Hotstuff Evaluation

Birju Patel

Project Goal

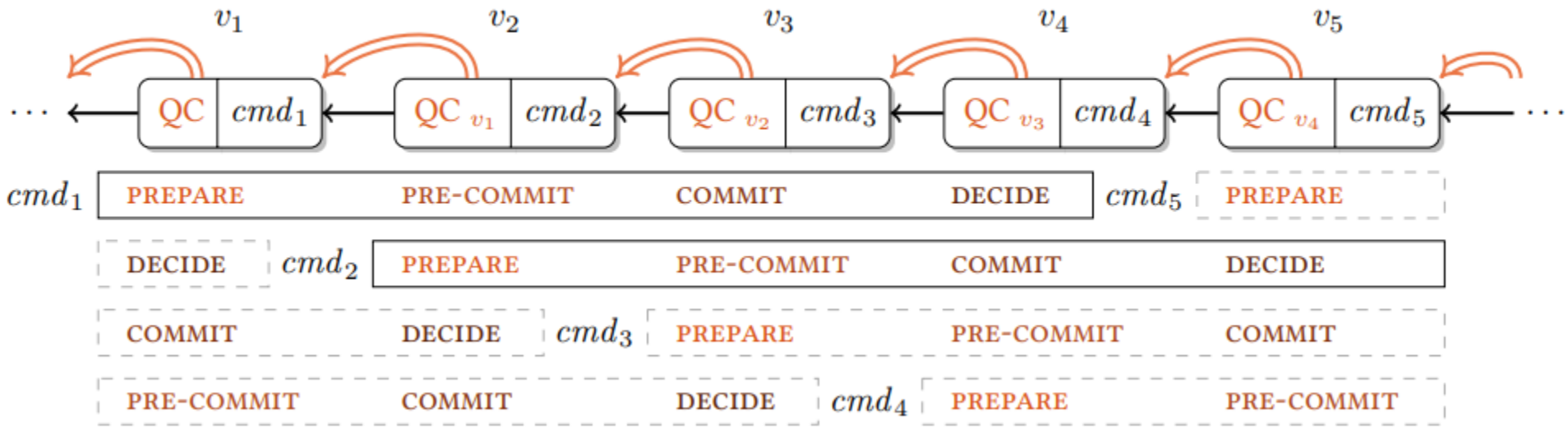
- Evaluate the performance of Hotstuff under attack
- Determine if a Hotstuff based BFT system was suitable for the SCADA application

Implementation

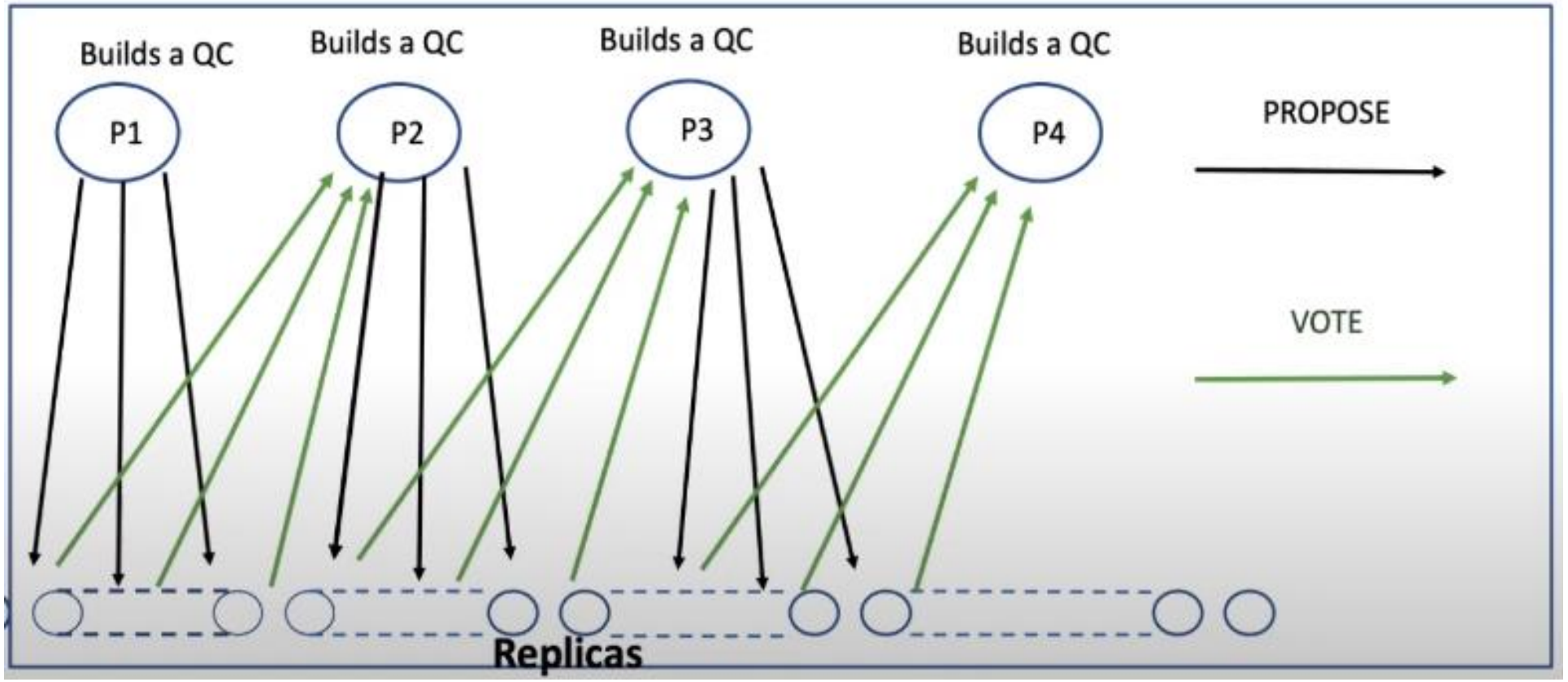
- I selected a BFT system that implemented the Hotstuff consensus protocol and used a mempool to disseminate transactions across the network
- https://github.com/asonnino/hotstuff

