SER 321 B Session

SI Session

Thursday November 16th 2023

7:00 - 8:00 pm MST

Agenda

- Review Distributed Structures
- Joining a Peer to Peer Cluster
- Assignment 5 Structure
- Consensus

SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
  - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features

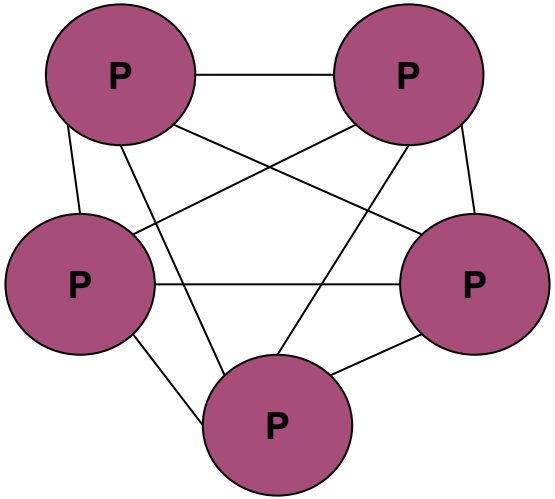
Zoom Chat

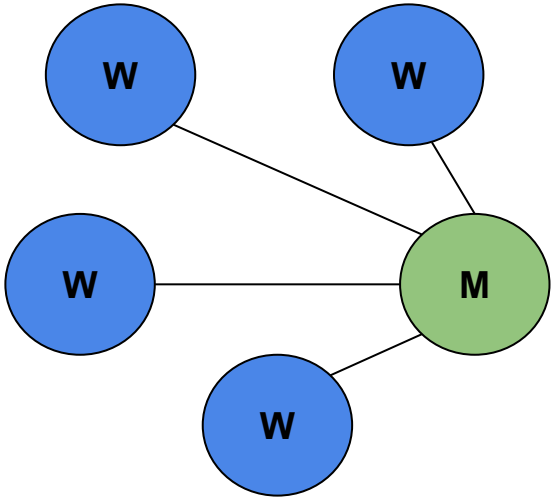
- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

Check out the recording for the discussion!

SER 321

Let's list some differences



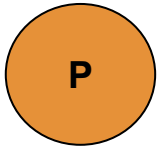


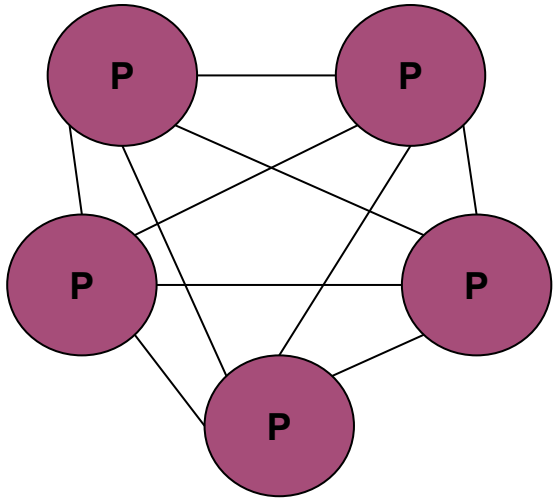
Check out the recording for the discussion!

SER 321

New Nodes

How does our lonely peer join the cluster?





Should we *even let* the peer into the cluster?

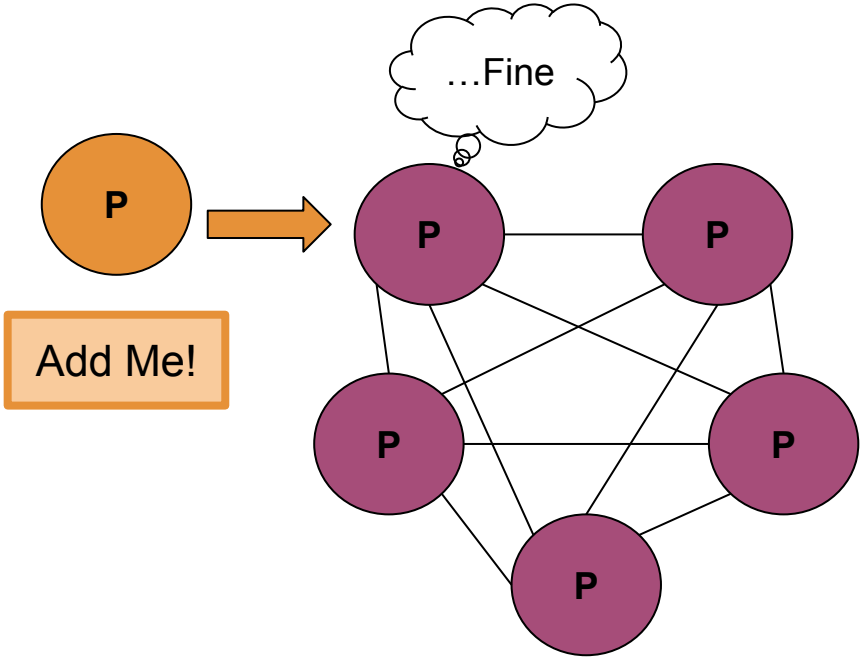
Could use Proof of Work

Assuming we *want* to allow the peer into the cluster…

Check out the recording for the discussion!

