MultiPaxos vs Raft

Which is more predictable?

Chris Jensen, Heidi Howard, Richard Mortier

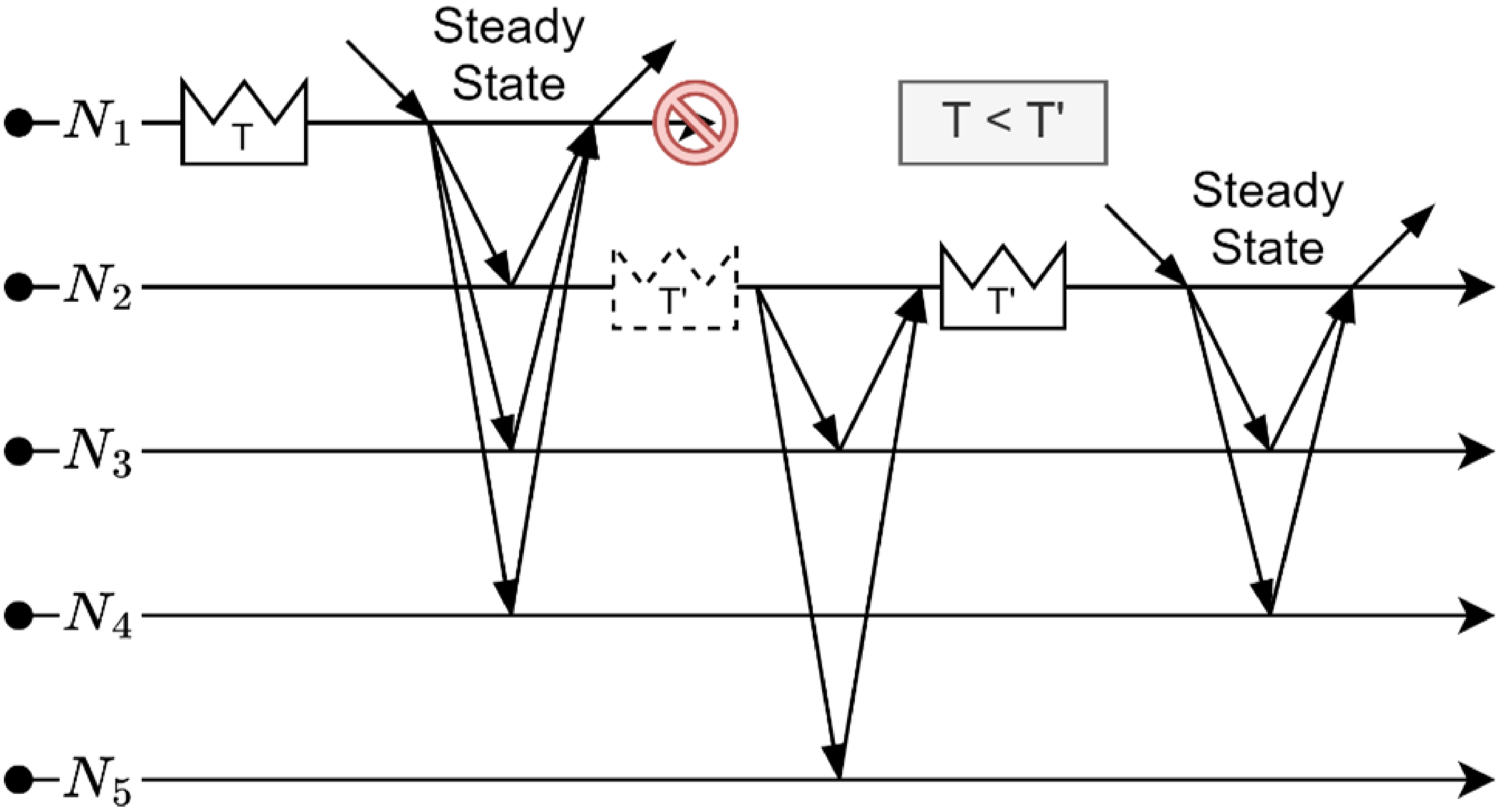
University of Cambridge

first.last@cl.cam.ac.uk

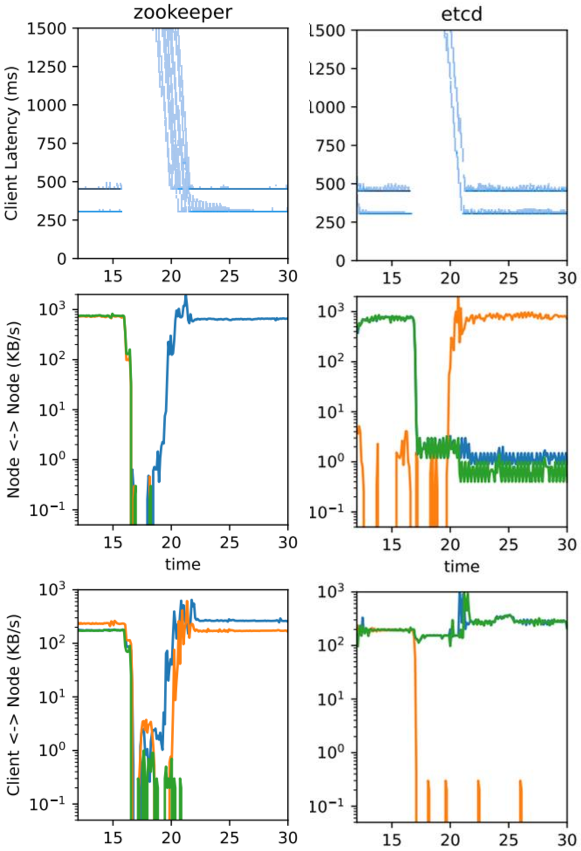
