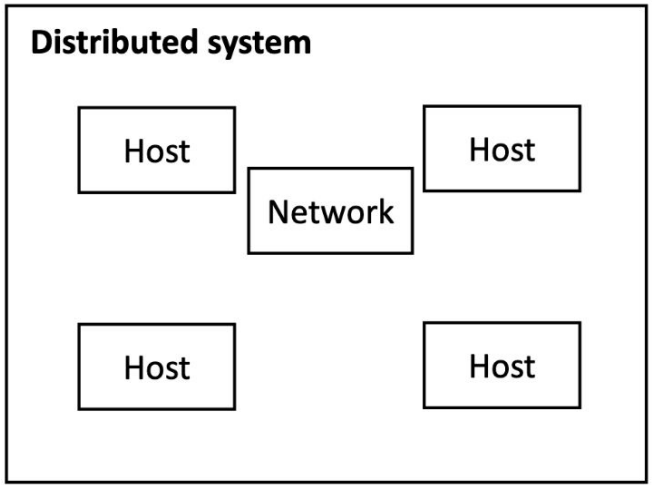
MichiganEngineering

EECS 498 - 003 Lab

Keshav Singh

Lab #5

A distributed system is composed of multiple hosts **and a network**



Distributed system: attempt #2

```
module DistributedSystem {
  datatype Variables =
    Variables(hosts:seq<Host.Variables>,
              network: Network.Variables)

  predicate HostAction(v, v', hostid, msgOps) {
    && Host.Next(v.hosts[hostid],v'.hosts[hostid],msgOps))
    && forall otherHost:nat | otherHost != hostid ::
        v'.hosts[otherHost] == v.hosts[otherHost]
  }

  predicate Next(v, v', hostid, msgOps: MessageOps) {
    && HostAction(v, v', hostid, msgOps)
    && Network.Next(v, v', msgOps)
  }
}
```