Chained Hotstuff

- Three phase commit
- Linear view change
- View change after each phase
- Blockchain based



Chained Hotstuff



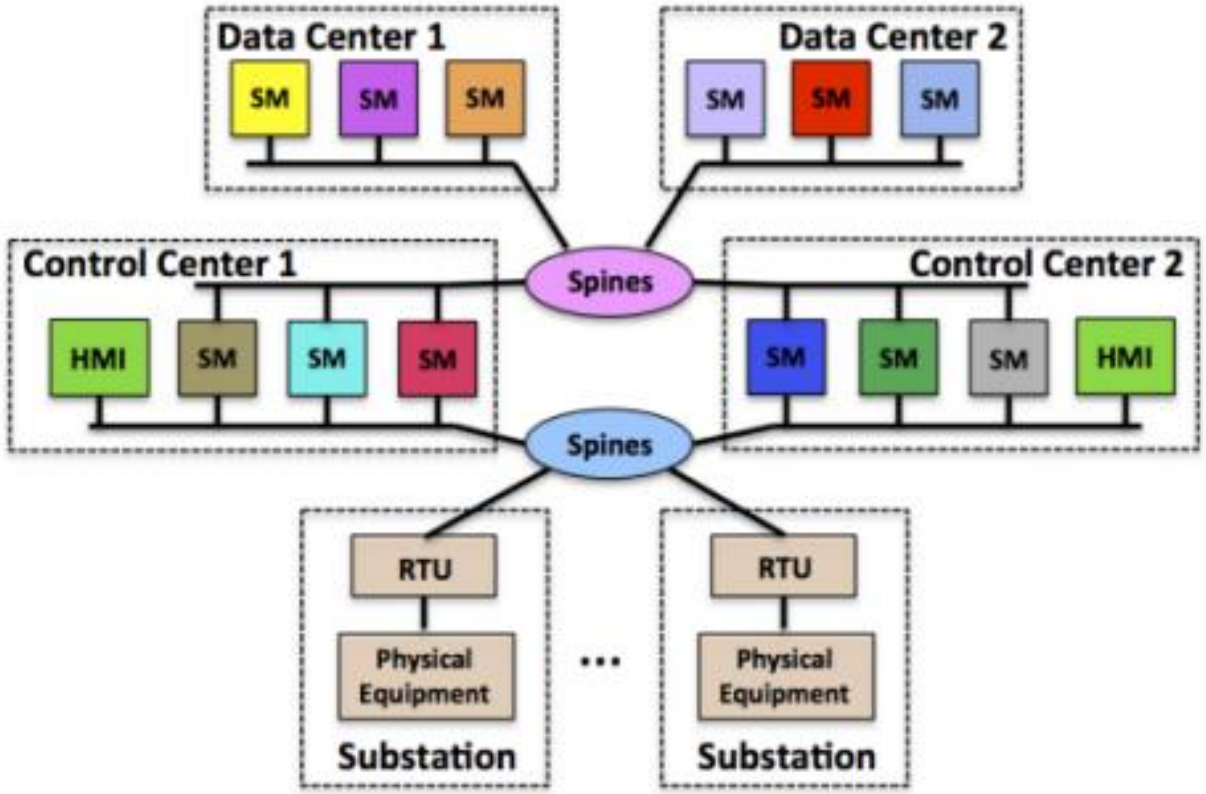
Mempool

- In traditional BFT systems, the transaction that the system is trying to reach consensus on is included in consensus messages
- The mempool separates BFT into two layers
  - Consensus – Deals with ordering blocks of transactions
  - Mempool – Deals with disseminating blocks of transactions across the network