SER 321

New Nodes

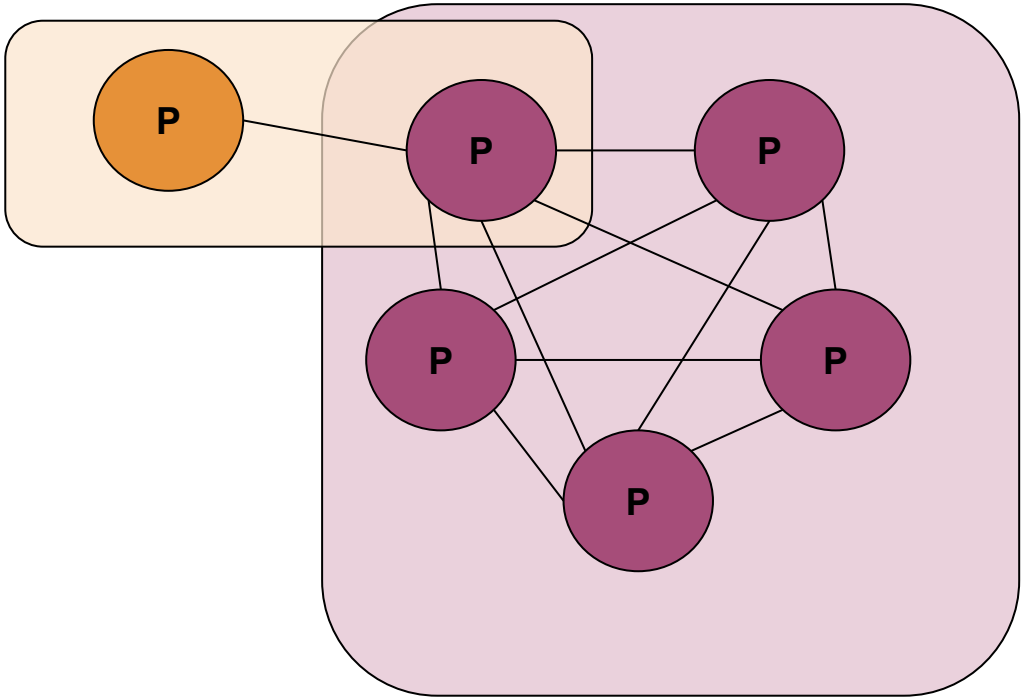


Assuming we *want* to allow the peer into the cluster…

Check out the recording for the discussion!

SER 321
New Nodes

Is that it?



Assuming we *want* to allow the peer into the cluster…

Check out the recording for the solution!

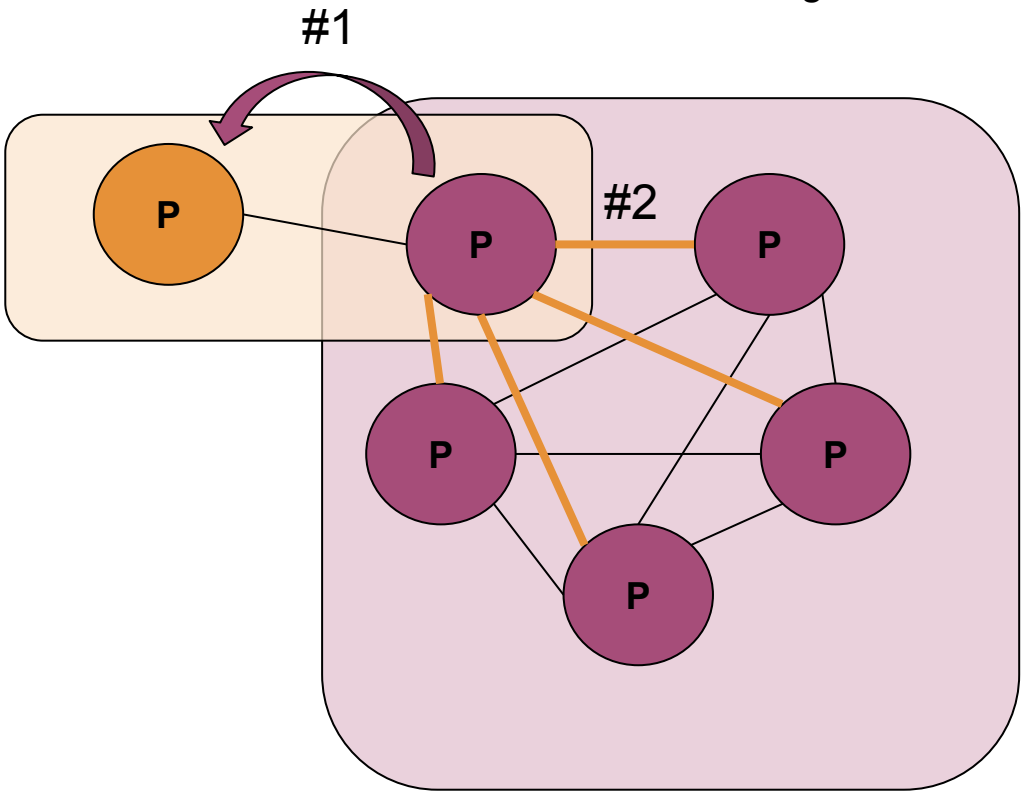
SER 321

New Nodes

Need to do two things…

1.

2.