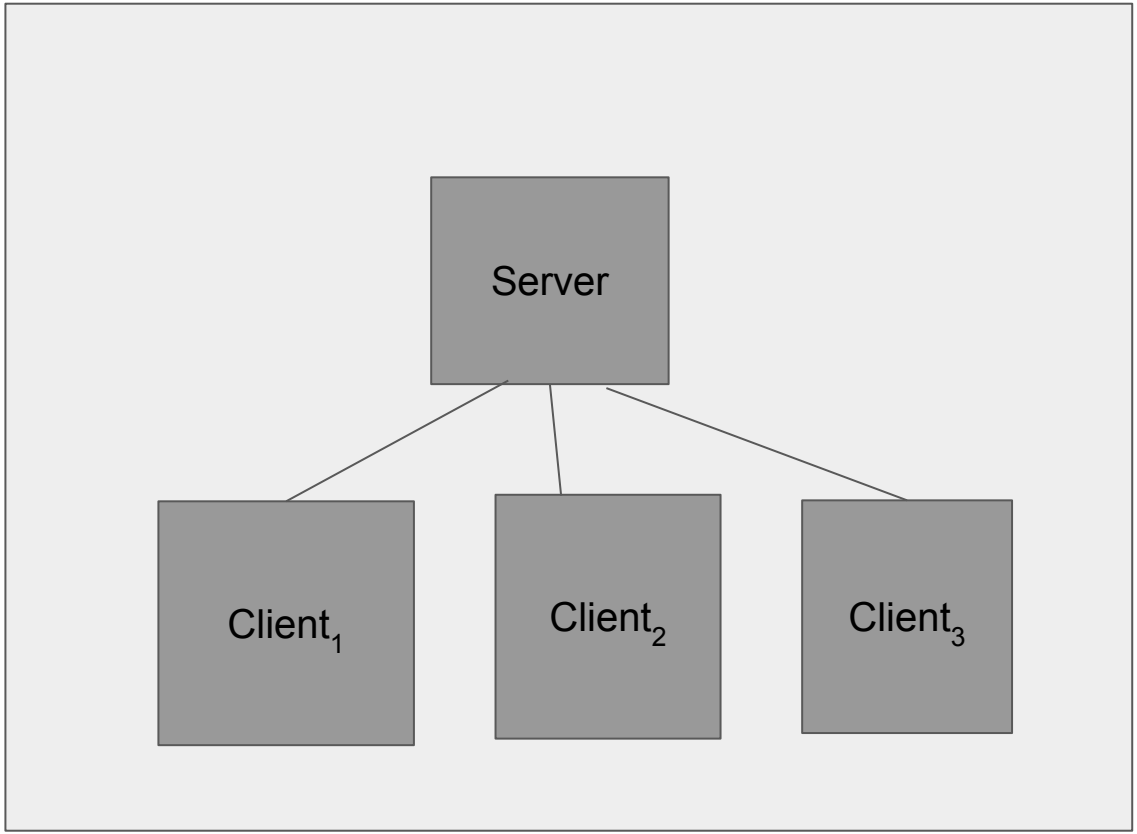
Binding variable

Network model

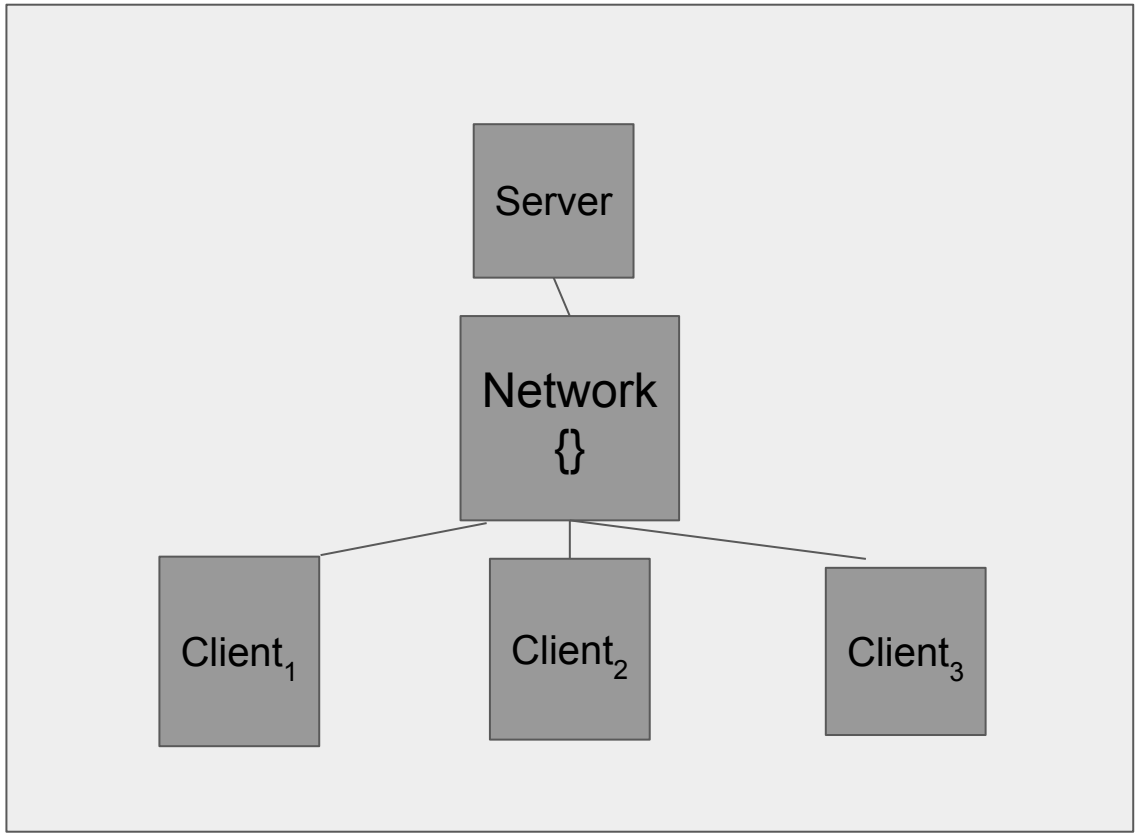
- Many choices for modelling networks, in this class we will (mostly) stick to the bag of messages model

```
predicate Next(v, v', msgOps:MessageOps) {
  // can only receive messages that have been sent
  && (msgOps.recv.Some? ==> msgOps.recv.value in v.sentMsgs)
  // Record the sent message, if there was one
  && v'.sentMsgs ==
     v.sentMsgs + if msgOps.send.None? then {}
                  else {msgOps.send.value}
}
}
```