Consensus: the sync protocol for distributed datastores

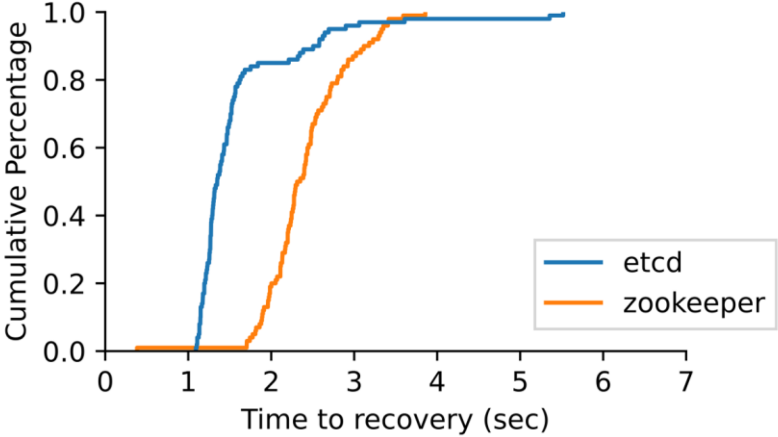
APACHE ZooKeeper™
Apache MESOS™
cassandra
etcd
Istio
kubernetes
CCF Confidential Consortium Framework
CockroachDB
Fly.io
stripe
Consul
CLOUDFLARE®

