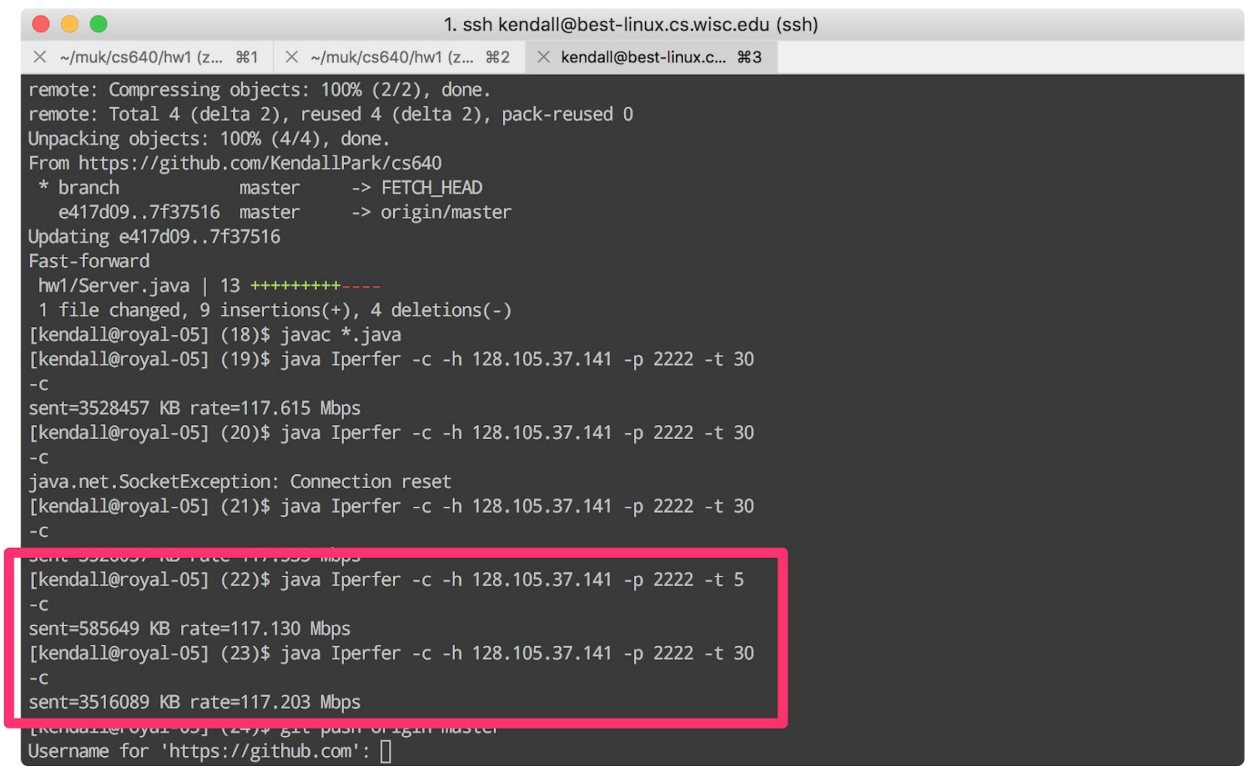
Kendall Park
Paul Lipka
Assignment 1

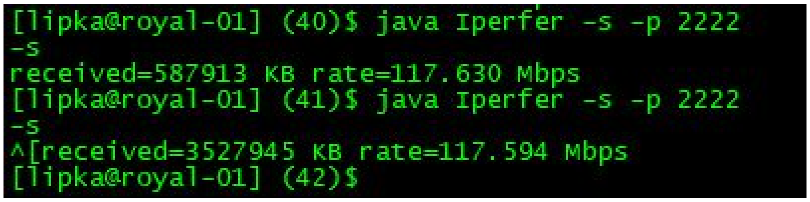
Part 1 - Iperfer on Wired Environment

Under which environment - wired or wireless, would you expect greater throughput?

We expect that a wired environment will have greater throughput. Wireless is subject to have greater noise (and therefore, error checking, dropped packets, etc) while wired is insulated from the environment. Additionally, with wired connections, the heads of the sender and receptor are directly connected, whereas with wireless the transmission is broadcasted . This lowers the signal-to-noise ratio, which lowers the capacity of the transmission.