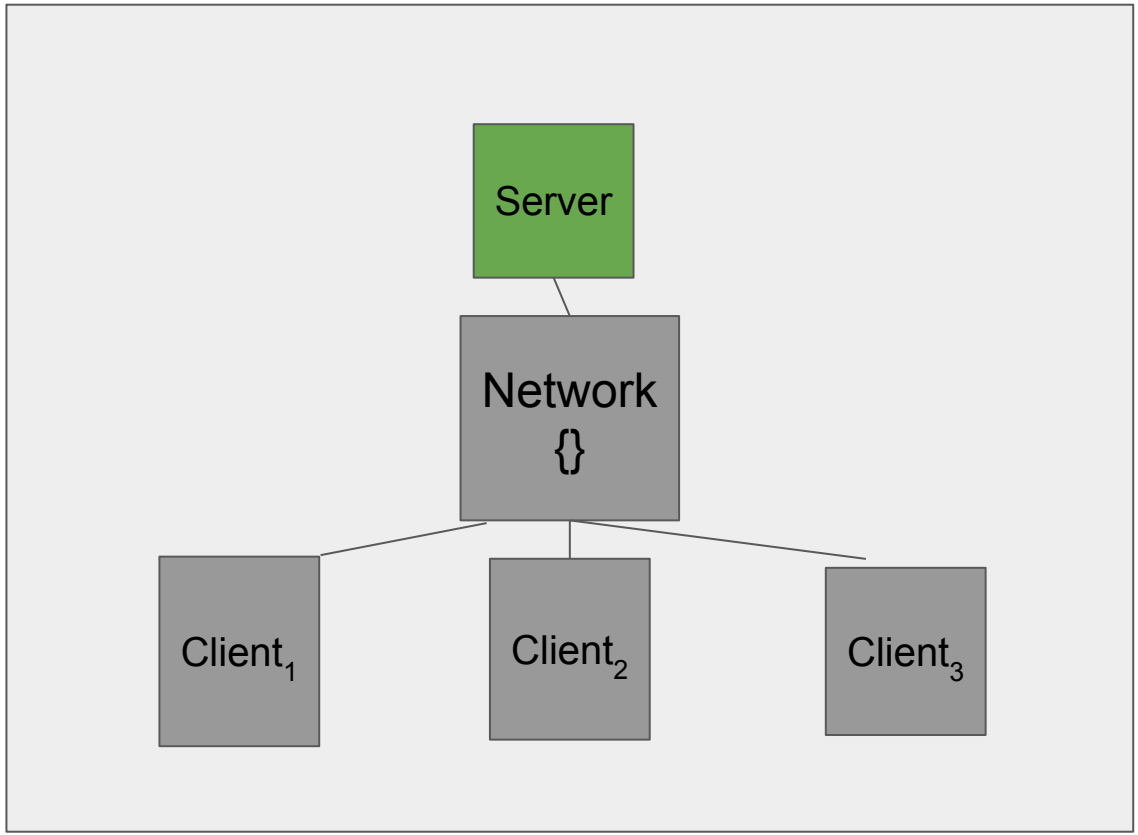
Lock Service From Homework



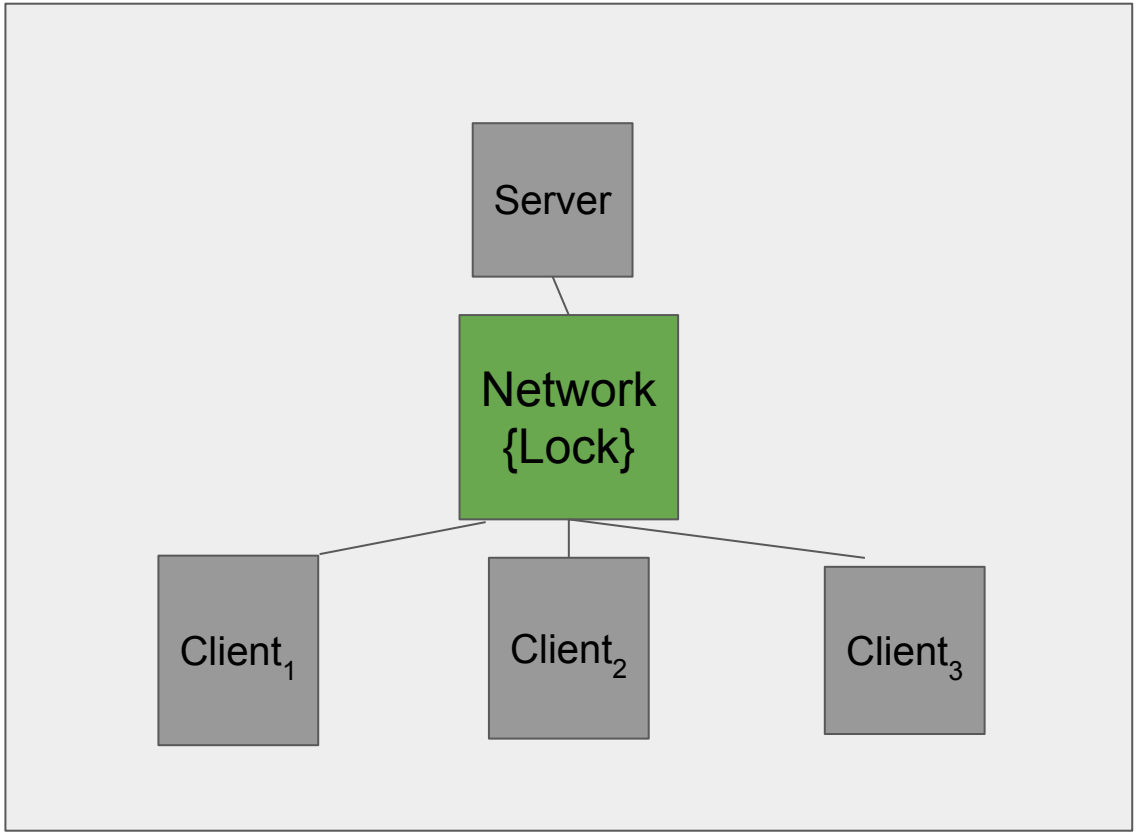
Lock Service With Network



