Computer Networks
Jacobs University Bremen
Dragi Kamov

Course: CO20-320301
February 22nd, 2019

Homework 1

Problem 1.1

Solution:

a) The calculations are calculated over an Ethernet cable on the campus network using the terminal on Ubuntu and running *ping -4 -c 10 website*.

Websites	IP address	Minimum time	Maximum time	Avarage time
amazon.com	176.32.98.166	103.748ms	105.272ms	104.328ms
www.amazon.com	104.111.232.172	10.950ms	11.905ms	11.467ms
www.jacobs-university.de	178.63.121.203	15.549ms	19.618ms	16.537ms
moodle.jacobs-university.de	212.201.46.73	1.328ms	3.567ms	2.208ms

The calculations were done using *ping* on Ubuntu 18.04 on 17th of February at 21:00. Also, I noticed that when I am trying to *ping www.amazon.com* I am measuring the round-trip times between a different server than *amazon.com* and their times are drastically different.

b) The calculations were done on the *mtr* utility v0.92 on 17th of February at 21:23.

Websites	Automous systems and hops		
amazon.com	AS680 x 3 hops	AS1299 x 6 hops	AS16509 x 1 hop
www.amazon.com	AS680 x 4 hops	AS16509 x 2 hops	
www.jacobs-university.de	AS680 x 3 hops	AS24940 x 3 hops	
moodle.jacobs-university.de	AS680 x 1 hop		

I noticed something interesting, for each website tracked I went through one autonomous system AS680, which belongs to the German National Research and Education Network.

Problem 1.2

Solution:

a)

ASN	Name of organization	Assigned by register
AS680	German National Research and Education Network	RIPE
AS1299	TELIANET	RIPE
AS16509	Amazon	ARIN
AS24940	HETZNER-AS	RIPE

b) The IPv6 prefix *2001:638:709::/48* is used by International University Bremen which is the old name of Jacobs University Bremen. The prefix is not globally announced, but *2001:638::/32* is globally announced and it belongs to the German National Research and Education Network.

Problem 1.3

Solution:

a)

[3] local 10.0.0.1 port 59344 connected with 10.0.0.2 port 5001			
[ID]	**Interval**	**Transfer**	**Bandwidth**
[3]	0.0-10.0 sec	11.2 MBytes	9.44 Mbits/sec
[3]	10.0-20.0 sec	10.9 MBytes	9.12 Mbits/sec
[3]	20.0-30.0 sec	10.6 MBytes	8.91 Mbits/sec
[3]	30.0-40.0 sec	10.8 MBytes	9.02 Mbits/sec
[3]	40.0-50.0 sec	11.0 MBytes	9.23 Mbits/sec
[3]	50.0-60.0 sec	10.6 MBytes	8.91 Mbits/sec
[3]	0.0-60.2 sec	65.1 MBytes	9.08 Mbits/sec

From the python script we could see that all of the links are limited to a bandwidth of 10mbps so seeing these bandwidths I would say that they are what I expected.

b) Pinging h2 from h1 while there is no *iperf* measurement in the background is causing a delay and that could be observed from the following results:

	Minimum time	**Maximum time**	**Avarage time**
During data rate:	10.898ms	22.301ms	15.322ms
While no *iperf* measurements:	0.023ms	0.072ms	0.049ms

According to me, we have a case of a transmission delay, the delay is caused because the link is so full that the ping needs to wait for space before pushing to the link.

