Assignment 1: Sockets, Mininet, & Performance

CS640 Fall 2019

Henry Xu,
Yudai Yaguchi

Part 1:

For time validation, we check if the time is a positive integer. For hostname validation, we check if the IP address contains four sections, and if each section is an integer between 0 and 255.

For comparison between wired vs wireless, we had expected the wired one has bigger throughput in both client and server, because wired cable should provide faster internet access.

Part 1- Iperfer on Wired Environment.
(using CS lab Linux Machines)
Client

```
[hxu@emperor-05] (5)$ java Iperfer -s -p 9999
received=58644 KB rate=11.703053282777889 Mbps
[hxu@emperor-05] (6)$
```

Server

```
[yudai@emperor-06] (14)$ java Iperfer -c -h 128.105.37.245 -p 9999 -t 5
Connectiong to 128.105.37.245 @ port number 9999
Connection Established

sent 58644 KB rate=11.7288 Mbps
[yudai@emperor-06] (15)$
```

Part 1- Iperfer on Wireless Environment.
(using home Wifi)
Client

```
  2019-09-19 17:14:57  Yudais-MacBook-Pro in ~/Desktop/UW-Madison/2019 fall/CS 640/h1
 ○ → java Iperfer -c -h 10.0.0.57 -p 9999 -t 5
 Connectiong to 10.0.0.57 @ port number 9999
 Connection Established

 sent 72281 KB rate=14.4562 Mbps
```

Server

```
henry@henry-ThinkPad-S2-2nd-Gen:~/Desktop/fa19/cs640/proj1/iperfer$ java Iperfer -s
 -p 9999
received=72281 KB rate=14.245368545526214 Mbps
henry@henry-ThinkPad-S2-2nd-Gen:~/Desktop/fa19/cs640/proj1/iperfer$
```

Part 3:

Q1

L1: h1->h2
80 ms,
sent 83763 KB rate=2.7921 Mbps
received=83763 KB rate=2.375311932849365 Mbps

L2: h2->h3
20 ms,
sent 191473 KB rate=6.382433333333334 Mbps
received=191473 KB rate=5.962661933233682 Mbps

L3: h3->h4
40 ms,
sent 155996 KB rate=5.199866666666667 Mbps
received=155996 KB rate=4.756700716572649 Mbps

L4: h2->h5
30 ms,
sent 115794 KB rate=3.8598 Mbps
received=115794 KB rate=3.577090605789132 Mbps

L5: h3->h6
30 ms,
sent 115972 KB rate=3.865733333333333 Mbps
received=115972 KB rate=3.5789408714973456 Mbps

Q2

Prediction: Since it is using, L1, L2, and L3, we thought the latency is the summation of those. So, the answer should be 80 + 20 + 40 = 140ms.
For throughput, since we need to go through L1, L2, and L3, the bottleneck of these becomes the speed. We learned this from the class.
Latency = 140 ms, throughput = received=2.375311932849365 Mbps (bottleneck)

Average RTT = 140.3 ms
Actual throughput:
sent 79611 KB rate=2.6537 Mbps
received=79611 KB rate=2.2567395186665533 Mbps
Our prediction is correct.

Q3

Two pairs: h1->h4 & h7->h9
Prediction:

Latency = 280 ms, throughput = 1.19 mbps. Because we assume the network is using static multiplexing, then the throughput of two pairs should be the same, i.e. half of the throughput in Q2, and the latency should be twice.
Actual:
H1h4 average RTT = 140.3. h7h9 average RTT = 140.3.
h1h4 measured throughput:
sent 54391 KB rate=1.8130333333333333 Mbps
received=54391 KB rate=1.7061702060917845 Mbps
h7h9 measured throughput:
sent 19185 KB rate=0.6395 Mbps
received=19185 KB rate=0.607581707626045 Mbps

The latency is the same as q2 and the summation of these two throughputs is equal to q2 throughput, but h1h4 pair is about three times faster than h7h9 pair.

Explanation:
This network is using FDM, where it allocates more capacity to h1h4 pair. For latency, one packet can fit into the capacity of both, so two packets from different pairs can be sent simultaneously. Therefore, the latency is equal to the one in Q2. For throughput, even though the amount of data you can send in total is same as Q2, because h1h4 pair has a bigger capacity, so it can send more data in 30s. That's why it has a higher rate.

Three pairs: h1->h4 & h7->h9 & h8->h10
Prediction:
Latency = 420 ms, throughput = 0.85 mbps. It is almost the same reason as 2 pairs. Because we assume the network is using static multiplexing, then the throughput of three pairs should be the same, i.e. 1 third of the throughput in Q2, and the latency should be triple.

Actual:
H1h4 average RTT = 140.2 h7h9 average RTT = 140.3. h8h10 average RTT = 140.3
H1h4 measured throughput:
sent 57897 KB rate=1.9299 Mbps
received=57897 KB rate=1.7366147754881671 Mbps
h7h9 measured throughput:
sent 17057 KB rate=0.6498273423907423 Mbps
received=17057 KB rate=0.5021638649277239 Mbps
h8h10 measured throughput:
sent 5830 KB rate=0.19433333333333333 Mbps
received=5830 KB rate=0.18001605632063236 Mbps

Explanation:
This is also almost the same reason as 2 pairs. This network is using FDM, where it allocates more capacity to h1h4 pair and h7h9 pair gets fewer than that and the remaining goes to h8h10. For latency, one packet can fit into the capacity of 3 of these, so three packets from

different pairs can be sent simultaneously. Therefore, the latency is equal to the one in Q2. For throughput, even though the amount of data you can send in total is same as Q2, because h1h4 pair has a bigger capacity and h7h9 goes next, the throughput of h1h4 is highest and that of h8h10 is smallest.

Q4:
Prediction:

H1h4 latency = 140, throughput = 2.375311932849365.
Because from Q3, we now assume the network is using FDM, so the latency should just be equal to Q2. And because the two connections are sharing L2, and we assume L2 equally allocates capacity to both pair, the capacity of each pair should be 5.9626619 / 2 = 3. Then the bottleneck throughput for h1h4 pair is L1, which is 2. 375311932849365.

H5h6 latency = 80, throughput = 3
Same reason. But the bottleneck throughput for h5h6 pair is L2, which is 3.

Actual:
H1h4 average RTT = 140.2, h5h6 average RTT = 80.2. Our prediction is correct!
H1h4 measured throughput:
sent 75439 KB rate=2.5146333333333333 Mbps
received=75439 KB rate=2.1895570906135715 Mbps
h5h6 measured throughput:
sent 114919 KB rate=3.830633333333333 Mbps
received=114919 KB rate=3.470719700401679 Mbps

Our prediction for h1h4 is correct. For h5h6, possibly L2 does not equality allocates capacity to h1h4 and h5h6 by using FDM. It allocates a little more capacity to h5h6, so the actual throughput is greater than our prediction. But the allocation to h1h4 is still greater than capacity of L1. So L1 is still the bottleneck for h1h4, that's why our prediction is correct.