High level overview of these protocols



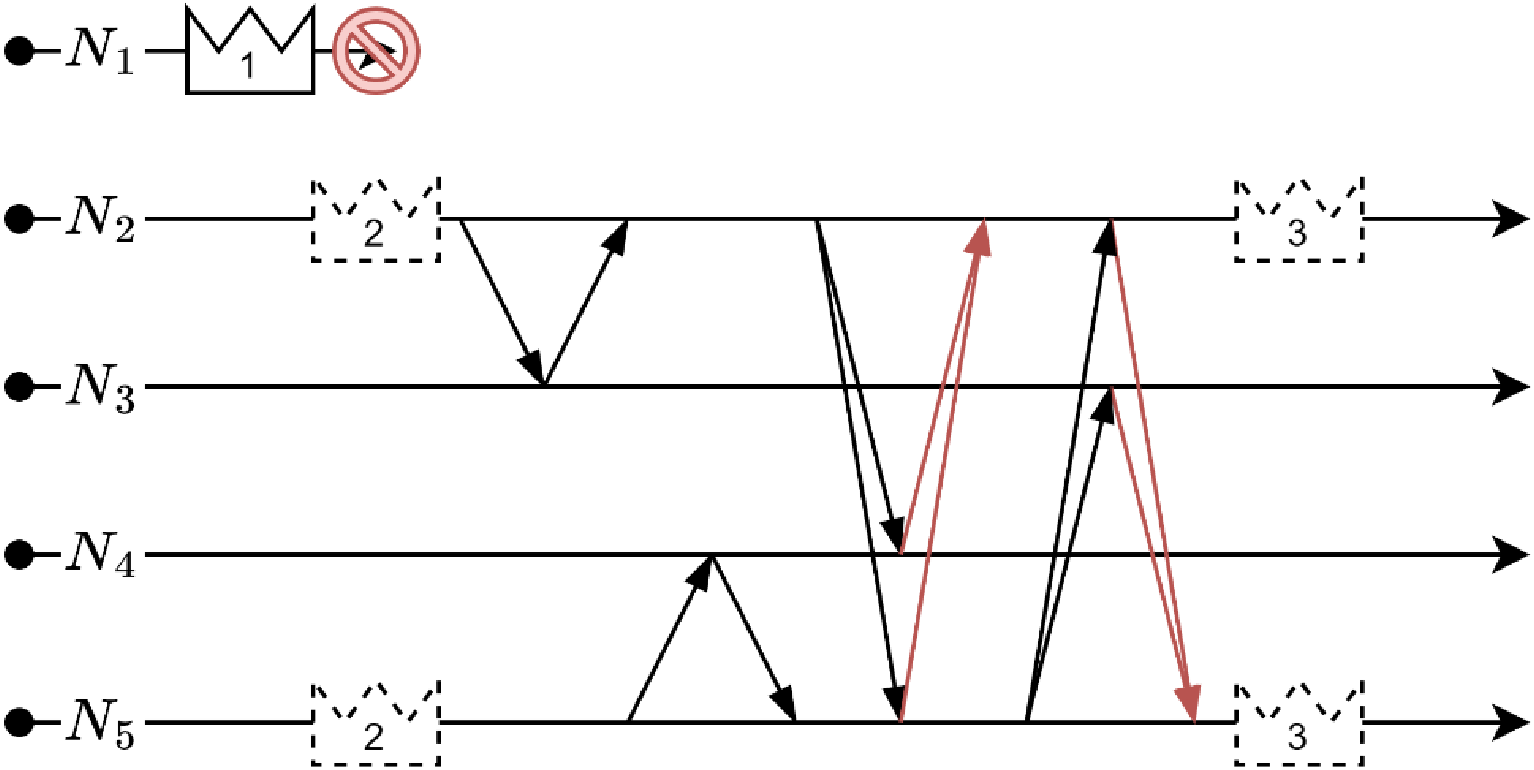
Leader failure is costly (tested with Reckon)





