

ELEC / COMP 177 – Fall 2015

Computer Networking

→ Network Tester (Project 3 and 4)

Some slides from Kurose and Ross, *Computer Networking*, 5th Edition

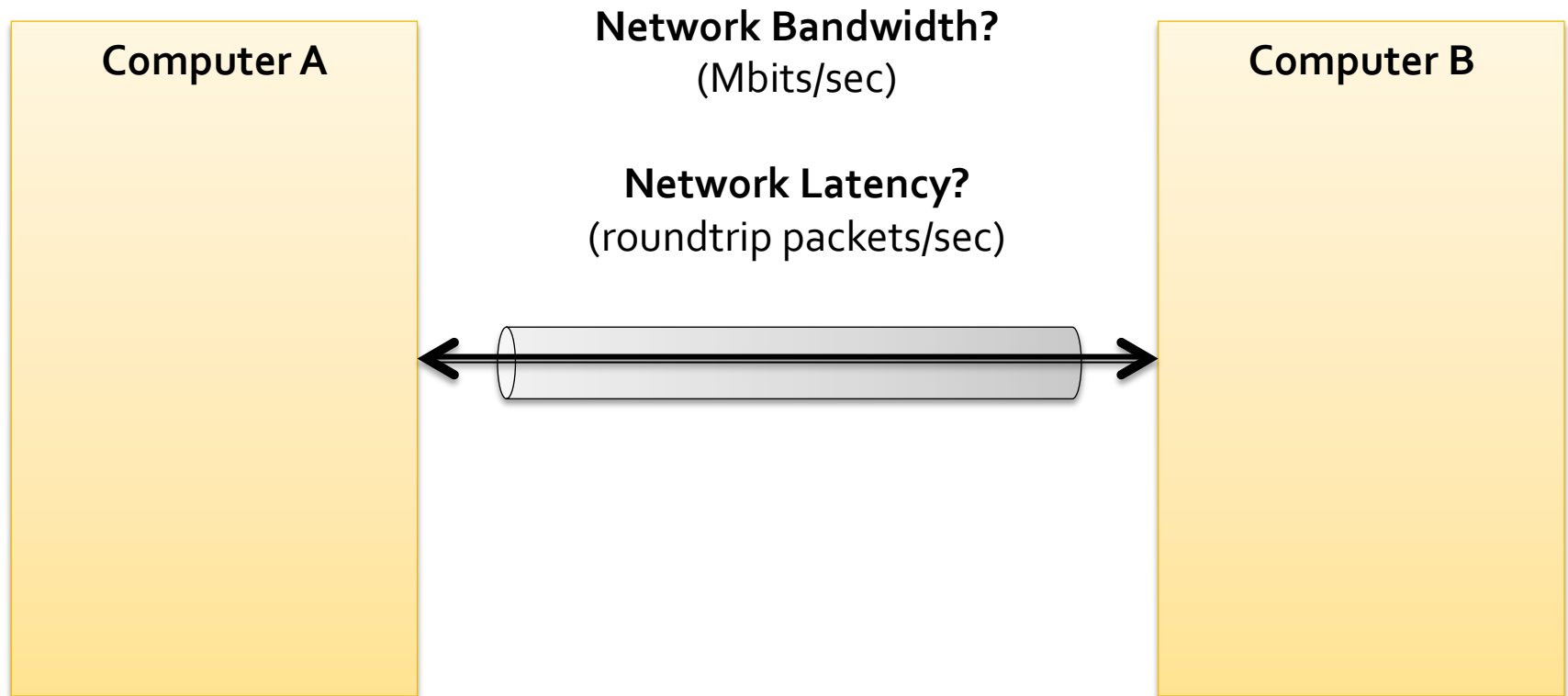
Upcoming Schedule

- **Presentation 2 – Privacy/Security**