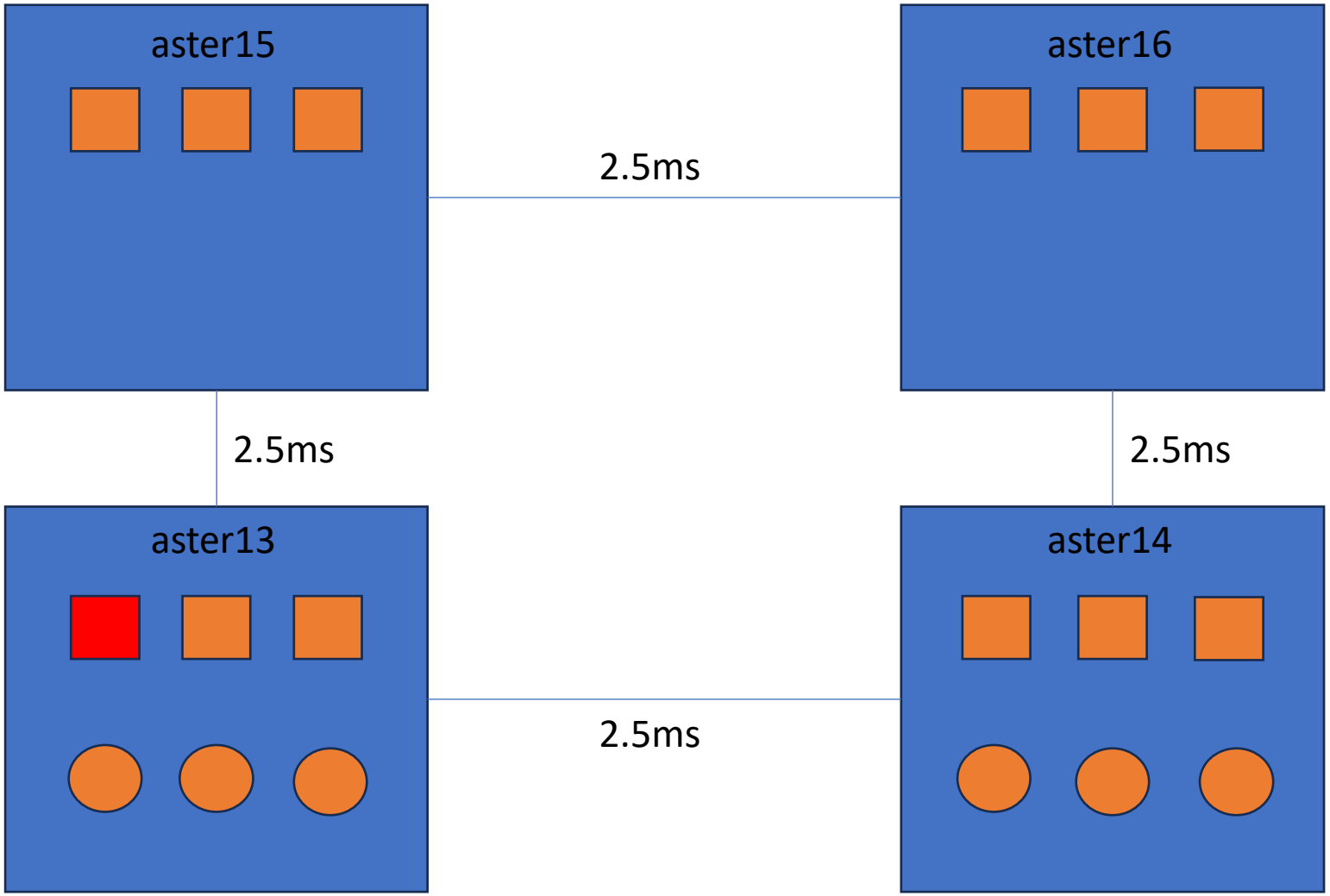
Mempool

- The mempool layer takes transactions from clients and packages them in blocks
- It then broadcasts those blocks in an unordered manner across the network
- A unique hash is created for each block
- This hash is then sent to the consensus layer, and the hash is included on consensus messages