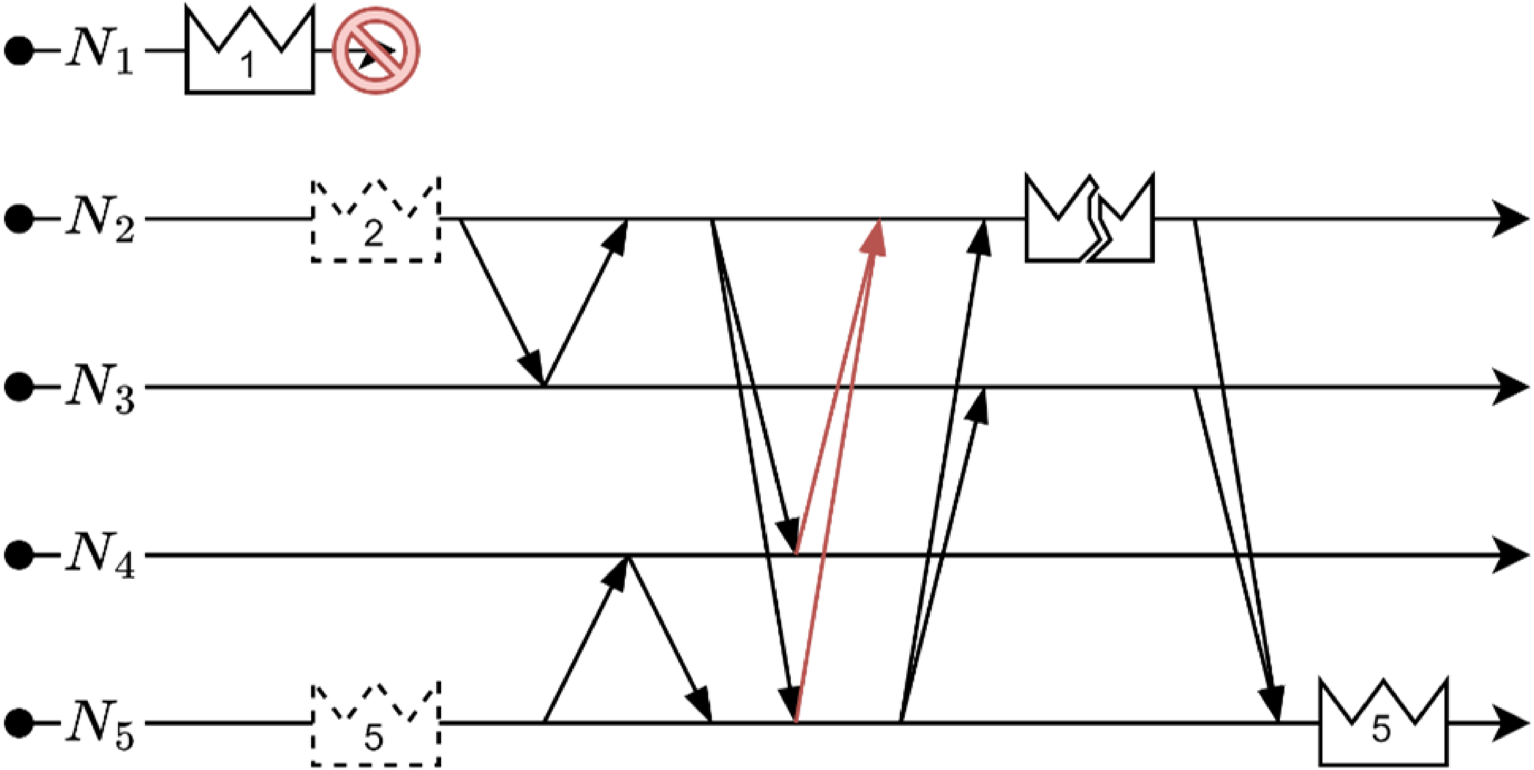
Submitted manuscript, benchmark: github.com/cjen1/reckon