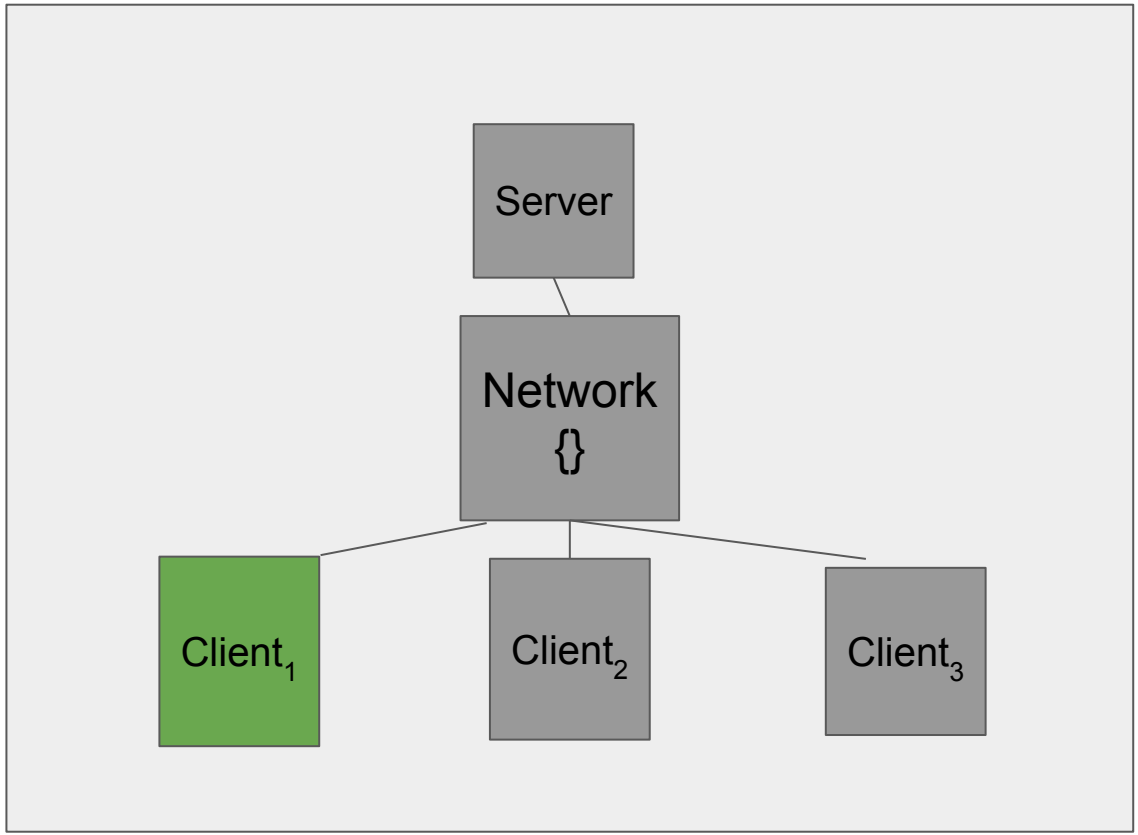
Lock Service With Network: Init



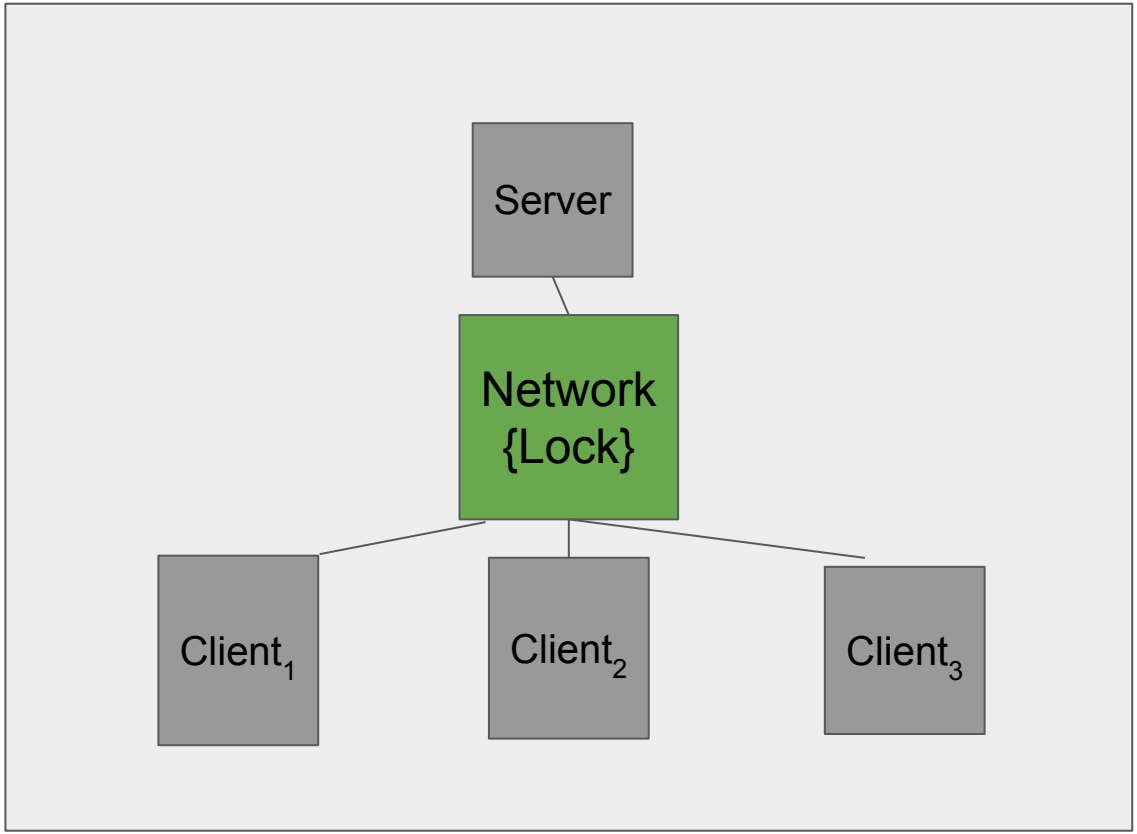
Lock Service With Network: Server Release