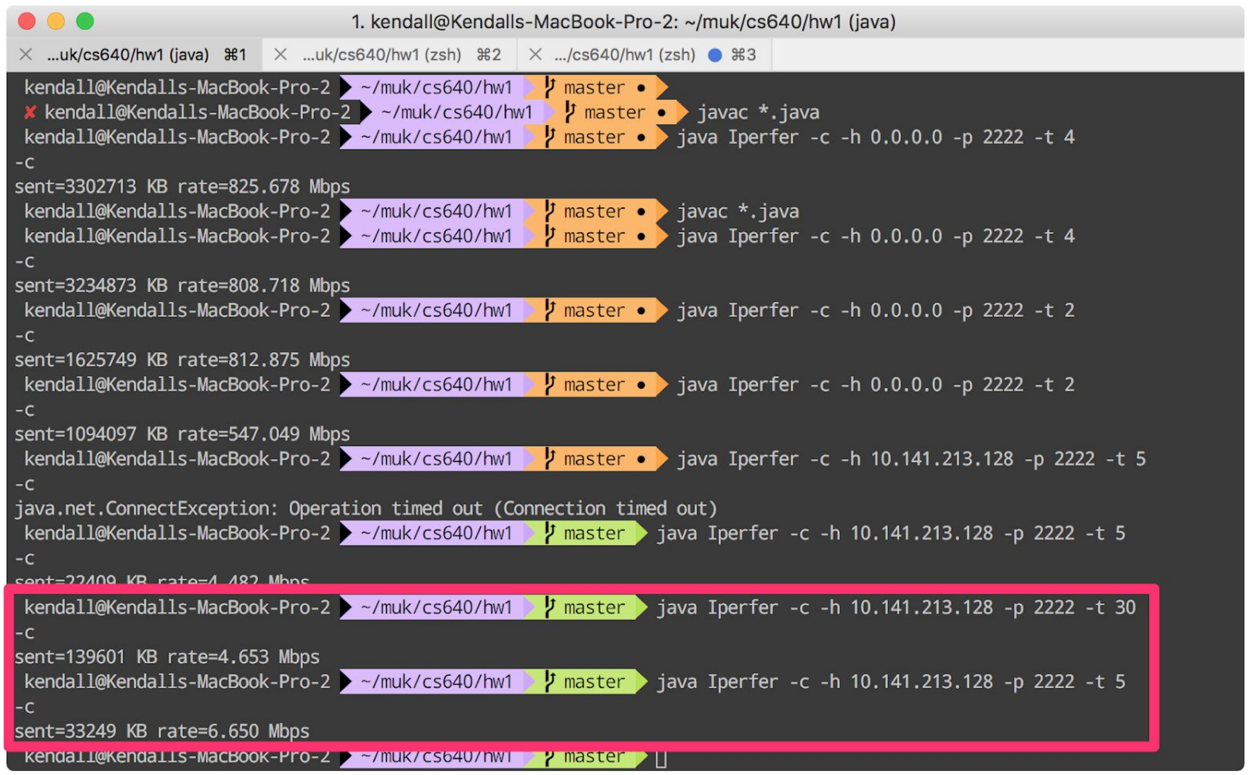
NOTE: We initially tested this with MBps, not Mbps. We corrected the units later on. During office hours, I was told this would not matter.