etcd = Raft style election - majority vote decides leader

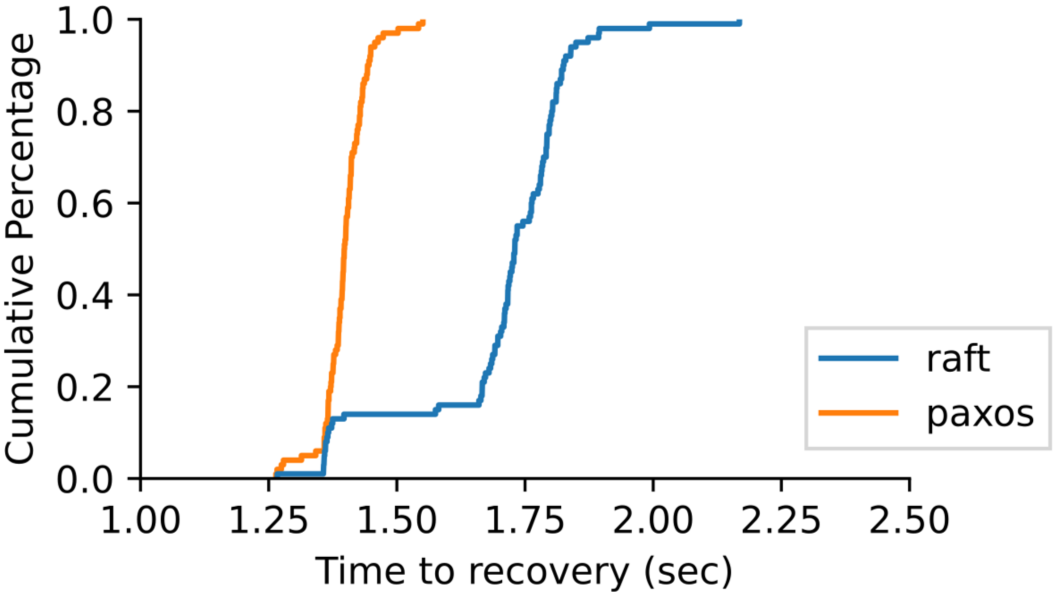


zookeeper ≈ Paxos - statically assign terms to nodes

the highest termed node is elected



Empirical testing Paxos and Raft using OCons



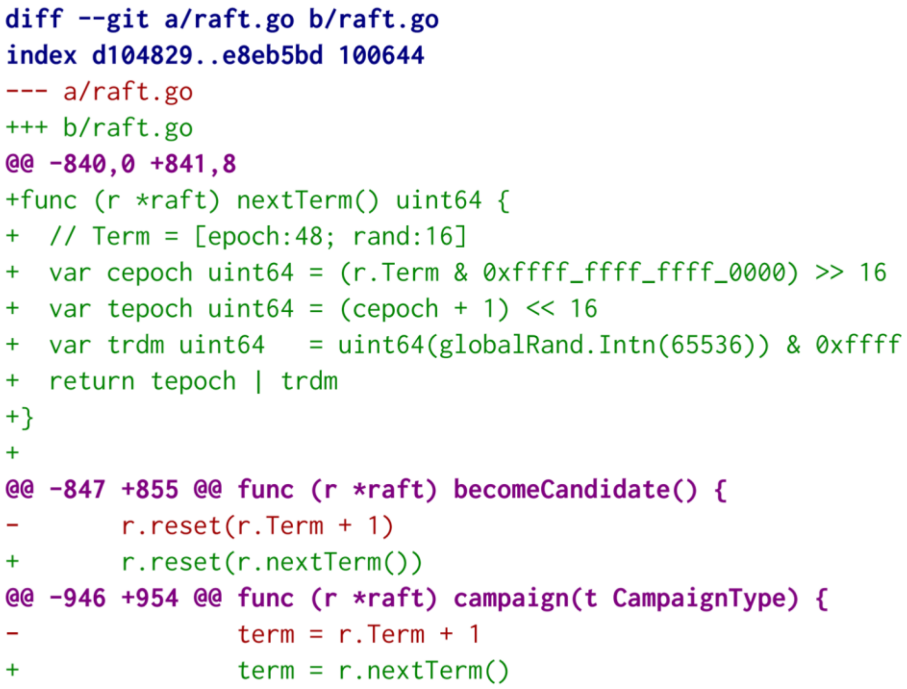
Paxos is more **predictable** than *Raft.*

But *Raft* is more **popular**.

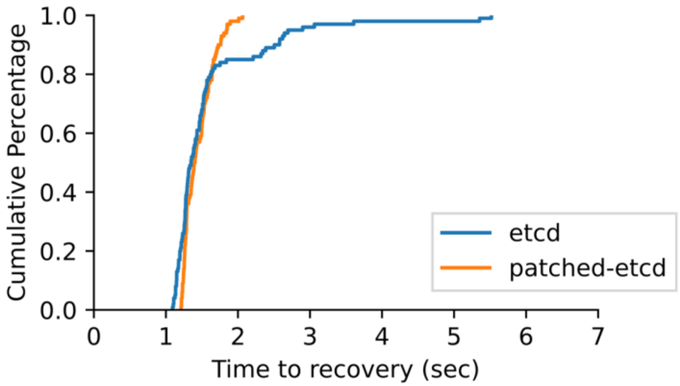
So can we make *Raft* **predictable**?