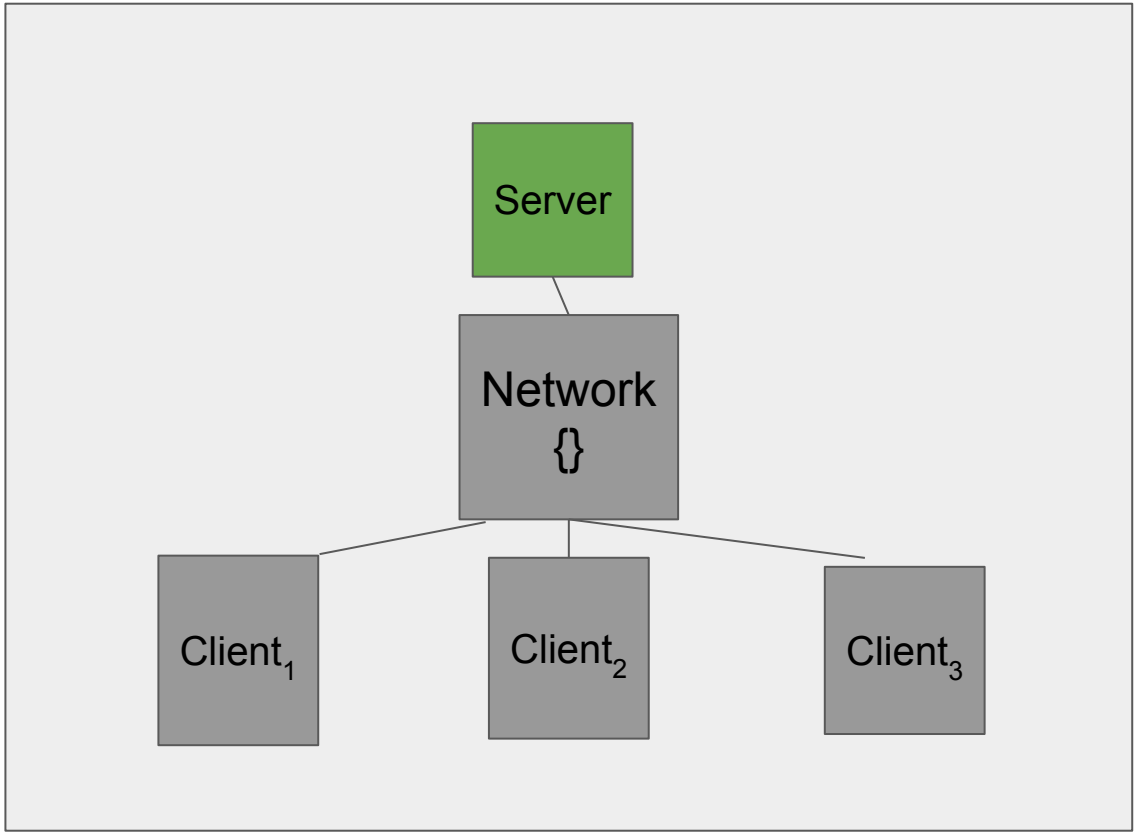
Lock Service With Network: Client Acquire