 - Topic Approval – **Due Today**

Network Tester

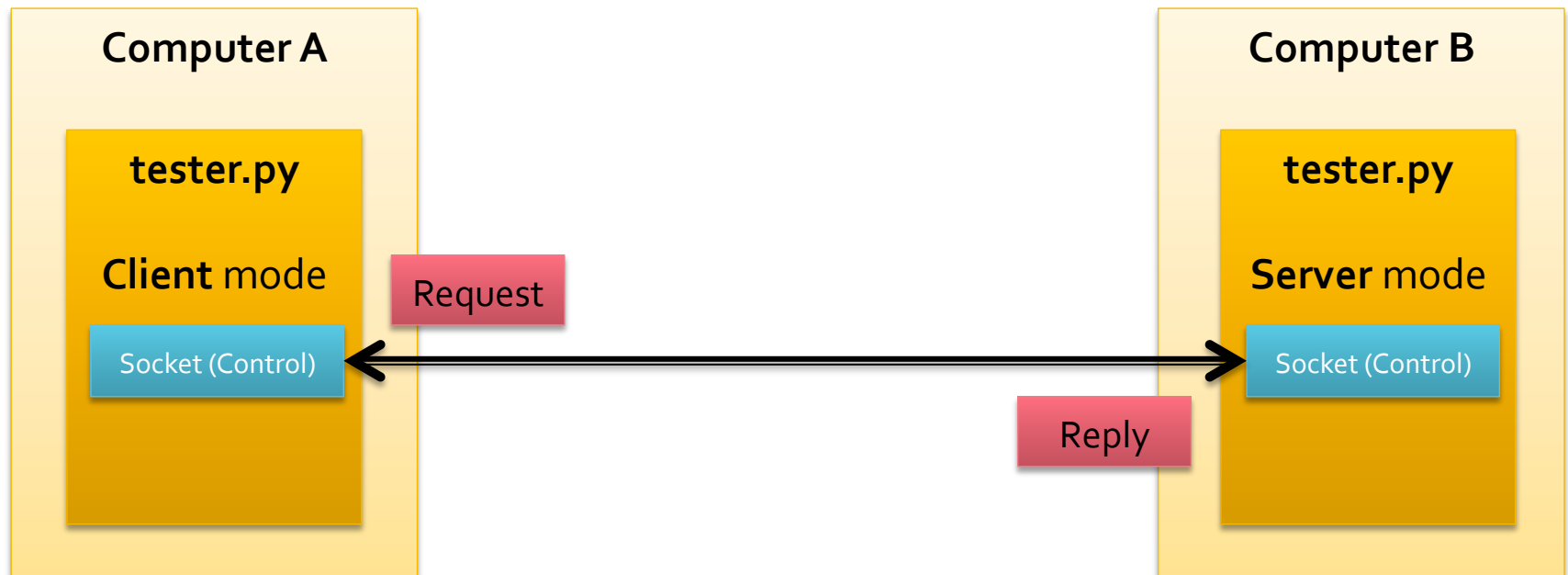
Project 3 and 4

Network



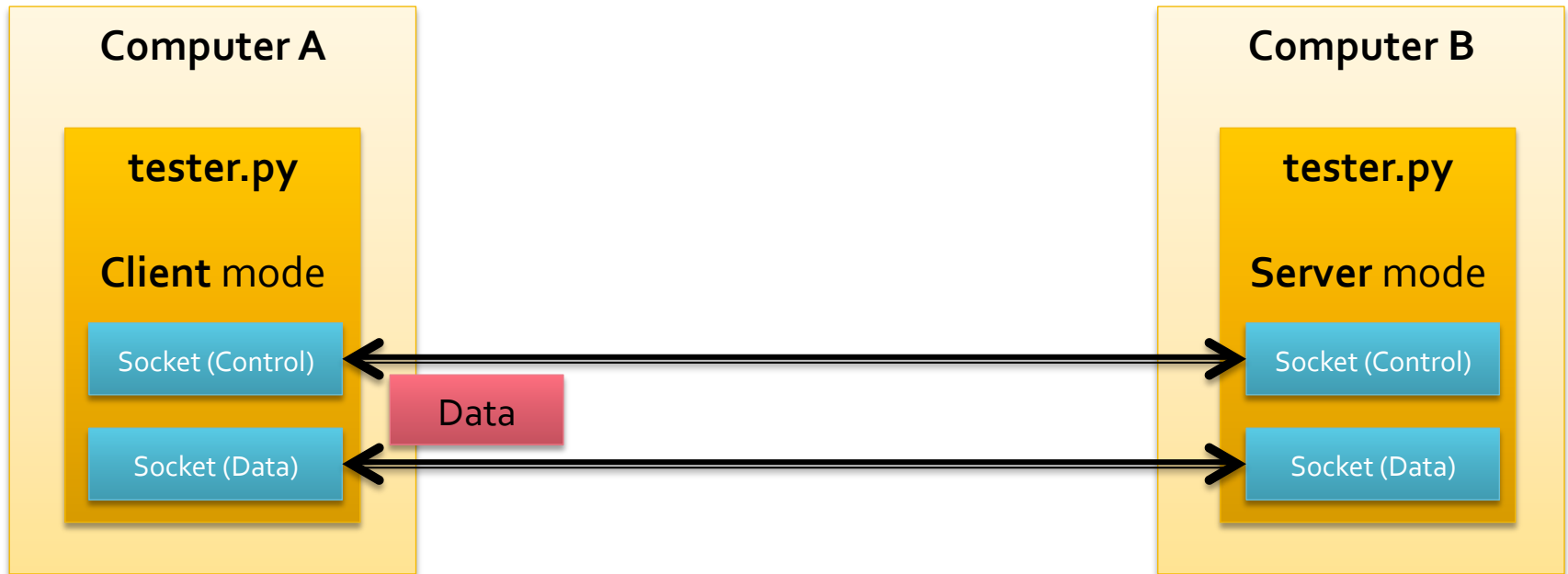
tester.py Benchmark Program

Showing TCP STREAM test...



