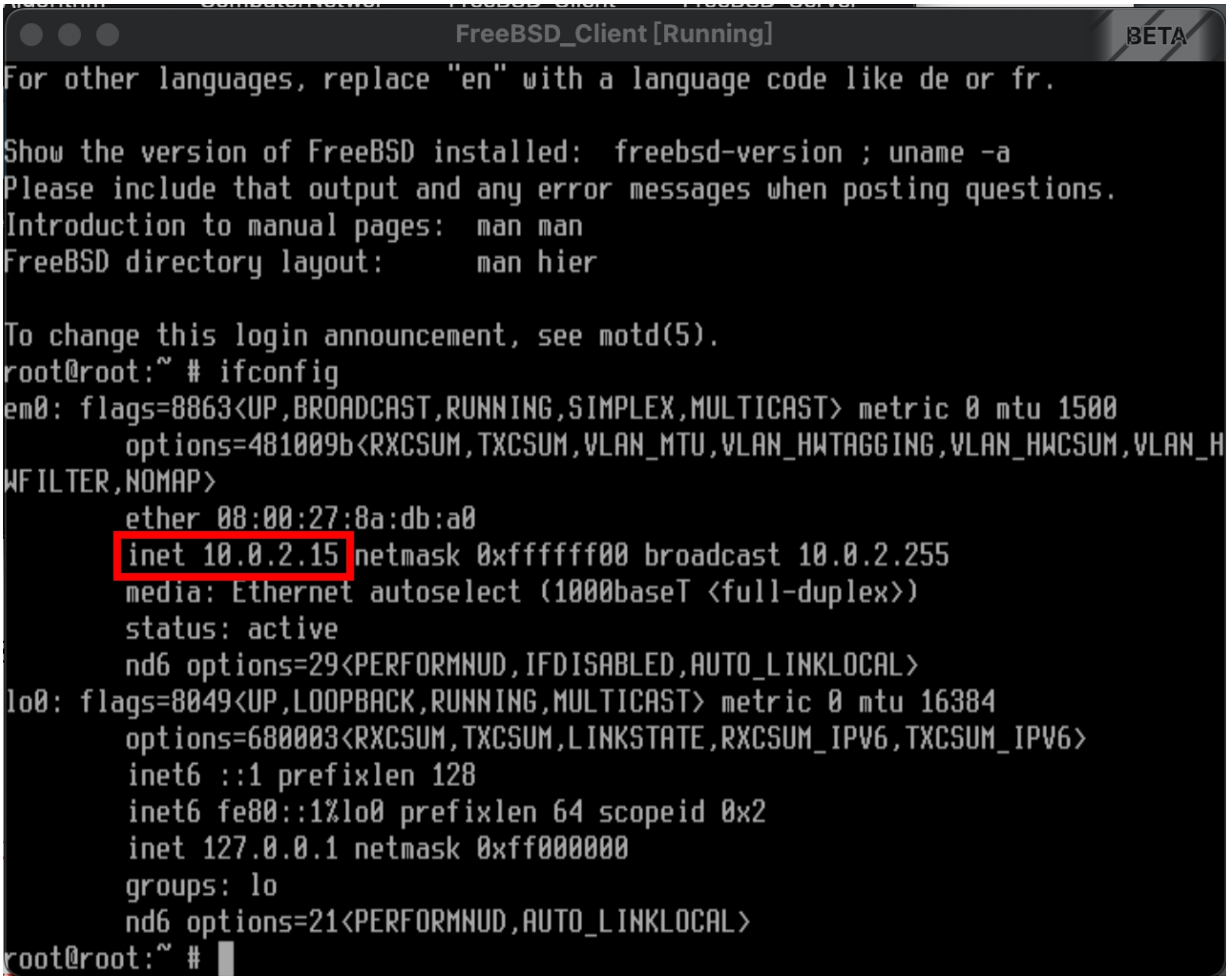
Assignment2 Report