Lock Service With Network: Client Release



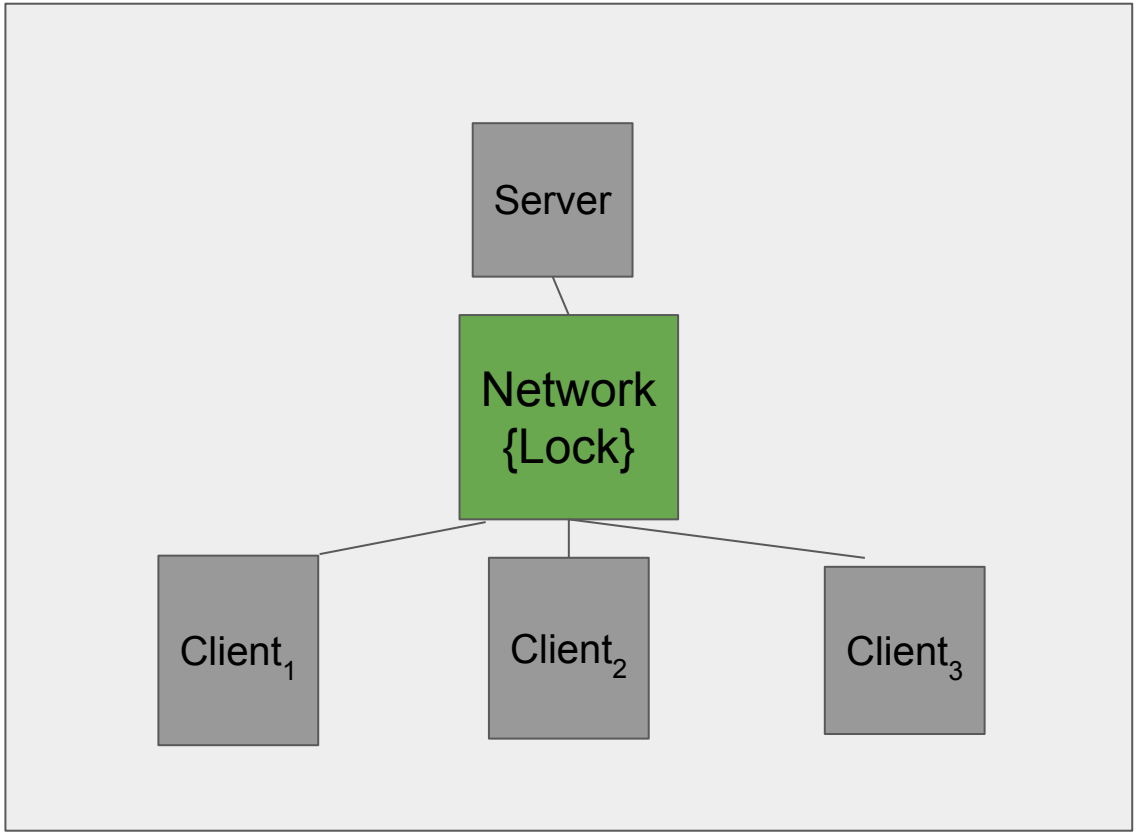
Lock Service With Network: Server Acquire



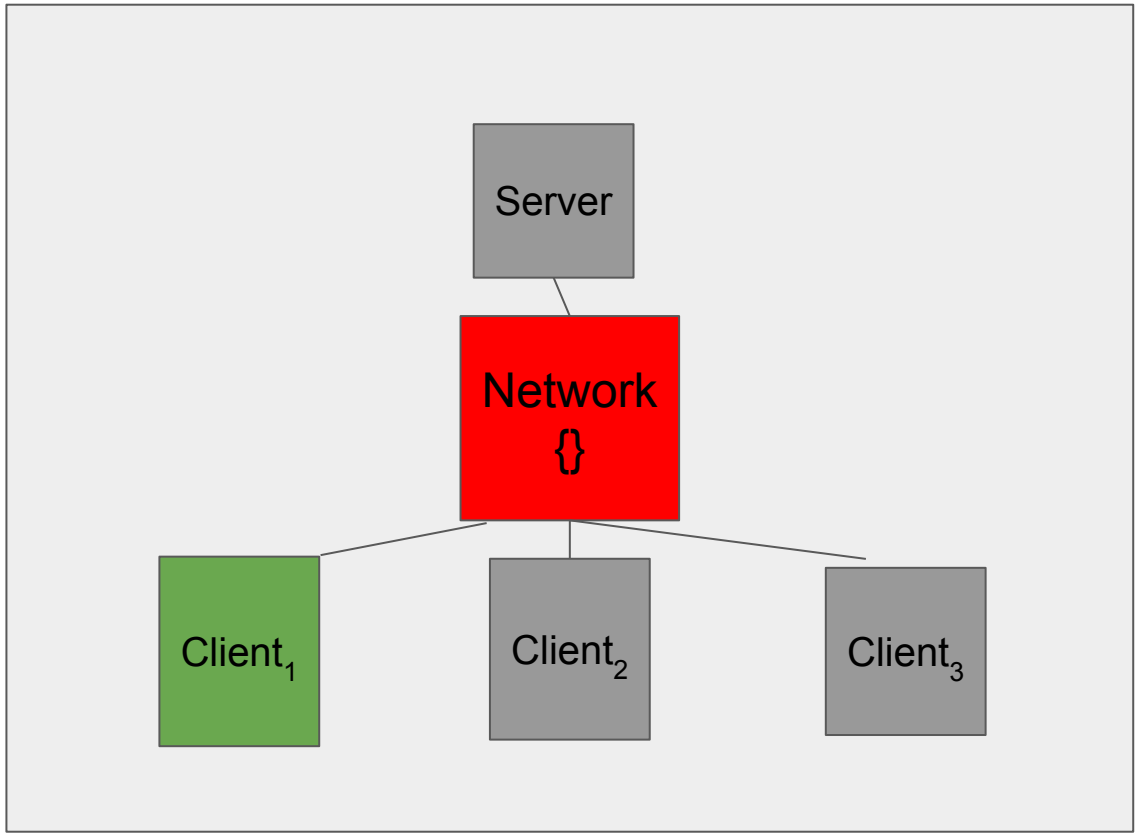
Was this behavior accurate?