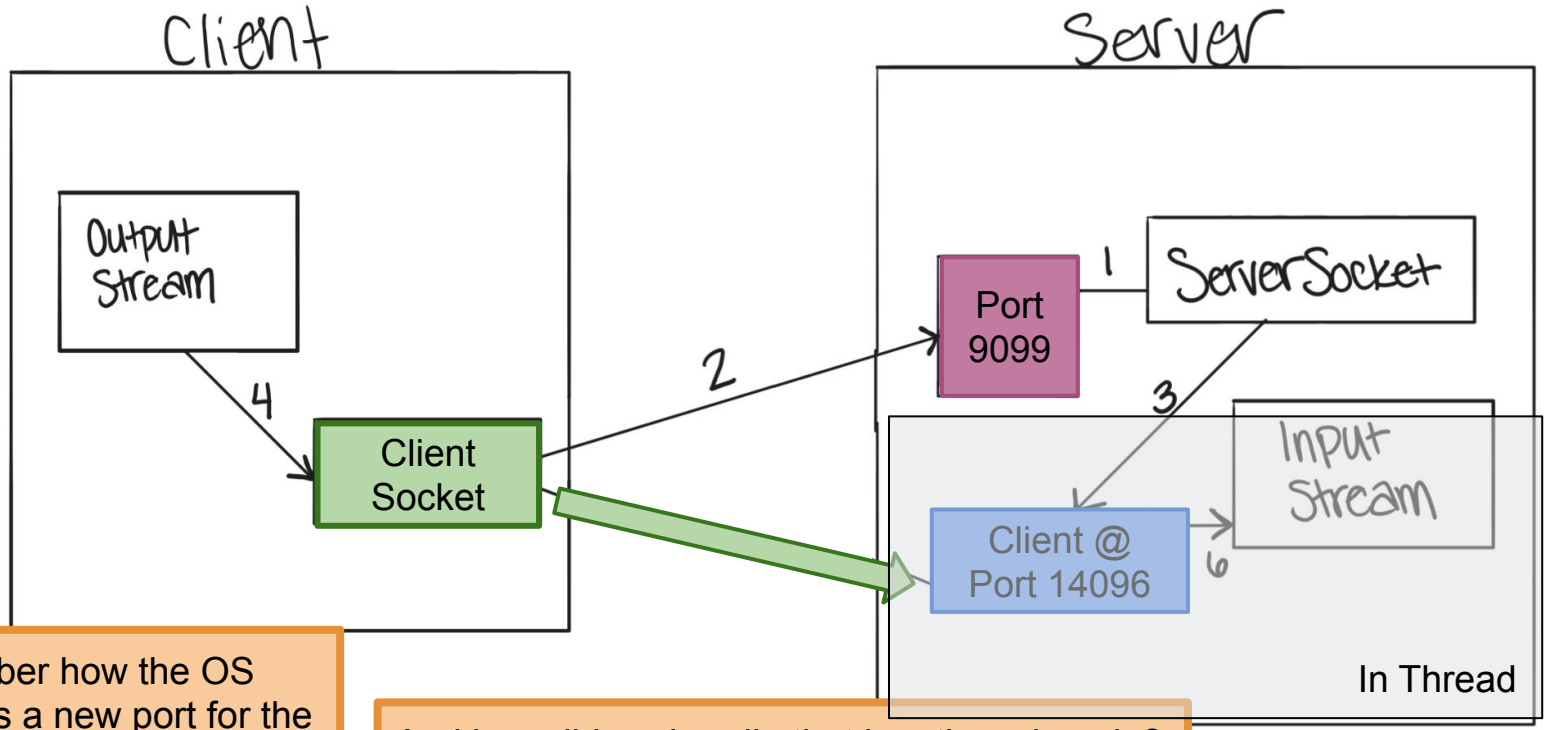


Assuming we *want* to allow the peer into the cluster…

SER 321

Quick Aside for Peer to Peer

```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```



Remember how the OS allocates a new port for the client?

And how did we handle that in a thread again?

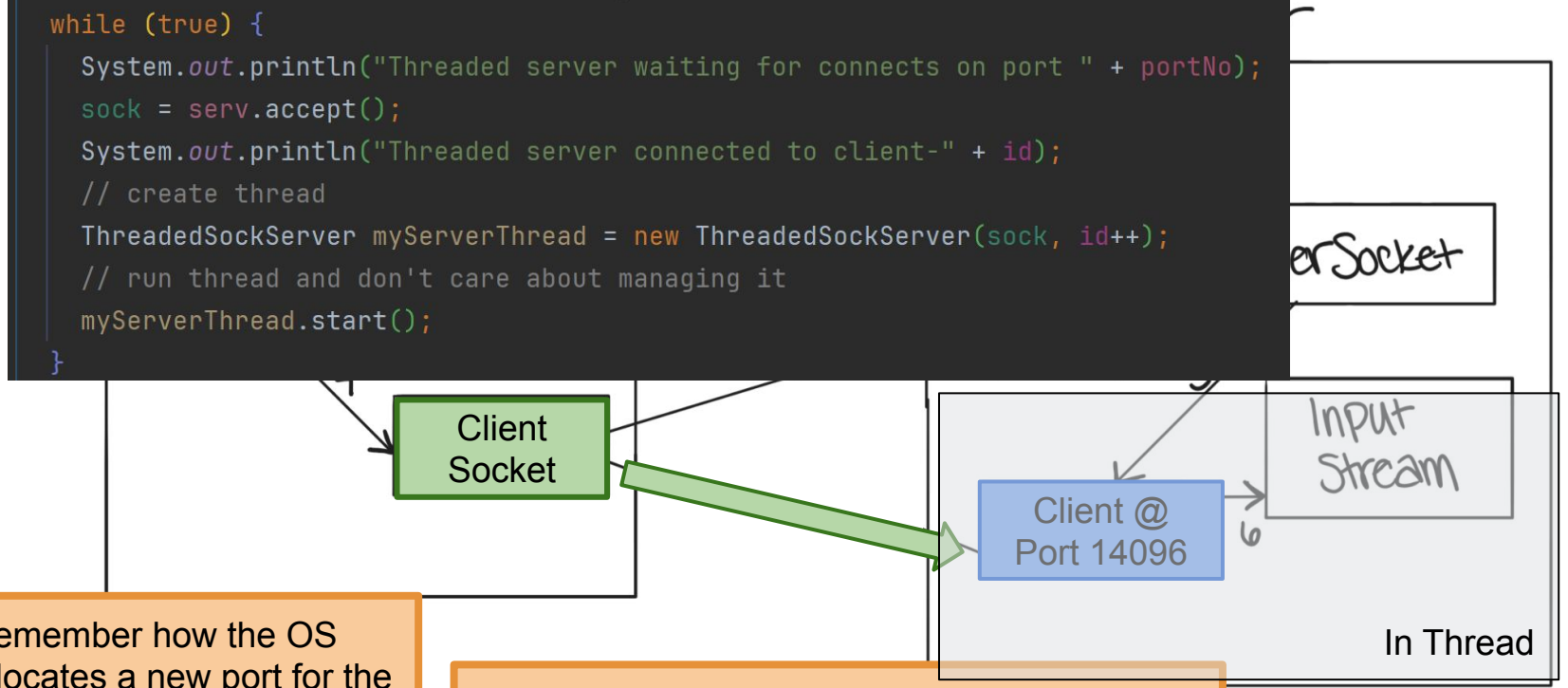
ThreadedSockServer from examples repo

SER 321

Quick Aside for Peer to Peer

