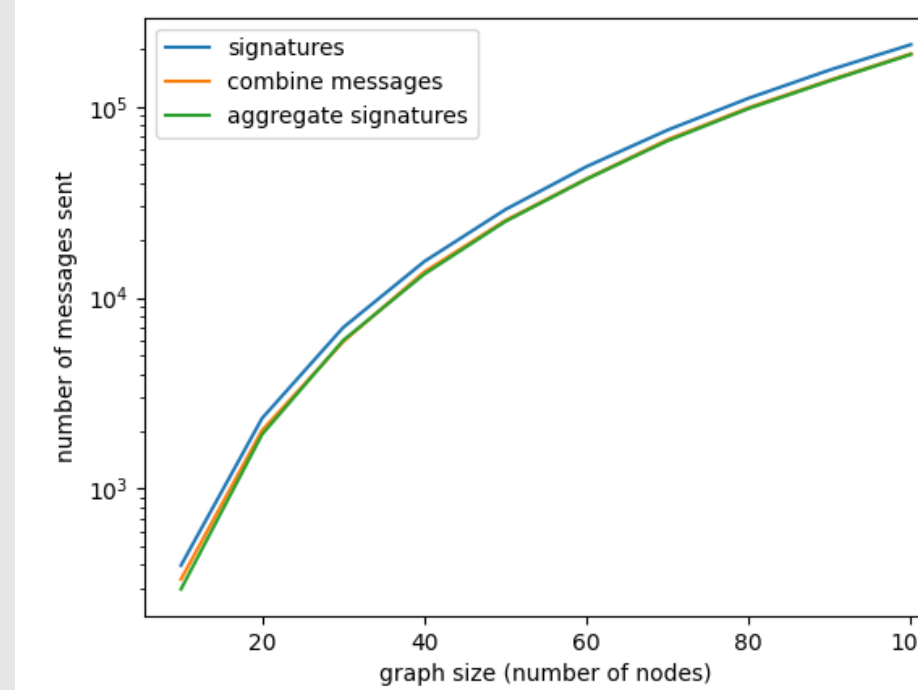
Comparison of message complexity between Bracha-Dolev with signatures and without



Comparison of broadcast latency between Bracha-Dolev with signatures and without



Comparison of message complexity between the different modifications, without the "stop after Bracha-deliver modification".



Comparison of message complexity between the different modifications, with the "stop after Bracha-deliver modification".

(3) Improvements made

1. **Signatures** – Use signatures instead of paths to validate messages.
2. **Combining messages** - When needing to broadcast one message and forward another in the Dolev layer, create a message with multiple signatures, each representing one message.
3. **Aggregate or Multi- signatures** – Combine messages, but use aggregate or multi signatures to have 1 or 2 signatures instead of n . Where n is the number of messages combined.
4. **Stop participating after Bracha-Deliver** – After a node has Bracha-Delivered, it can stop participating in the protocol.

(4) Evaluation setup

- The modifications were implemented in kotlin.
- Each node has multiple message handlers running concurrently to handle messages.
- When testing, multiple nodes are launched on the same computer.
- The different nodes communicate through queues

(6) Future Work

- Combine signatures, trusted execution environments and sharding to further increase performance.