- Discuss with your neighbor whether this behavior is accurate given the assumptions of distributed systems we discussed in class

Lock Service With Network



Lock Service With Network: Client Acquire



Guaranteeing Safety

- Q: If messages can be delivered twice is safety guaranteed in previous protocol?

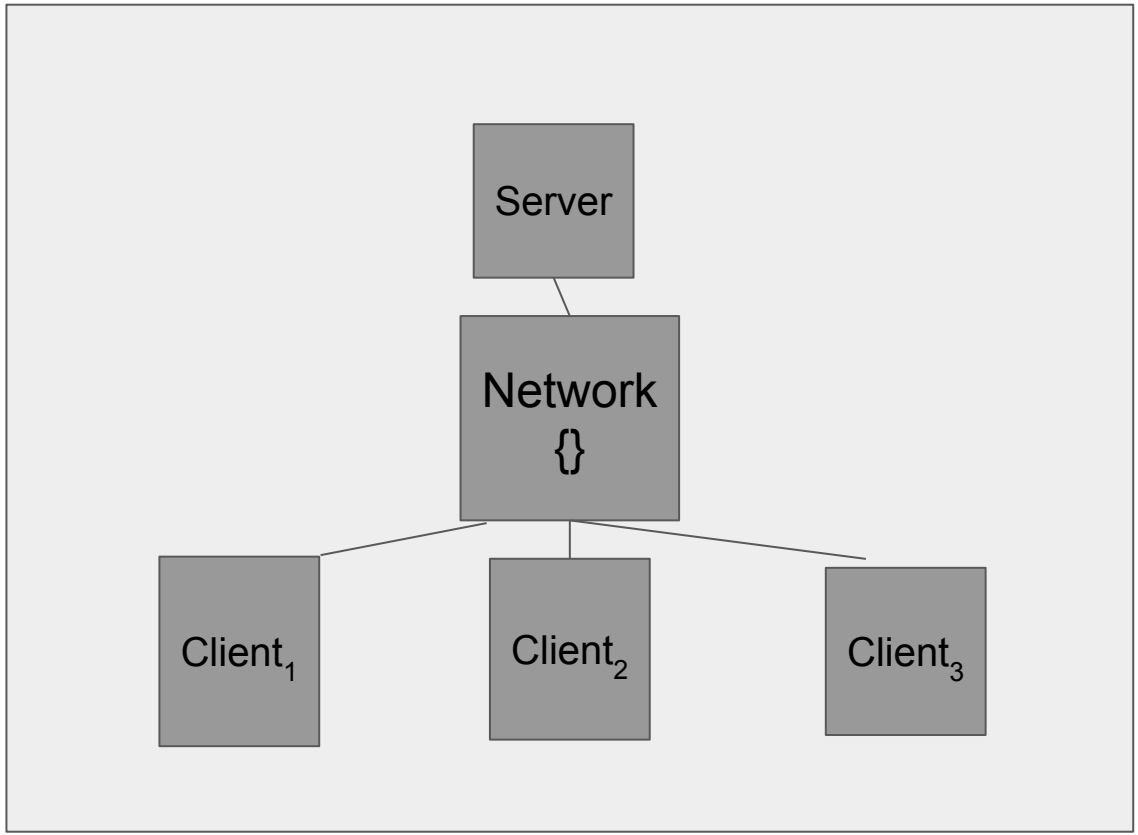
Guaranteeing Safety

- Q: If messages can be delivered twice is safety guaranteed in previous protocol?
- A: Clients/Server may receive a stale lock. How can we avoid this from happening?