```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```

```java
ServerSocket serv = new ServerSocket(portNo);
while (true) {
  System.out.println("Threaded server waiting for connects on port " + portNo);
  sock = serv.accept();
  System.out.println("Threaded server connected to client-" + id);
  // create thread
  ThreadedSockServer myServerThread = new ThreadedSockServer(sock, id++);
  // run thread and don't care about managing it
  myServerThread.start();
}
```



Remember how the OS allocates a new port for the client?

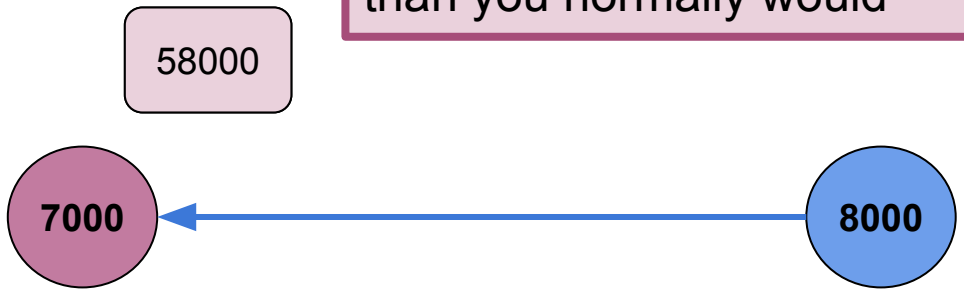
And how did we handle that in a thread again?

SER 321

Quick Aside for Peer to Peer

IF YOU USE PEER TO PEER

Because we use threads to *simulate* the nodes, we have to handle the communication a little differently than you normally would



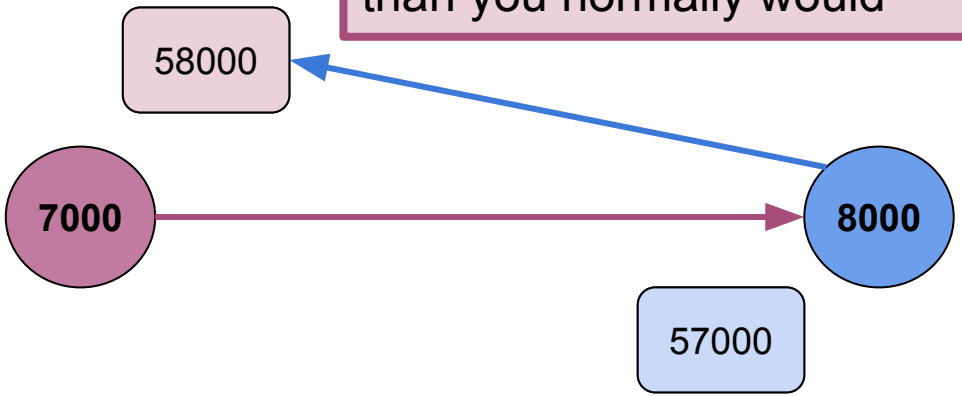
Check out the recording for the discussion!

SER 321

Quick Aside for Peer to Peer

IF YOU USE PEER TO PEER

Because we use threads to *simulate* the nodes, we have to handle the communication a little differently than you normally would



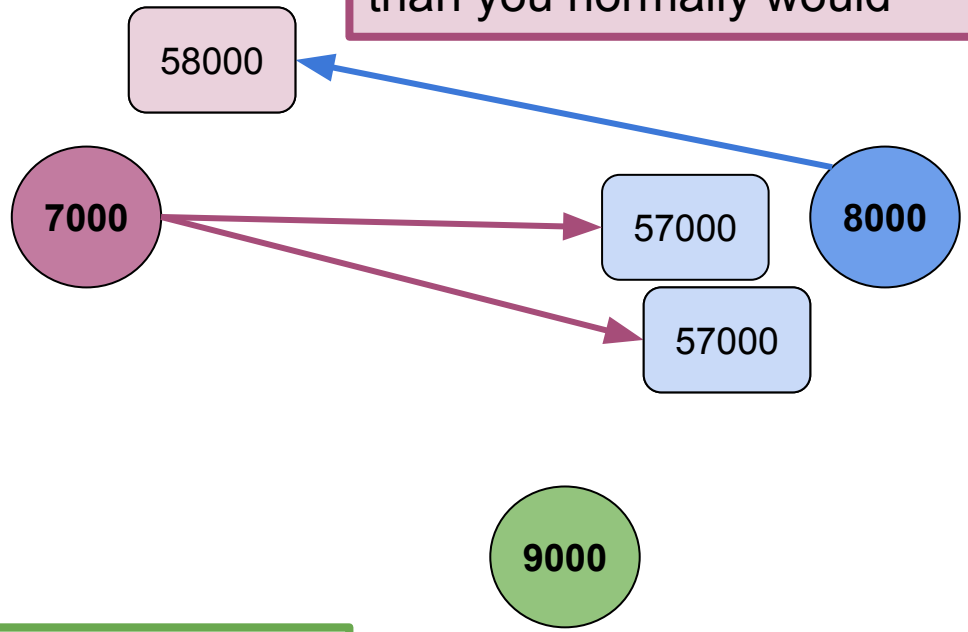
Check out the recording for the discussion!