Testing Setup



Testing Setup



Potential Threats

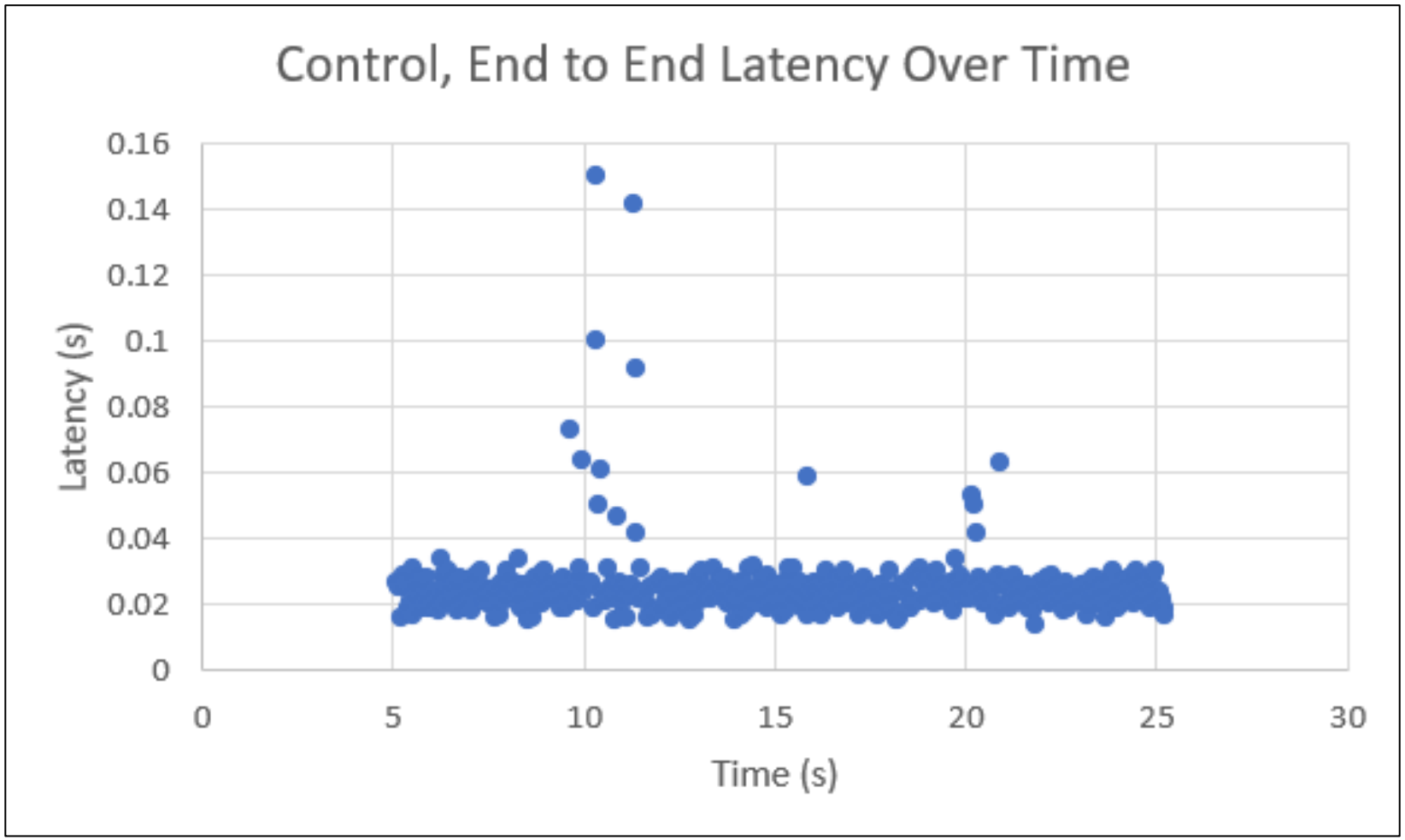
- Crash failures
- Network attacks
- Byzantine attacks

Test Scenarios

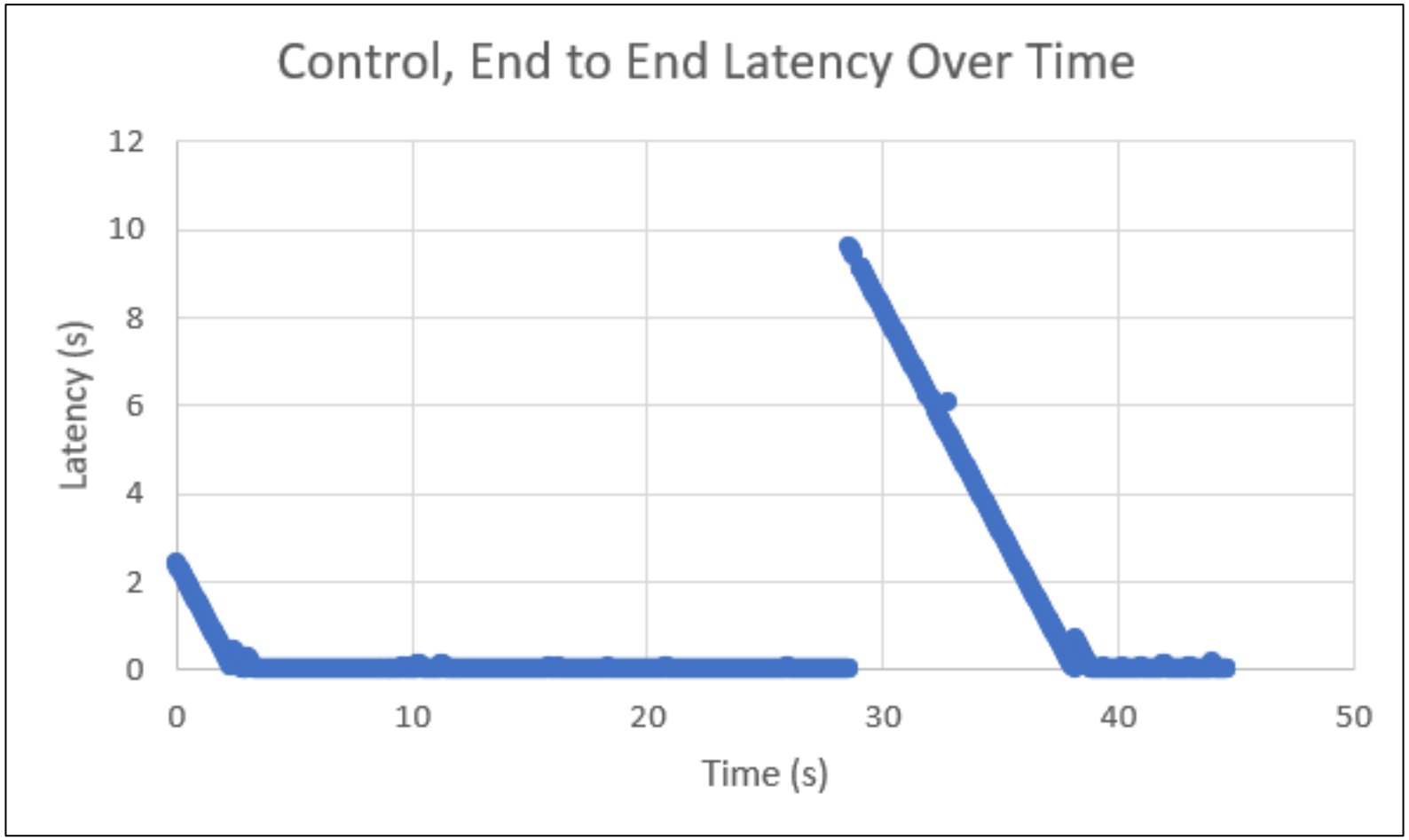
- Control - No interference
- Kill 1 – Kill and restart a single replica
- Site Isolation – Drop all communication to and from a single site, then reconnect the site
- Site Isolation + Kill 2 – Isolate a site and simultaneously kill 2 replicas, then reconnect the site and restart the replicas
- Delay Attack – A byzantine replica injects 500ms of delay when it becomes leader
- Fork Attack – A byzantine replica sends conflicting proposals when it becomes leader