Benchmark: github.com/cjen1/reckon   Implementations: github.com/cjen1/ocons

Idea: randomise lower bits of the term, increment the rest

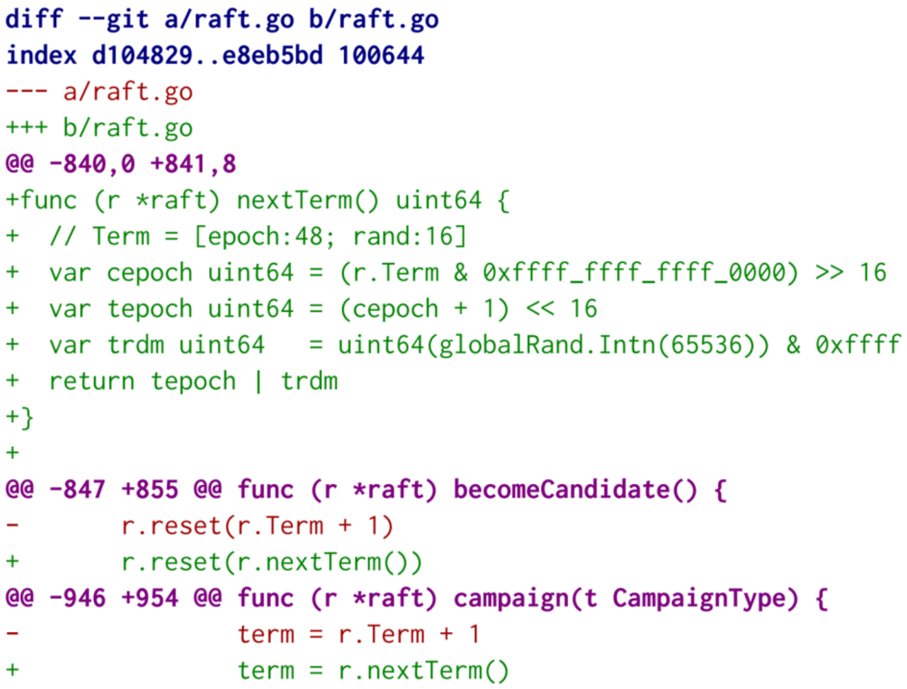


```
diff --git a/raft.go b/raft.go
index d104829..e8eb5bd 100644
--- a/raft.go
+++ b/raft.go
@@ -840,0 +841,8
+func (r *raft) nextTerm() uint64 {
+  // Term = [epoch:48; rand:16]
+  var cepoch uint64 = (r.Term & 0xffff_ffff_ffff_0000) >> 16
+  var tepoch uint64 = (cepoch + 1) << 16
+  var trdm uint64   = uint64(globalRand.Intn(65536)) & 0xffff
+  return tepoch | trdm
+}
+
@@ -847 +855 @@ func (r *raft) becomeCandidate() {
-       r.reset(r.Term + 1)
+       r.reset(r.nextTerm())
@@ -946 +954 @@ func (r *raft) campaign(t CampaignType) {
-               term = r.Term + 1
+               term = r.nextTerm()
```

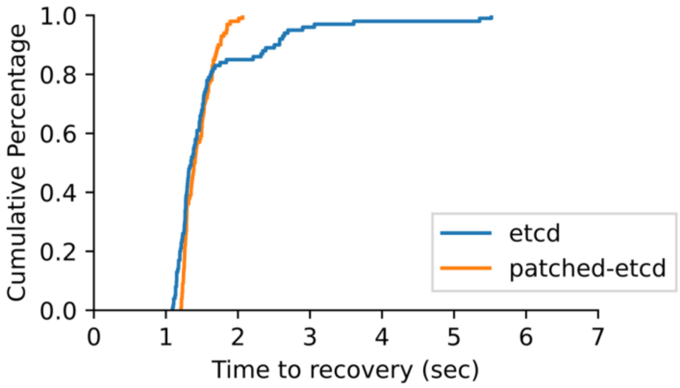


Applicable to most *Raft* implementations

Thanks for listening!



```
diff --git a/raft.go b/raft.go
index d104829..e8eb5bd 100644
--- a/raft.go
+++ b/raft.go
@@ -840,0 +841,8
+func (r *raft) nextTerm() uint64 {
+  // Term = [epoch:48; rand:16]
+  var cepoch uint64 = (r.Term & 0xffff_ffff_ffff_0000) >> 16
+  var tepoch uint64 = (cepoch + 1) << 16
+  var trdm uint64   = uint64(globalRand.Intn(65536)) & 0xffff
+  return tepoch | trdm
+}
+
@@ -847 +855 @@ func (r *raft) becomeCandidate() {
-       r.reset(r.Term + 1)
+       r.reset(r.nextTerm())
@@ -946 +954 @@ func (r *raft) campaign(t CampaignType) {
-               term = r.Term + 1
+               term = r.nextTerm()
```