SER 321

Quick Aside for Peer to Peer

IF YOU USE PEER TO PEER

Because we use threads to *simulate* the nodes, we have to handle the communication a little differently than you normally would



What happens for this third peer?

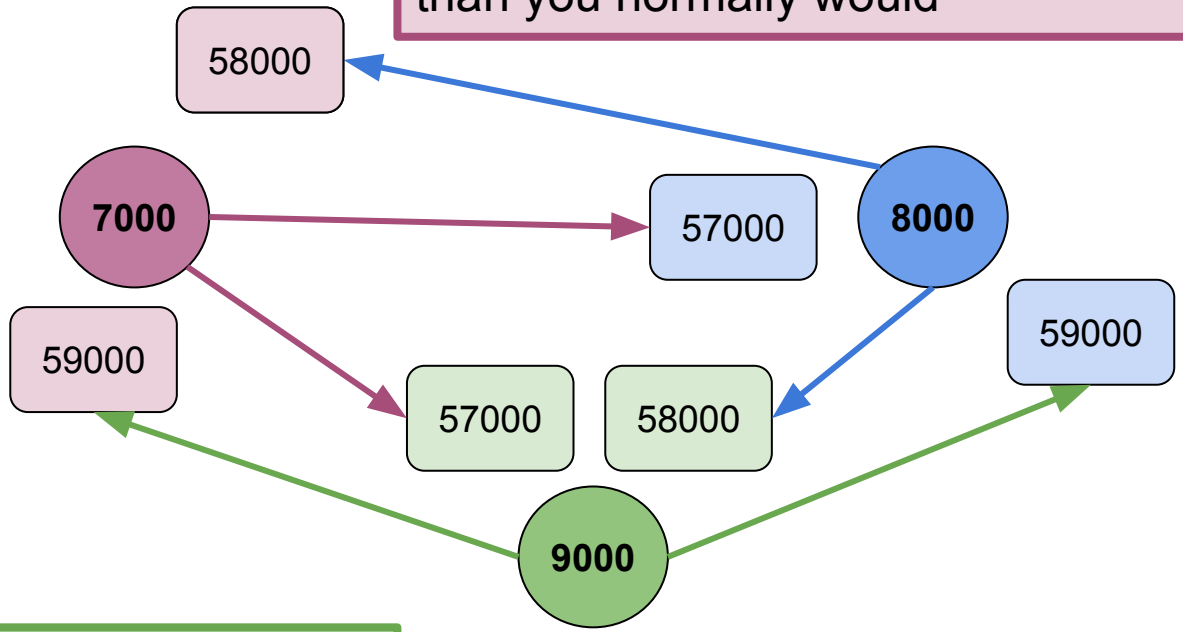
Check out the recording for the discussion!

SER 321

Quick Aside for Peer to Peer

IF YOU USE PEER TO PEER

Because we use threads to *simulate* the nodes, we have to handle the communication a little differently than you normally would



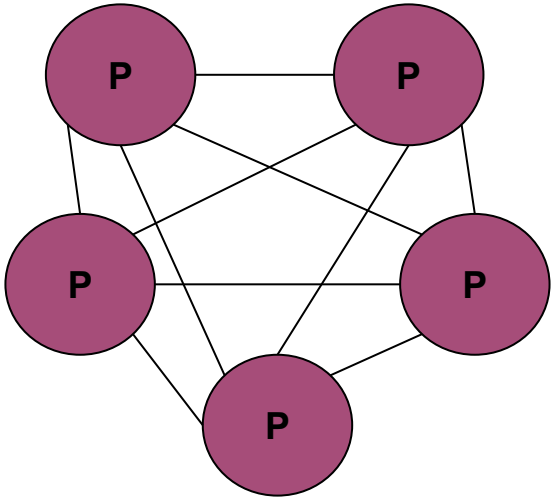
What happens for this third peer?

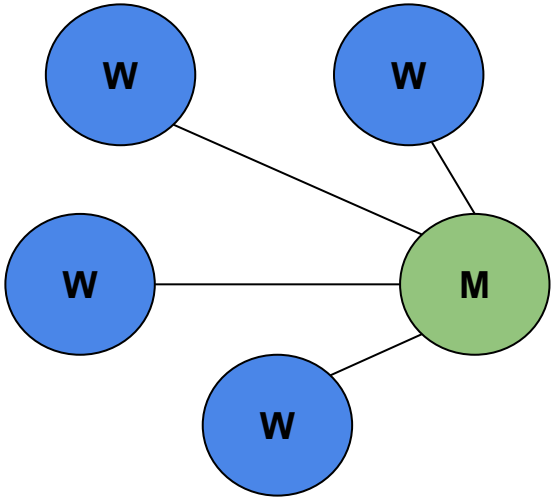
Check out the recording for the discussion!

Check out the recording for the discussion!

SER 321
Assignment Structure

What distributed structure do you want to use for Assignment 5?



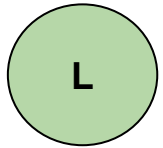


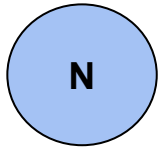
SER 321

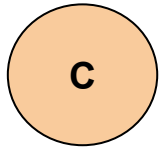
Assignment Structure

We need:

How Many?







Client

Leader

Nodes

Check out the recording for the solution!