Results

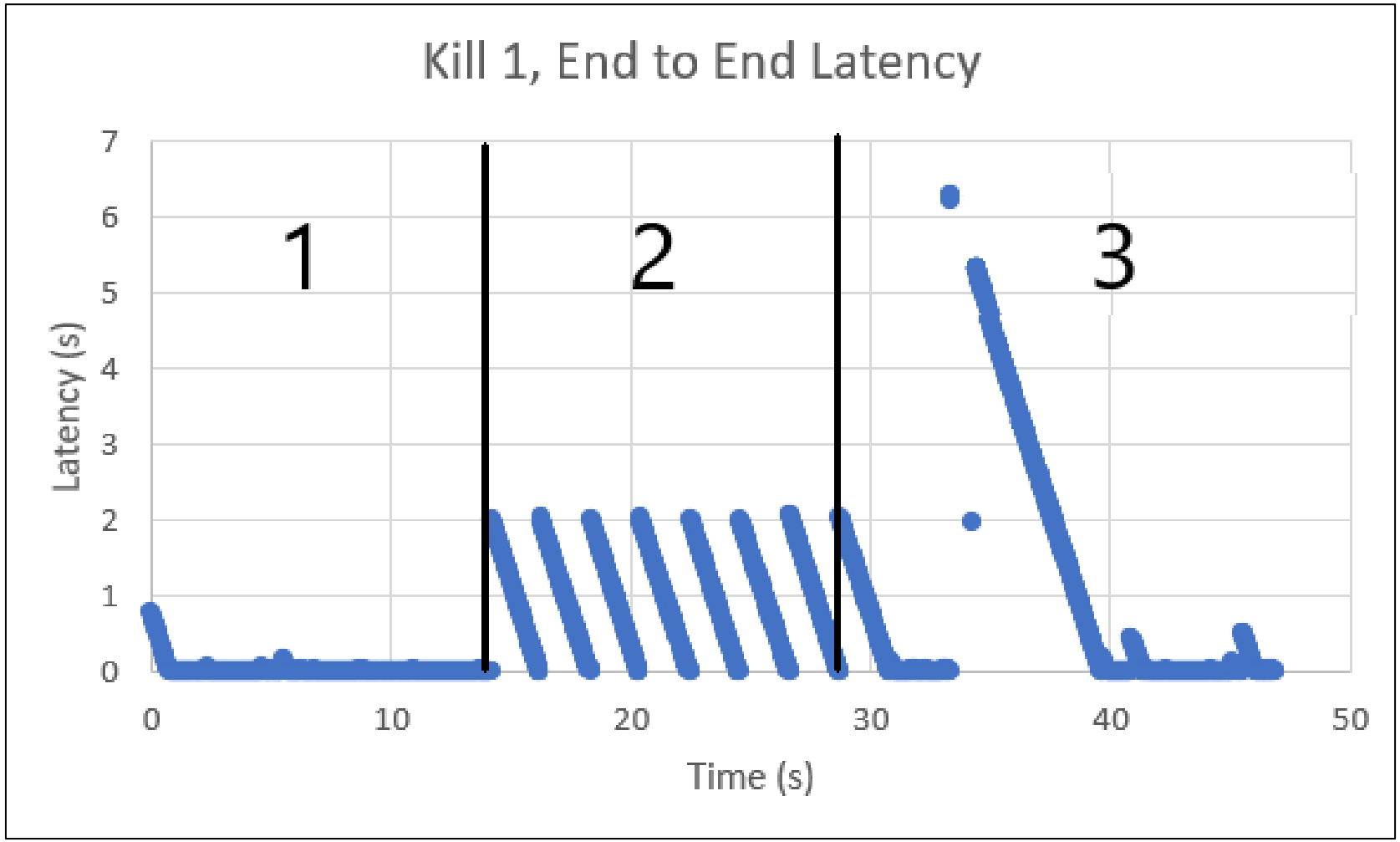
Control


Control, End to End Latency Over Time
Latency (s)
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25
30
Time (s)