Problem 1.4

Solution:

a) I ran *iperf* on h1 and h2 in the background and ran a ping between h3 and h4 and the *iperf* didn't impact the observed round-trip times as I observed after running the ping once more when the *iperf* wasn't running on h1 and h2. The round-trip times looked like the following:

Minimum time	**Maximum time**	**Avarage time**
0.024ms	0.489ms	0.133ms

b) While running both *iperf* measurements concurrently they don't impact each other. I ran a background *iperf* on h1 and h2 and another *iperf* shown in the terminal of h3 and h4. After it finished I ran *iperf* again on h3 and h4 where the results were similar. Results concurrently:

<table>
<tr><td colspan="6">[3] local 10.0.0.1 port 52072 connected with 10.0.0.2 port 5001</td></tr>
<tr><td colspan="6">[3] local 10.0.0.3 port 56102 connected with 10.0.0.4 port 5001</td></tr>
<tr><th>[ID]</th><th>Interval</th><th>Transfer h1 → h2</th><th>Bandwidth h1 → h2</th><th>Transfer h3 → h4</th><th>Bandwidth h3 → h4</th></tr>
<tr><td>[3]</td><td>0.0-10.0 sec</td><td>11.2 MBytes</td><td>9.44 Mbits/sec</td><td>11.5 MBytes</td><td>9.65 Mbits/sec</td></tr>
<tr><td>[3]</td><td>10.0-20.0 sec</td><td>11.2 MBytes</td><td>9.44 Mbits/sec</td><td>11.2 MBytes</td><td>9.44 Mbits/sec</td></tr>
<tr><td>[3]</td><td>20.0-30.0 sec</td><td>11.4 MBytes</td><td>9.54 Mbits/sec</td><td>11.2 MBytes</td><td>9.44 Mbits/sec</td></tr>
<tr><td>[3]</td><td>30.0-40.0 sec</td><td>11.1 MBytes</td><td>9.33 Mbits/sec</td><td>11.4 MBytes</td><td>9.54 Mbits/sec</td></tr>
<tr><td>[3]</td><td>40.0-50.0 sec</td><td>11.4 MBytes</td><td>9.54 Mbits/sec</td><td>11.1 MBytes</td><td>9.33 Mbits/sec</td></tr>
<tr><td>[3]</td><td>50.0-60.0 sec</td><td>11.2 MBytes</td><td>9.44 Mbits/sec</td><td>11.2 MBytes</td><td>9.44 Mbits/sec</td></tr>
<tr><td>[3]</td><td>0.0-60.1 sec</td><td>67.6 MBytes</td><td>9.44 Mbits/sec</td><td>67.8 MBytes</td><td>9.45 Mbits/sec</td></tr>
</table>

Problem 1.5

Solution:

a) If we were to run the two *iperf* measurements concurrently (h1 → h4 and h3 → h2) these are the results we would get:

<table>
<tr><td colspan="6">[3] local 10.0.0.1 port 43438 connected with 10.0.0.4 port 5001</td></tr>
<tr><td colspan="6">[3] local 10.0.0.3 port 57806 connected with 10.0.0.2 port 5001</td></tr>
<tr><th>[ID]</th><th>Interval</th><th>Transfer h1 → h4</th><th>Bandwidth h1 → h4</th><th>Transfer h3 → h2</th><th>Bandwidth h3 → h2</th></tr>
<tr><td>[3]</td><td>0.0-10.0 sec</td><td>9.25 MBytes</td><td>7.76 Mbits/sec</td><td>9.88 MBytes</td><td>8.28 Mbits/sec</td></tr>
<tr><td>[3]</td><td>10.0-20.0 sec</td><td>13.4 MBytes</td><td>11.2 Mbits/sec</td><td>12.0 MBytes</td><td>10.1 Mbits/sec</td></tr>
<tr><td>[3]</td><td>20.0-30.0 sec</td><td>13.6 MBytes</td><td>11.4 Mbits/sec</td><td>14.4 MBytes</td><td>12.1 Mbits/sec</td></tr>
<tr><td>[3]</td><td>30.0-40.0 sec</td><td>11.5 MBytes</td><td>9.65 Mbits/sec</td><td>12.0 MBytes</td><td>10.1 Mbits/sec</td></tr>
<tr><td>[3]</td><td>40.0-50.0 sec</td><td>15.2 MBytes</td><td>12.8 Mbits/sec</td><td>11.5 MBytes</td><td>9.65 Mbits/sec</td></tr>
<tr><td>[3]</td><td>50.0-60.0 sec</td><td>9.43 MBytes</td><td>7.91 Mbits/sec</td><td>13.9 MBytes</td><td>11.6 Mbits/sec</td></tr>
<tr><td>[3]</td><td>0.0-60.0 sec</td><td>72.3 MBytes</td><td>9.73 Mbits/sec</td><td>73.6 MBytes</td><td>9.62 Mbits/sec</td></tr>
</table>

I could notice here that I got bandwidths higher than the bandwidth limit which is set at 10.