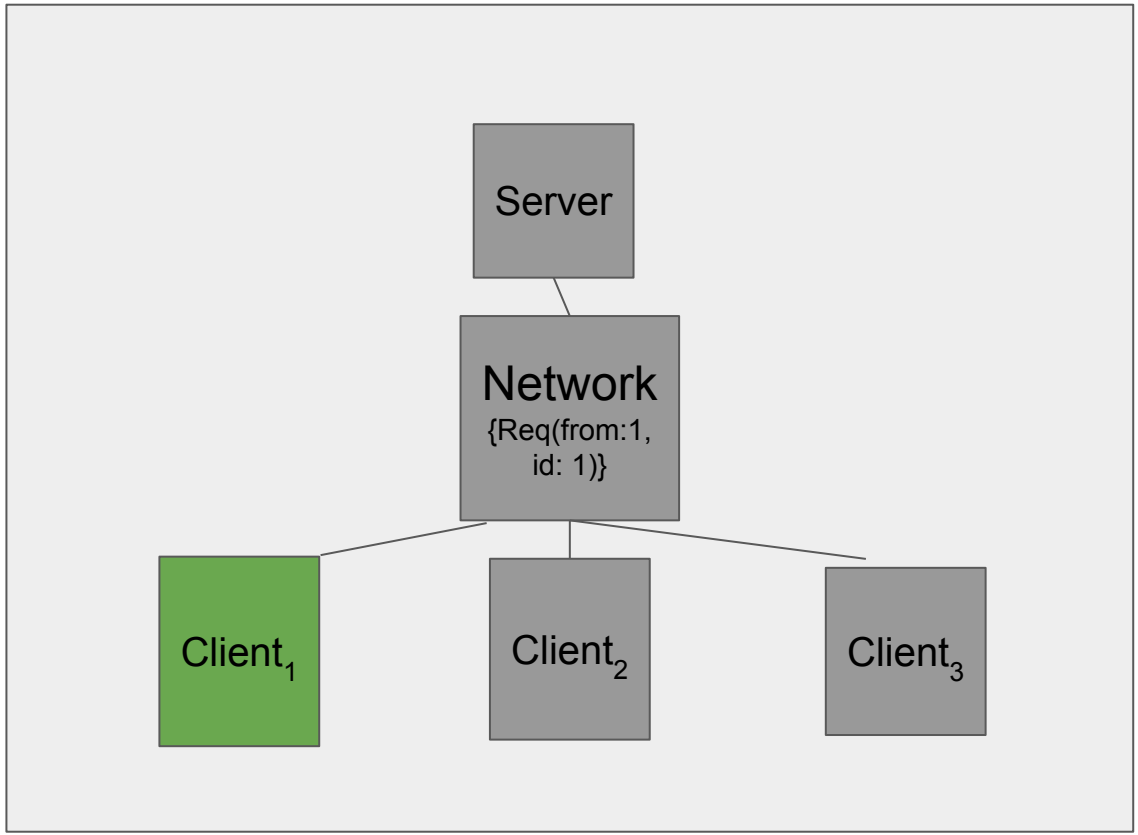
Version Numbers

- Need to have a version number (epoch) for the lock
- Server/Client can only receive lock if the epoch number is high enough

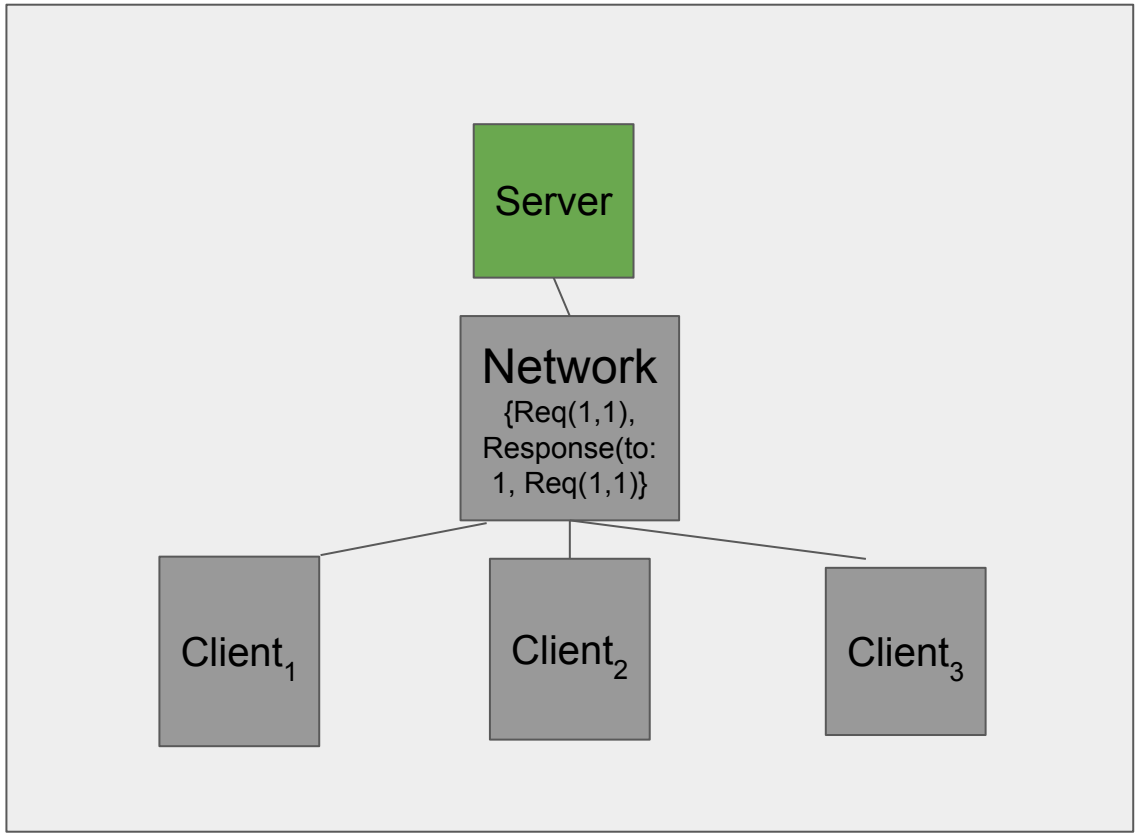
Echo Server



Echo Server



Echo Server


Server
Network
{Req(1,1),
Response(to:
1, Req(1,1)}
Client1
Client2
Client3

Echo Server

- Safety: we only receive messages from the server that we sent.
- We shall write the protocol steps for the Echo Server and prove the above safety property using inductive invariants
- Certain invariants/patterns can be generalized/applicable to other instances in class