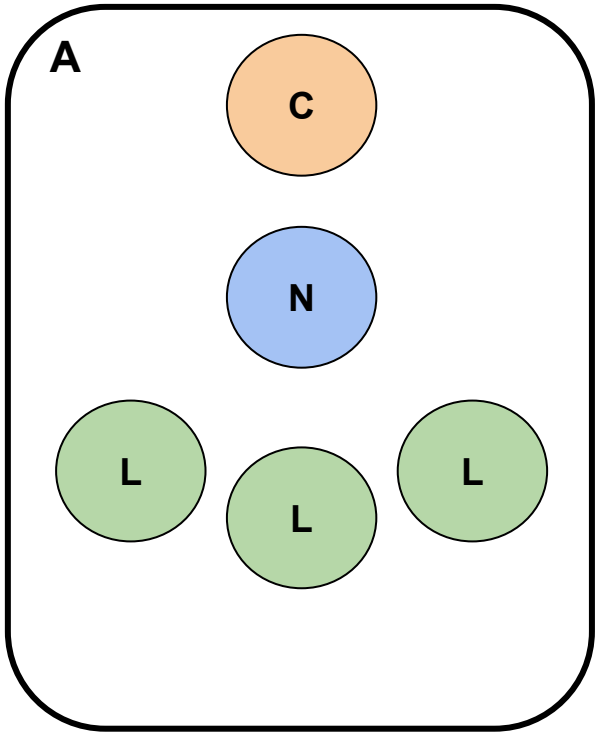
SER 321
Assignment Structure

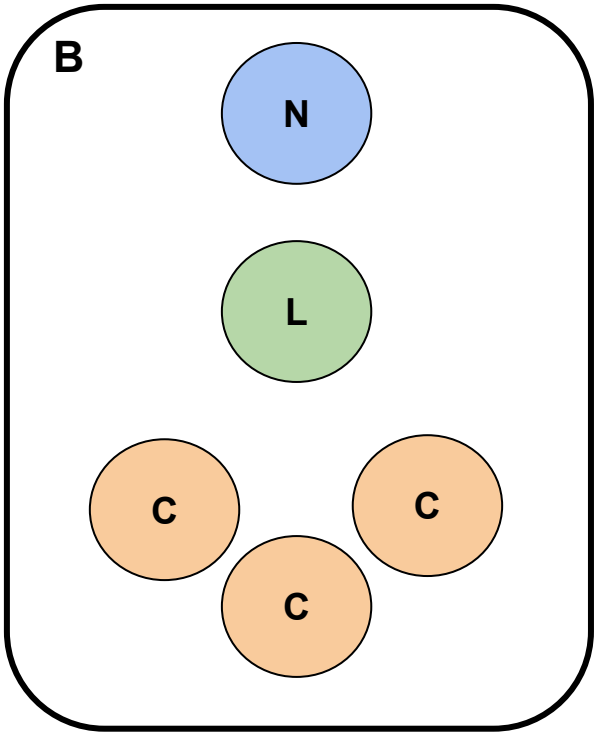
Which of these would work for this implementation?

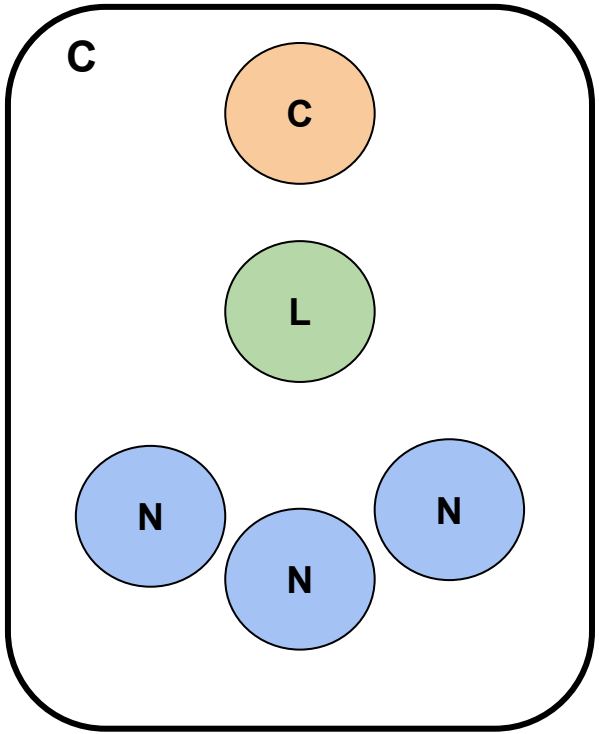
Client

Leader

Node



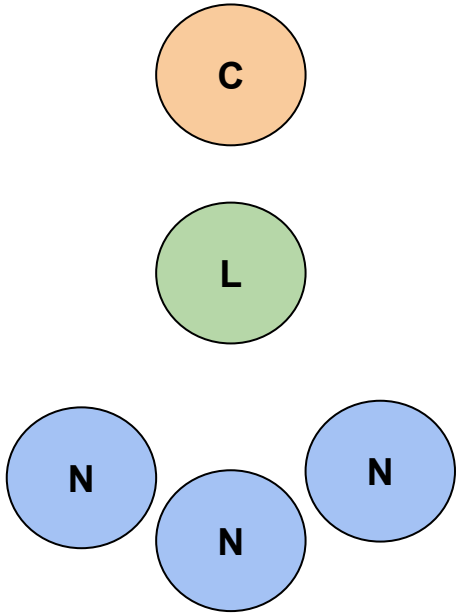


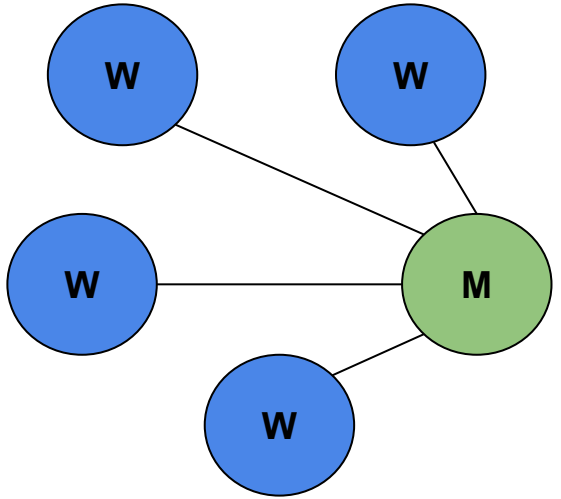


Check out the recording for the solution!