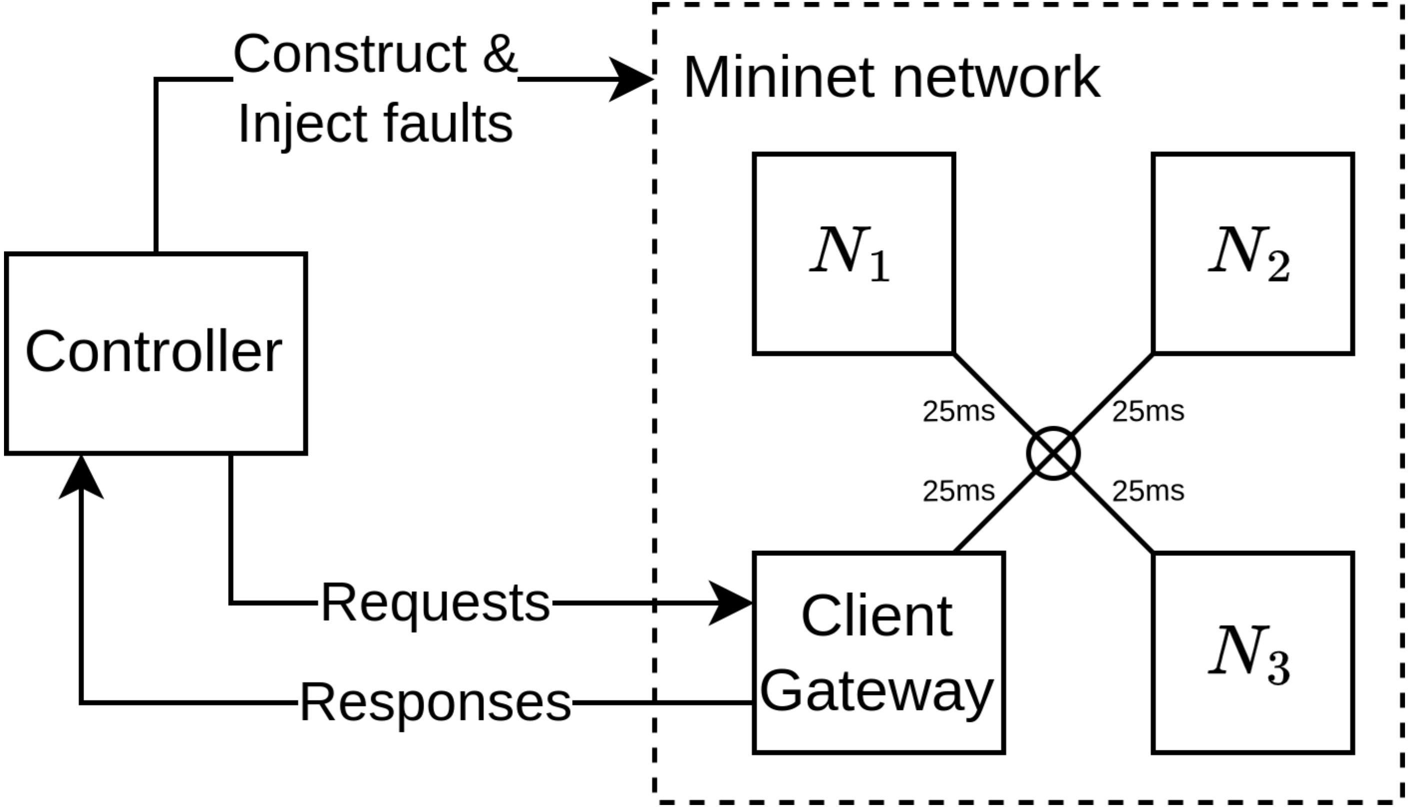
chris.jensen@cl.cam.ac.uk

@Cjen1@discuss.systems

github.com/Cjen1

Additional Slides

Reckon network


Construct &
Inject faults
Mininet network
Controller
N_1
N_2
25ms
25ms
25ms
25ms
Requests
Responses
Client
Gateway
N_3