And running the two *iperf* measurements concurrently (h1 → h3 and h2 → h4) these are the results we would get:

[3] local 10.0.0.1 port 39004 connected with 10.0.0.3 port 5001					
[3] local 10.0.0.2 port 53142 connected with 10.0.0.4 port 5001					
[ID]	**Interval**	**Transfer h1 → h3**	**Bandwidth h1 → h3**	**Transfer h2 → h4**	**Bandwidth h2 → h4**
[3]	0.0-10.0 sec	5.62 MBytes	4.72 Mbits/sec	6.88 MBytes	5.77 Mbits/sec
[3]	10.0-20.0 sec	9.62 MBytes	8.07 Mbits/sec	10.4 MBytes	8.75 Mbits/sec
[3]	20.0-30.0 sec	8.77 MBytes	7.35 Mbits/sec	5.53 MBytes	4.64 Mbits/sec
[3]	30.0-40.0 sec	4.97 MBytes	4.17 Mbits/sec	4.16 MBytes	3.49 Mbits/sec
[3]	40.0-50.0 sec	3.36 MBytes	2.81 Mbits/sec	3.17 MBytes	2.66 Mbits/sec
[3]	50.0-60.0 sec	2.75 MBytes	2.31 Mbits/sec	3.54 MBytes	2.97 Mbits/sec
[3]	0.0-60.0 sec	35.1 MBytes	4.75 Mbits/sec	33.7 MBytes	4.63 Mbits/sec

I could notice that when the two *iperf* measurements are being measured concurrently and they are measuring in the opposite directions the bandwidths are higher, but when the two *iperf* are measured in the same direction the bandwidths are slower.

b) Running the two *iperf* measurements concurrently (h1 → h4 and h3 → h6) these are the results we would get:

[3] local 10.0.0.1 port 43628 connected with 10.0.0.4 port 5001					
[3] local 10.0.0.3 port 36062 connected with 10.0.0.6 port 5001					
[ID]	**Interval**	**Transfer h1 → h4**	**Bandwidth h1 → h4**	**Transfer h3 → h6**	**Bandwidth h3 → h6**
[3]	0.0-10.0 sec	8.12 MBytes	6.82 Mbits/sec	7.00 MBytes	5.87 Mbits/sec
[3]	10.0-20.0 sec	11.1 MBytes	9.33 Mbits/sec	4.00 MBytes	3.36 Mbits/sec
[3]	20.0-30.0 sec	10.2 MBytes	8.60 Mbits/sec	8.38 MBytes	7.03 Mbits/sec
[3]	30.0-40.0 sec	9.62 MBytes	8.07 Mbits/sec	6.50 MBytes	5.45 Mbits/sec
[3]	40.0-50.0 sec	8.38 MBytes	7.03 Mbits/sec	5.88 MBytes	4.93 Mbits/sec
[3]	50.0-60.0 sec	9.00 MBytes	7.55 Mbits/sec	6.75 MBytes	5.66 Mbits/sec
[3]	0.0-60.0 sec	56.5 MBytes	7.88 Mbits/sec	38.5 MBytes	5.37 Mbits/sec

The two *iperf* measurements are not influenced by the fact that they are being ran concurrently, but we can notice a bandwidth reduction in h3 → h6, and I believe that is because we have a 5% loss on one of the links that h3 → h6 are using, and those 5% of packages that are lost are being re-transmitted again.