

NASA 2016 HW2 NA Part

B03902086 李鈺昇

1.

References:

<http://searchnetworking.techtarget.com/definition/10-Gigabit-Ethernet>

<http://www.cs.nthu.edu.tw/~nfhuang/chap04.htm>

<http://www.networking-forum.com/viewtopic.php?t=23208>

First, the network will be pretty slow if the load is too high, causing frame collisions to happen quite often, and thus decreases the throughput. Next, CSMA/CD is not necessary for star network topology, while old types of Internet using bus topology needed it. Finally, since 10GbE disables half-duplex and only uses full-duplex, there is no need to use CSMA/CD to detect collision.

2.

References:

<http://oilcut123.pixnet.net/blog/post/354490151-%5B%E6%95%B4%E7%90%86%5D-hi>

http://www.codealias.info/technotes/the_hidden_and_exposed_terminal_problem_in_wireless_csma/ca

https://en.wikipedia.org/wiki/Exposed_node_problem

https://en.wikipedia.org/wiki/IEEE_802.11_RTS/CTS

(a)

The hidden terminal problem happens when two or more nodes, say A and B, are to send data to another third node C, but A and B might know nothing about each other, causing collisions.

Another problem, the exposed terminal problem, happens when some node, say A, heard that another node B, is about to send data, and thus A postpones its own data, even if B isn't sending to A and won't affect A.

(b)

In order to solve these two problems, RTS/CTS is used. For the first problem, if A is going to send data to C, then it first sends an RTS packet to C, and then C sends back a CTS packet to A. After this, C doesn't accept any other node until A finishes its task, avoiding this type of collision.

For the second problem, A listens if B sends an RTS packet. But it seems that this doesn't completely solve this problem.

3.

References:

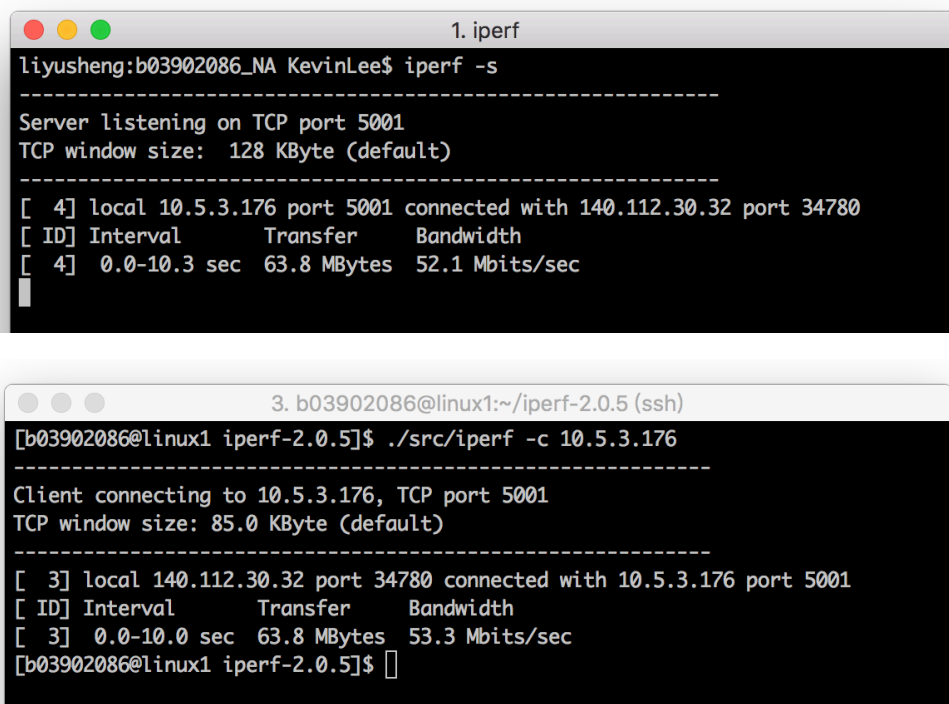
<https://prasadlinuxblog.wordpress.com/2012/08/15/step-by-step-guide-installation-of-iperf-benchmarking-tool/>

I installed iperf by the below commands:

```
wget http://sourceforge.net/projects/iperf/files/iperf-2.0.5.tar.gz
tar zxvf iperf-2.0.5.tar.gz
cd iperf-2.0.5/
./configure
make
make install
```

Surely the last step fails since I don't have the right privilege on the workstation. But I can still run the iperf program.

The media I used for this is WiFi ethernet (en0). The results are as follows:



The image contains two terminal window screenshots. The top window, titled '1. iperf', shows a server running 'iperf -s' on port 5001. It receives a connection from 140.112.30.32 on port 34780. The output shows a transfer of 63.8 MBytes over 0.0-10.3 seconds at a bandwidth of 52.1 Mbits/sec. The bottom window, titled '3. b03902086@linux1:~/iperf-2.0.5 (ssh)', shows a client running './src/iperf -c 10.5.3.176'. It connects to 10.5.3.176 on port 5001. The output shows a transfer of 63.8 MBytes over 0.0-10.0 seconds at a bandwidth of 53.3 Mbits/sec.

```
1. iperf
liyusheng:b03902086_NA KevinLee$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 4] local 10.5.3.176 port 5001 connected with 140.112.30.32 port 34780
[ ID] Interval      Transfer    Bandwidth
[ 4]  0.0-10.3 sec  63.8 MBytes 52.1 Mbits/sec

3. b03902086@linux1:~/iperf-2.0.5 (ssh)
[b03902086@linux1 iperf-2.0.5]$ ./src/iperf -c 10.5.3.176
-----
Client connecting to 10.5.3.176, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[ 3] local 140.112.30.32 port 34780 connected with 10.5.3.176 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  63.8 MBytes 53.3 Mbits/sec
[b03902086@linux1 iperf-2.0.5]$
```

This reminds me of the in-class test in R204, when the bandwidth is quite fast, even up to many Gbits/sec! And I think it's due to the difference between wireless WiFi and the wired net, or maybe the local host.