SER 321
Assignment Structure

Hrm, looks familiar…

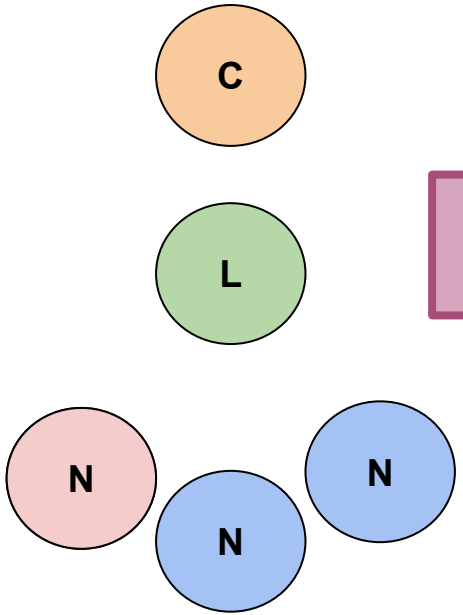




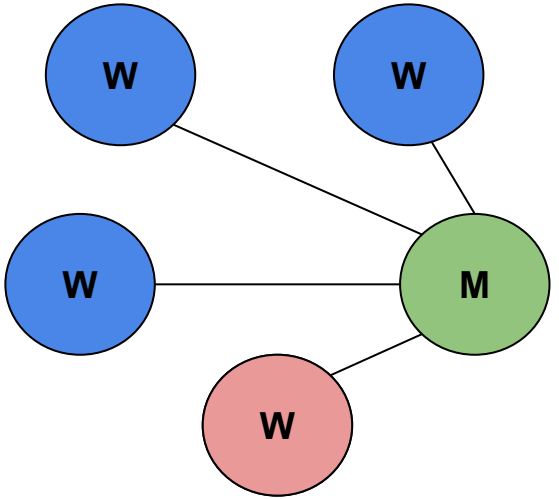
SER 321

Assignment - Consensus

How do we *find* and *handle* **faulty** nodes?



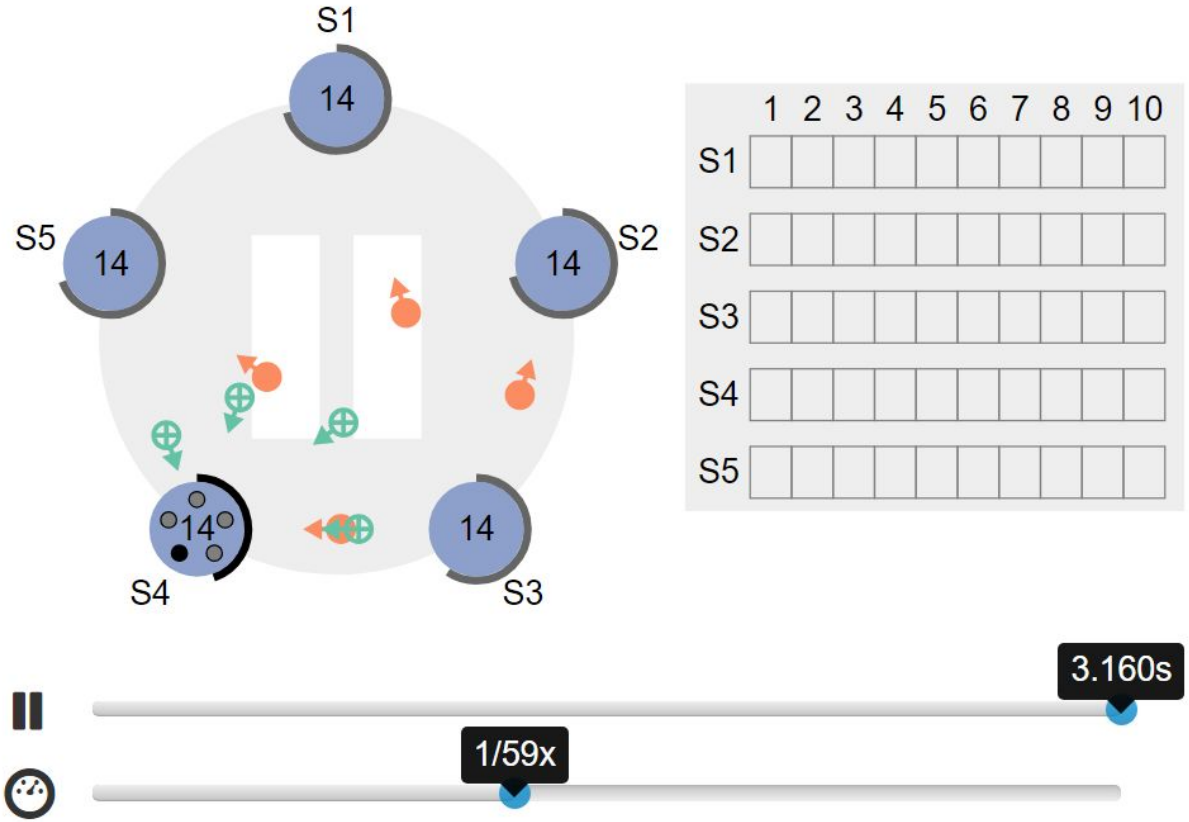
Consensus!



SER 321

RAFT

RAFT



Questions?