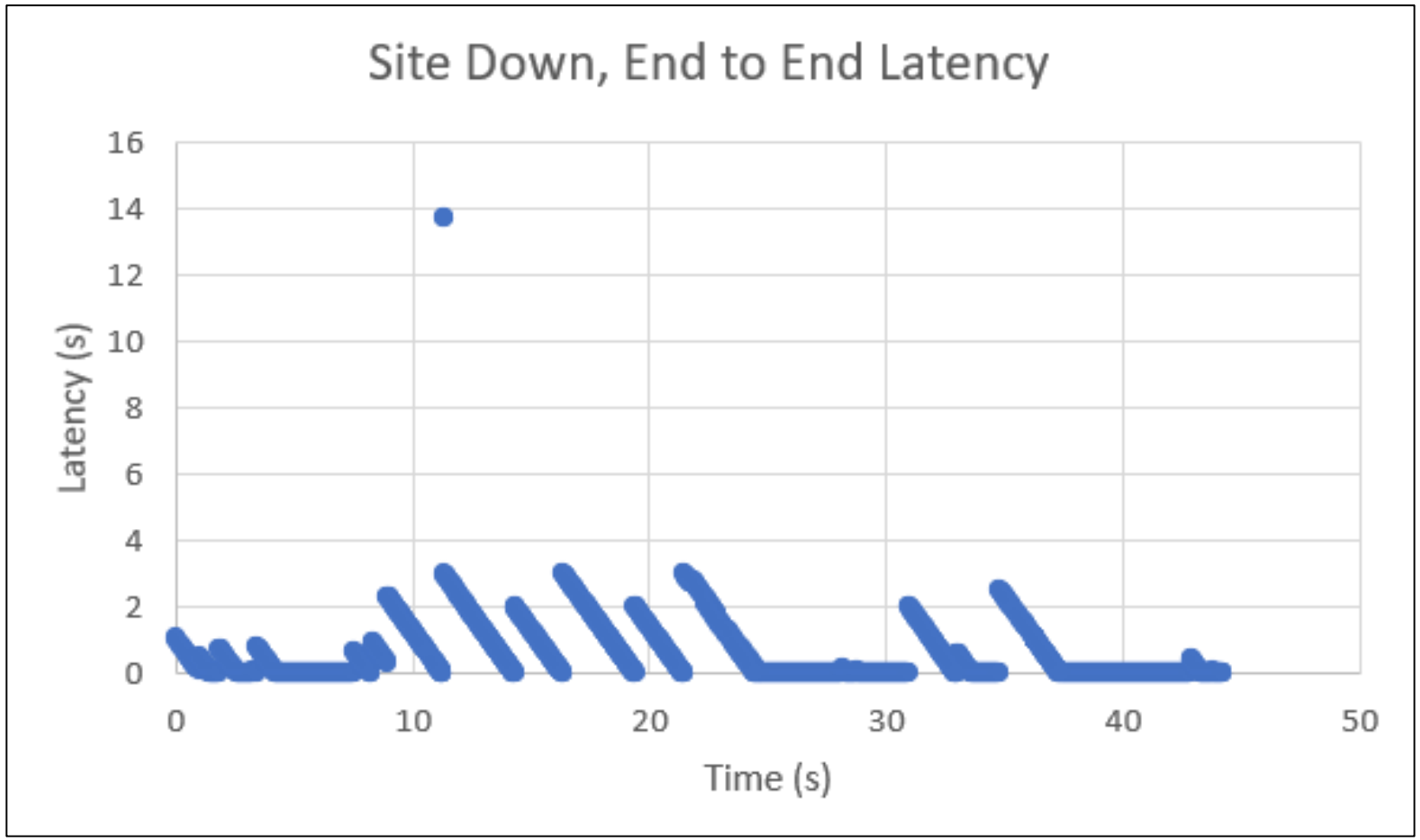
Control, Anomaly



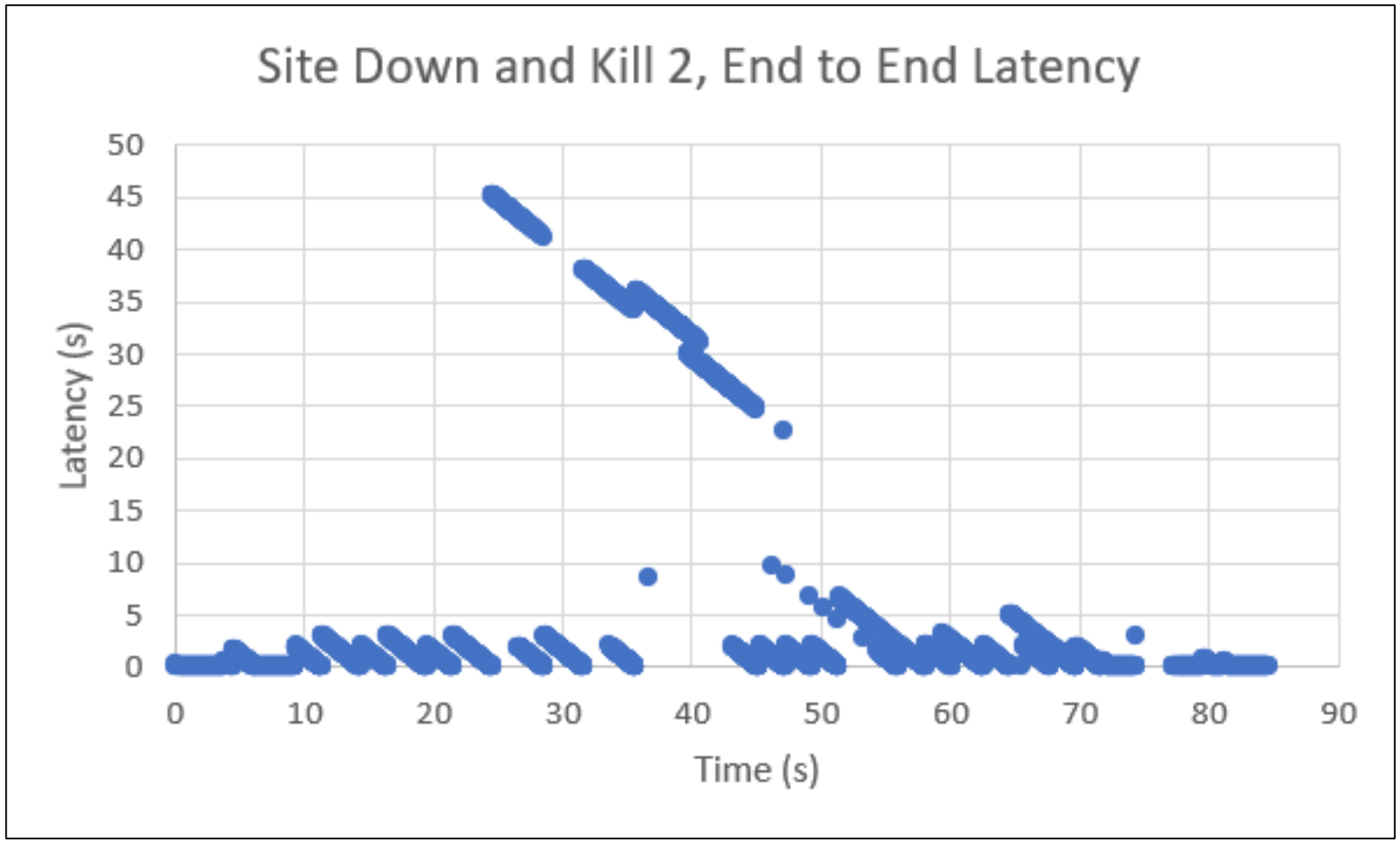
Kill 1


Kill 1, End to End Latency
Latency (s)
Time (s)
0
1
2
3
0
1
2
3
4
5
6
7
0
10
20
30
40
50