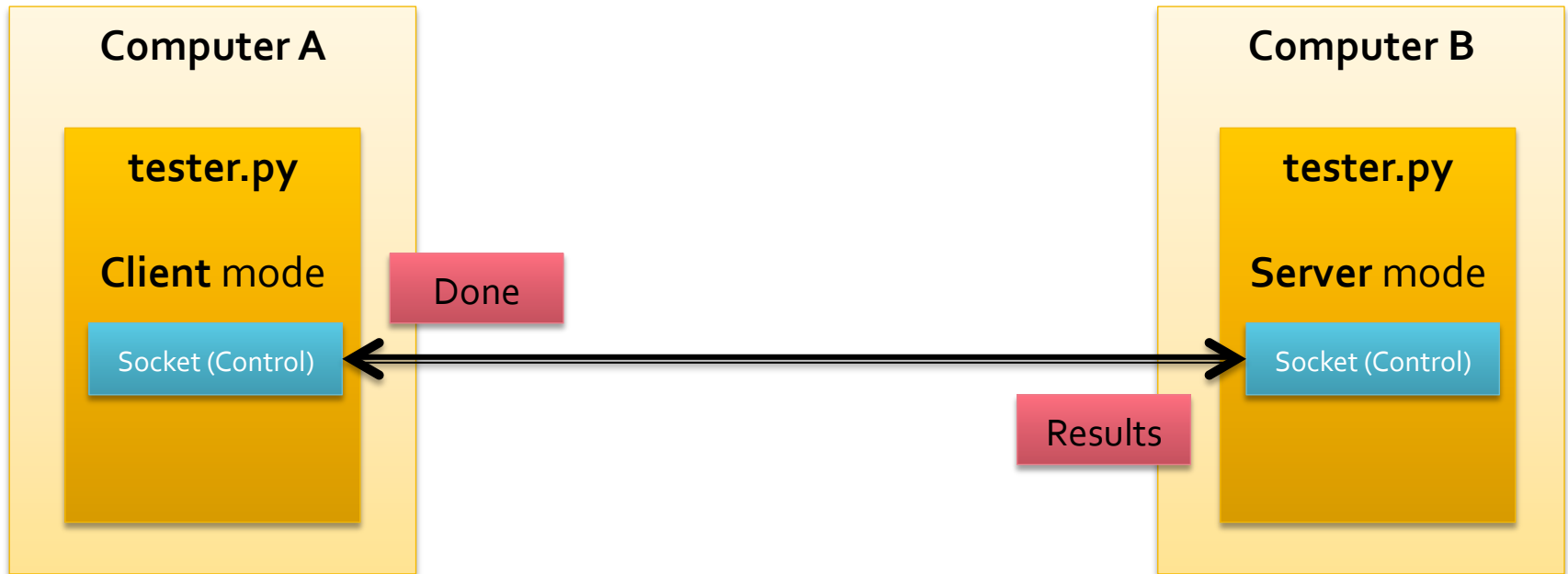
tester.py Benchmark Program

Showing TCP STREAM test...



tester.py Benchmark Program

Showing TCP STREAM test...



Demo

- TCP_STREAM demo
- TCP_RR demo
- Note how server stays running after a test finishes, waiting for more tests...
- Compare *localhost* test versus network test

Test: TCP_STREAM

- Client continuously sends TCP data to server
- Message size: 64kB
- Measurements:
 - Bandwidth: Megabits/sec
- Goal: Result should be limited by network bandwidth, not by crappy Python coding!
 - *I was able to get ~900Mbits/sec on gigabit Ethernet*

Test: UDP_STREAM

- Client continuously sends UDP data to server
- Message size: 1472 bytes
- Measurements:
 - Bandwidth: Megabits/sec
 - Packet loss (percentage of total packets sent)

Test: TCP_RR

- Client sends 1-byte TCP message to server
- Server sends 1-byte TCP message to client
- Repeat...
 - "RR" = "Round Robin"
- Measurements:
 - Roundtrip packets/sec

Test: UDP_RR

- Client sends 1-byte UDP message to server
- Server sends 1-byte UDP message to client
- Repeat...

- Measurements:
 - Roundtrip packets/sec
 - Packet loss
 - What will happen if we lose a packet on this test?

Parallel Programming

- Must use threads for this project for both client and server
 - Main (original) thread – control tasks
 - Child (new) thread – data tasks
- Why do we need it?
 - 1 problematic case: UDP_STREAM / UDP_RR test
 - How does the server know when the client has finished?

Parallel Programming

- Does the server need to allow multiple clients to connect simultaneously and run tests?
- Ans: **No**, that would distort the results of our bandwidth measurement, making the tool useless. Subsequent clients should wait for the first one to finish (serial, not parallel)