Survey:

https://bit.ly/asn_survey

22

Upcoming Events

SI Sessions:

- Sunday, November 19th 2023 at 7:00 pm MST
- Monday, November 20th 2023 at 4:00 pm MST
- Sunday, November 26th 2023 at 7:00 pm MST

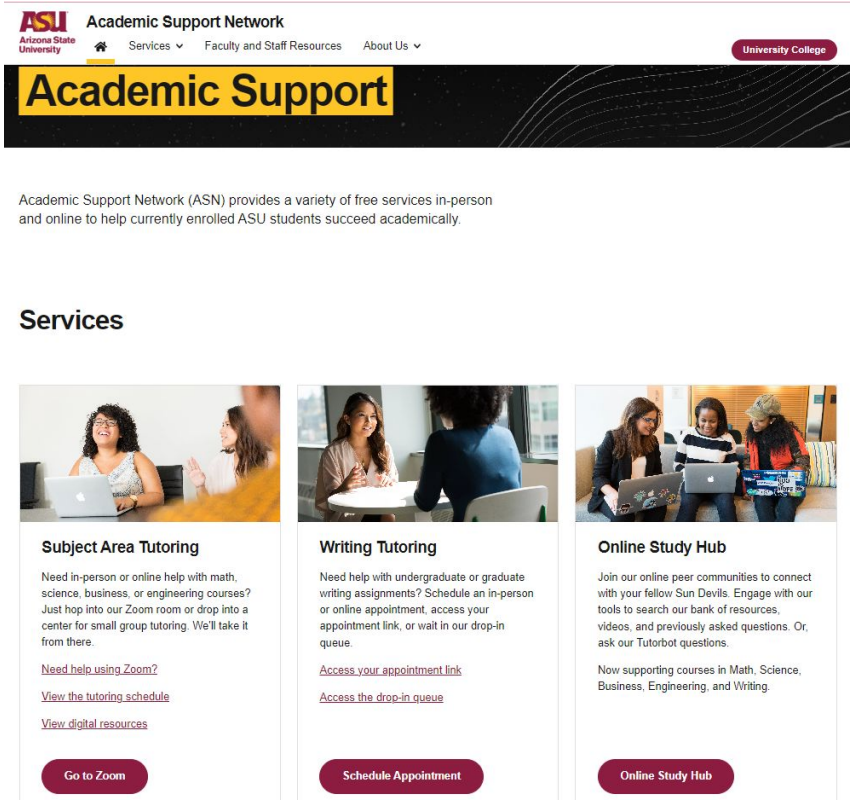
Review Sessions:

- **Monday, November 27th 2023**
- **5:00 - 7:00 pm MST**

More Questions?

Check out our other resources!

tutoring.asu.edu

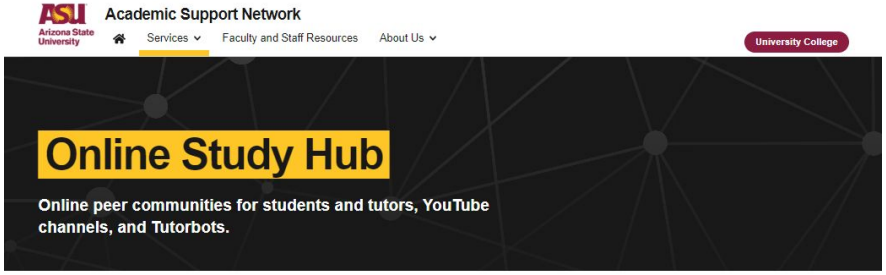


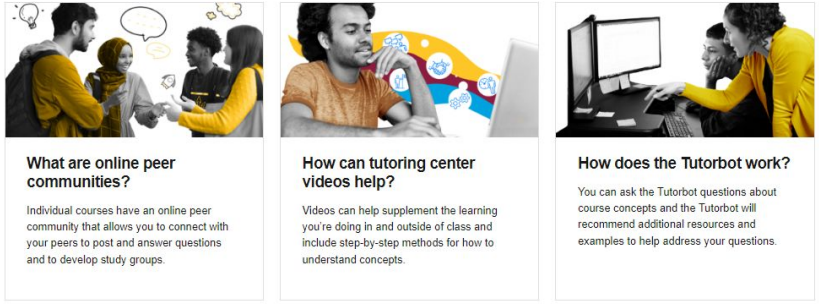
1. **Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.**
2. **Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.**

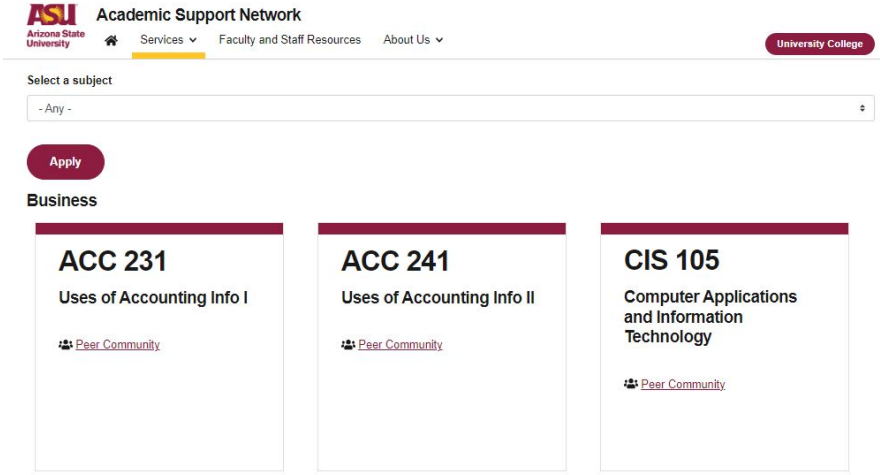
More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub







Don't forget to check out the Online Study Hub for additional resources!

Additional Resources

CoureRepo

Dining Philosophers Interactive

Raft Interactive