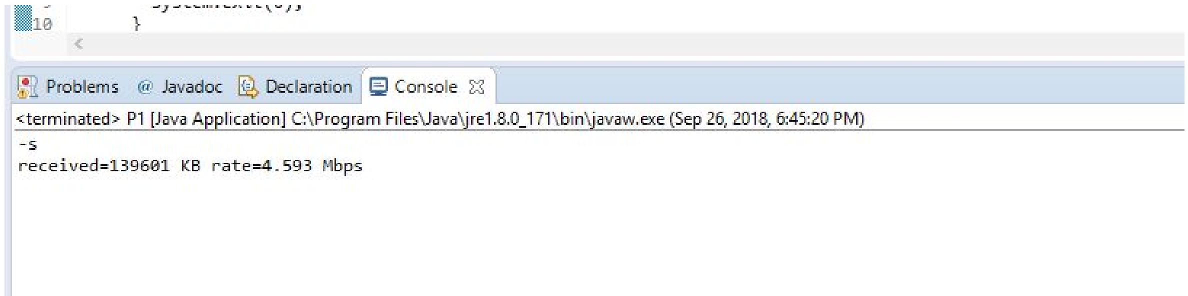


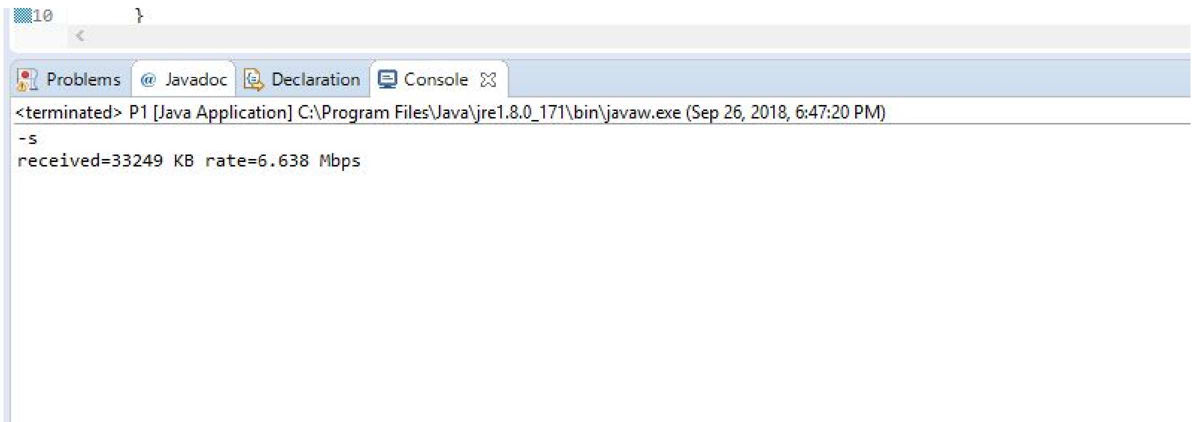
```
[lipka@royal-01] (40)$ java Iperfer -s -p 2222
-s
received=587913 KB rate=117.630 Mbps
[lipka@royal-01] (41)$ java Iperfer -s -p 2222
-s
^[received=3527945 KB rate=117.594 Mbps
[lipka@royal-01] (42)$
```

Part 2 - Iperfer on Wireless Environment



```
1. kendall@Kendalls-MacBook-Pro-2: ~/muk/cs640/hw1 (java)
...uk/cs640/hw1 (java) ⌘1   ...uk/cs640/hw1 (zsh) ⌘2   .../cs640/hw1 (zsh) ⌘3
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●
✘ kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  javac *.java
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  java Iperfer -c -h 0.0.0.0 -p 2222 -t 4
-c
sent=3302713 KB rate=825.678 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  javac *.java
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  java Iperfer -c -h 0.0.0.0 -p 2222 -t 4
-c
sent=3234873 KB rate=808.718 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  java Iperfer -c -h 0.0.0.0 -p 2222 -t 2
-c
sent=1625749 KB rate=812.875 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  java Iperfer -c -h 0.0.0.0 -p 2222 -t 2
-c
sent=1094097 KB rate=547.049 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master ●  java Iperfer -c -h 10.141.213.128 -p 2222 -t 5
-c
java.net.ConnectException: Operation timed out (Connection timed out)
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master  java Iperfer -c -h 10.141.213.128 -p 2222 -t 5
-c
sent=22409 KB rate=4.482 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master  java Iperfer -c -h 10.141.213.128 -p 2222 -t 30
-c
sent=139601 KB rate=4.653 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master  java Iperfer -c -h 10.141.213.128 -p 2222 -t 5
-c
sent=33249 KB rate=6.650 Mbps
kendall@Kendalls-MacBook-Pro-2  ~/muk/cs640/hw1  master
```





Did it match the prediction you made at the start of this section? In either case, explain your results.

The rate of transfer for 30 seconds was ~117 Mbps and ~4.7 Mbps for the wired and wireless transmissions, respectively. This aligns with our hypothesis that the wired environment would have greater throughput.

Part 3: Measurements in Mininet

Q1: Link Latency and Throughput

First, you should measure the RTT and bandwidth of each of the five individual links between switches (`L1` - `L5`). You should run `ping` with 30 packets and store the output of the measurement on each link in a file called `latency_L#.txt`, replacing `#` with the link number from the topology diagram above. You should run `Iperfer` for 30 seconds and store the output of the measurement on each link in a file called `throughput_L#.txt`, replacing `#` with the link number from the topology diagram above.

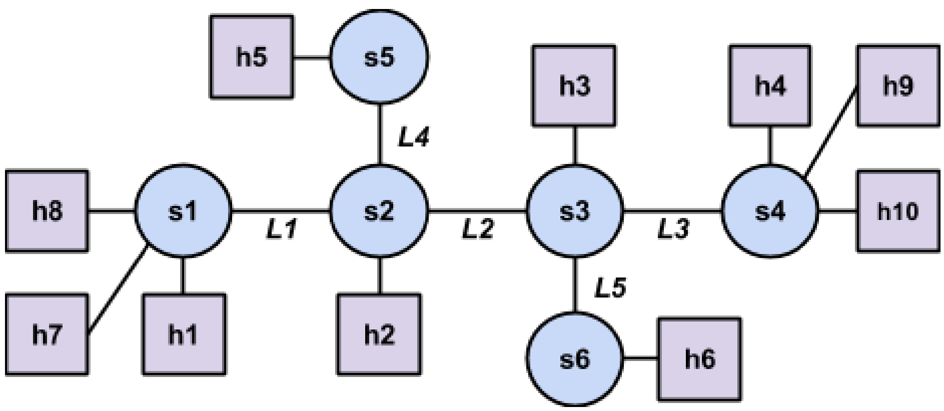
Measure RTT and bandwidth of each of the five individual links between switches