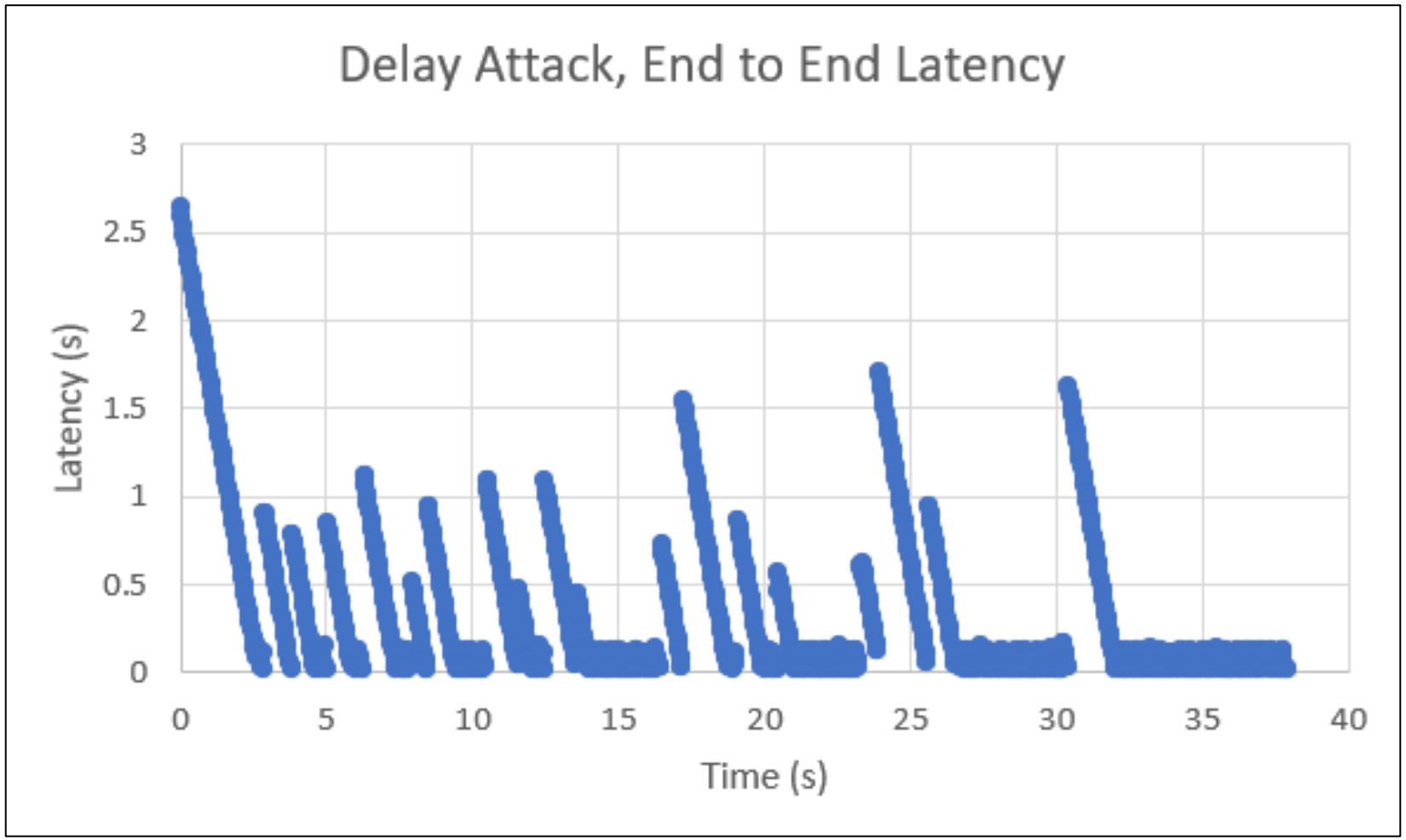
Site Isolation



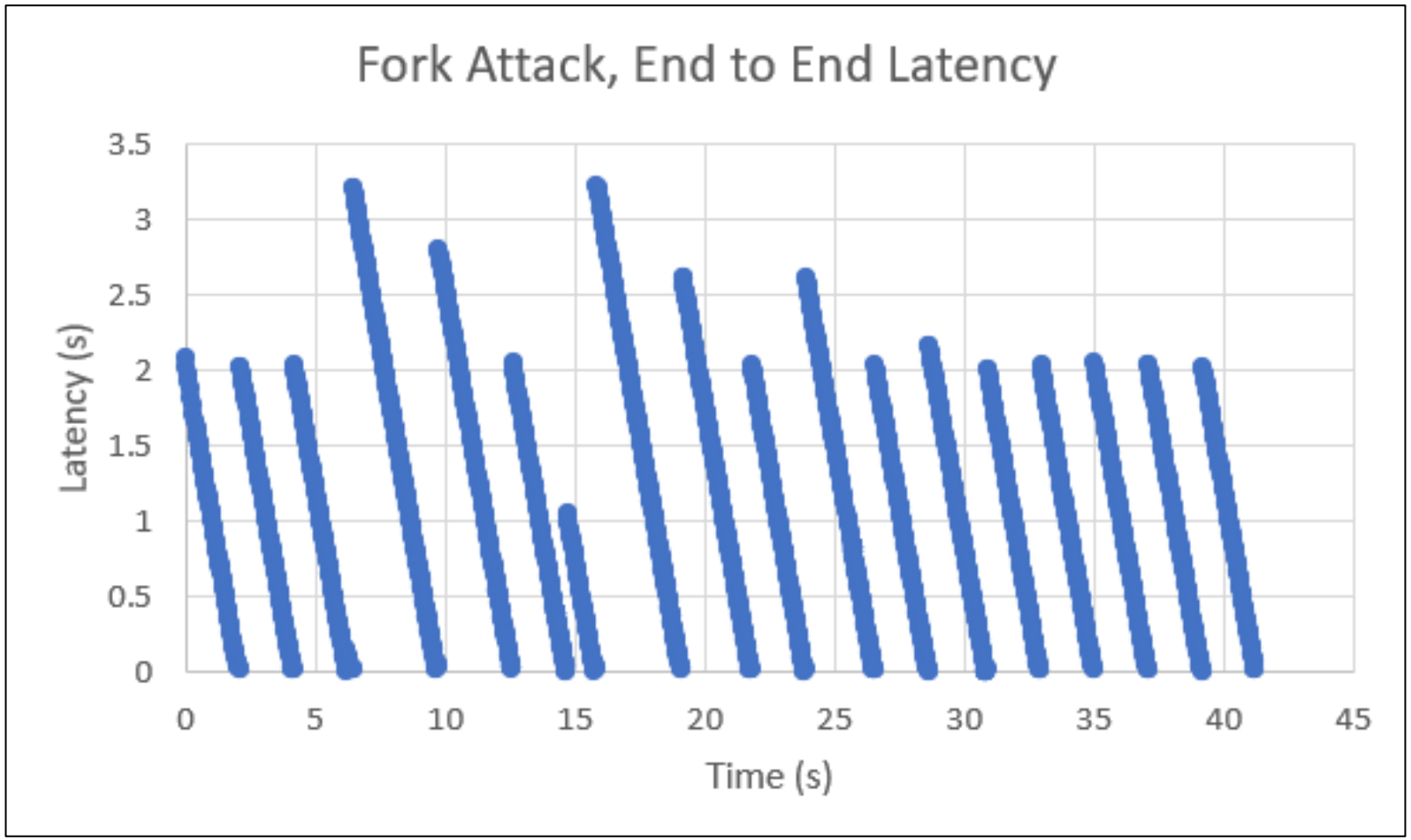
Site Isolation + Kill 2



Delay Attack



Fork Attack


Fork Attack, End to End Latency
Latency (s)
3.5
3
2.5
2
1.5
1
0.5
0
0
5
10
15
20
25
30
35
40
45
Time (s)