1. Run ping with 30 packets and store the output of the measurement on each link in a file called latency_L#.txt

2. Run Iperfer for 30 seconds and store the output of the measurement on each link in a file called throughput_L#.txt

Q2: Path Latency and Throughput

Now, assume h1 wants to communicate with h4. **What is the expected latency and throughput of the path between the hosts?** Put your prediction in your answers.pdf file. Measure the latency and throughput between h1 and h4 using ping and Iperfer. It does not matter which host is the client and which is the server. Use the same parameters as above (30 packets / 30 seconds) and store the output in files called latency_Q2.txt and throughput_Q2.txt. **Put the average RTT and measured throughput in your answers.pdf file and explain the results. If your prediction was wrong, explain why.**



L1 average RTT = 80.407ms
L2 average RTT = 20.329ms
L3 average RTT = 40.336ms

Expected two-way latency (RTT) would be approximately (80 + 20 + 40) = 140ms; we add the latency between the switches to get the total latency; we divide by two because latency is only the time it takes for a one-way trip, not a round trip.

Expected throughput would be min(L1, L2, L3) throughput which is approximately 20 Mbps; L1 is the bottleneck, you can't send data through L1 at a faster rate so throughput for the entire transmission is limited to L1's throughput.

Findings:
Average RTT of h1 to h4 = 140.653ms
Throughput was is 18.152 Mbps, which is close to our prediction of ~20 Mbps.

The above match our predictions.

Q3: Effects of Multiplexing

Next, assume multiple hosts connected to s1 want to simultaneously talk to hosts connected to s4. **What is the expected latency and throughput when two pairs of hosts are communicating simultaneously? Three pairs?** Put your predictions in your answers.pdf file.
Use ping and Iperfer to measure the latency and throughput when there are two pairs of hosts communicating simultaneously; it does not matter which pairs of hosts are communicating as long as one is connected to s1 and one is connected to s4 . Use the same parameters as above. You do not need to submit the raw output, but you should put the average RTT and measured throughput for each pair in your answers.pdf file and explain the results. If your prediction was wrong, explain why.
Repeat for three pairs of hosts communicating simultaneously.

Since the latency only pertains to the speed it takes for one packet to reach its destination, it should stay the same (~140ms) in both two or three pairs of simultaneous talk. However, because the bandwidth will be split between the data being sent over, the throughput should be split between the maximum throughput of the connection (~20 Mbps). We expect the throughput of the two or three pairs sharing the connection to add up to ~20 Mbps.

RTT h1 -> h4 = 140.728ms
RTT h7 -> h9 = 140.668ms
Throughput h1 -> h4 = 12.596 Mbps
Throughput h7 -> h9 = 6.524 Mbps

RTT h1 -> h4 = 140.465ms
RTT h4 -> h1 = 140.482ms
RTT h4 -> h7 = 140.821ms
Throughput h1 -> h4 = 10.432 Mbps
Throughput h7 -> h9 = 7.347 Mbps
Throughput h8 -> h10 = 4.802 Mbps

Our predictions match our results.

Q4: Effects of Latency

Lastly, assume h1 wants to communicate with h4 at the same time h5 wants to communicate with h6. **What is the expected latency and throughput for each pair?** Put your prediction in your answers.pdf file.
Use ping and Iperfer to conduct measurements, storing the output in files called latency_h1-h4.txt, latency_h5-h6.txt, throughput_h1-h4.txt, and throughput h5-h6.txt. Put the average RTT and measured throughput in your answers.pdf file and explain the results. **If your prediction was wrong, explain why.**

We predict the latency (RTT) for H1 -> H4 to be the latency of L1 + L2 + L3 = ~140ms
We predict the latency (RTT) for H5 -> H6 to be the latency of L4 + L2 + L5 = 30 + 20 + 30 = ~80ms

We predict the throughput of H1 -> H4 to be the throughput of L1 (~20 Mbps), as it is the bottleneck of L1, L2, and L3.
We predict the throughput of H5 -> H6 to be the throughput of L4 or L5 (~30 Mbps), as they are the bottleneck of L4, L2, and L5.

H1 -> H4 RTT = 140.751ms
H5 -> H6 RTT = 80.739ms

H1 -> H4 rate = 18.048ms Mbps
H5 -> H6 rate = 27.752 Mbps

Our predictions match our experimental results.