Part1: Environment Setup

- Problems

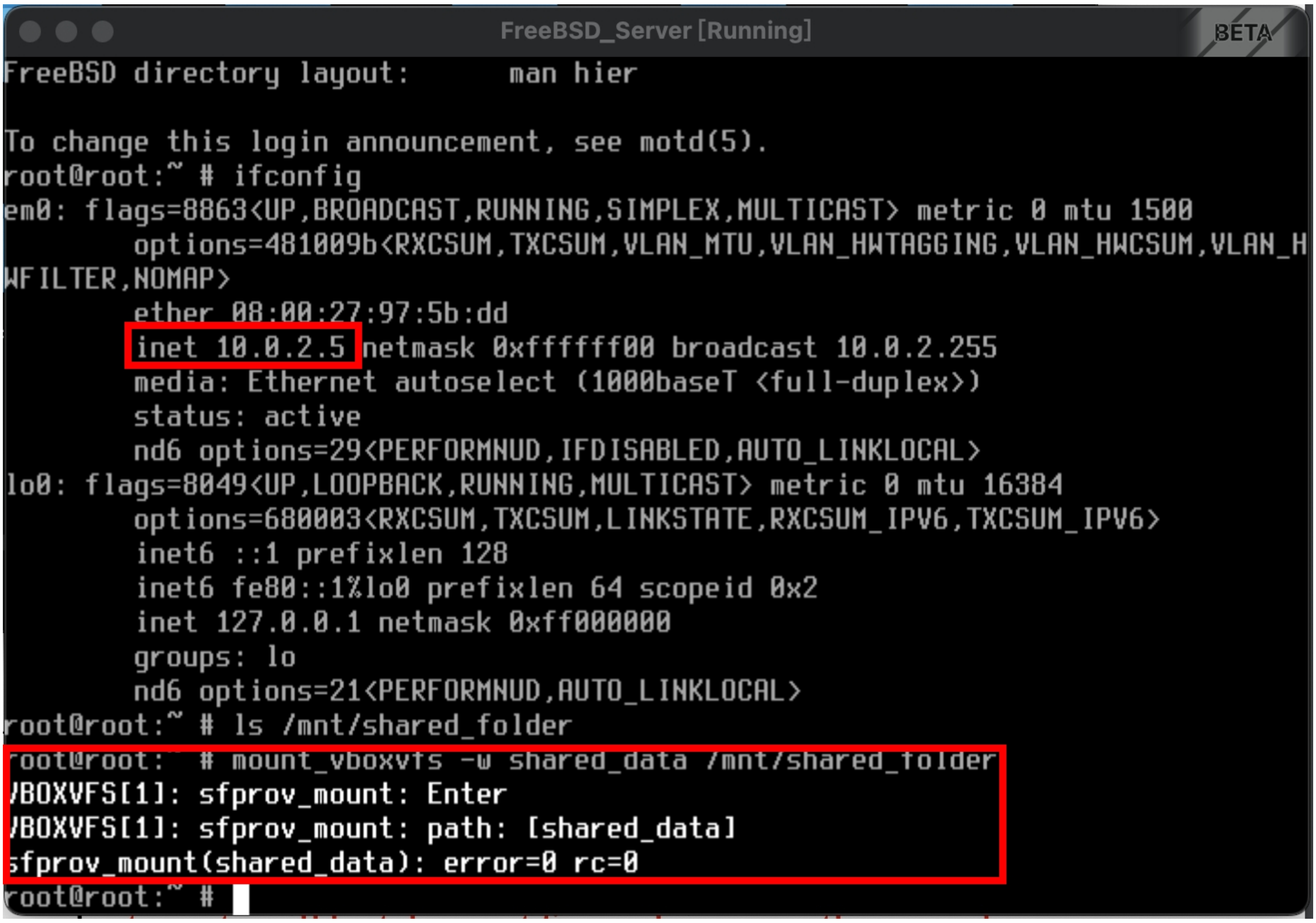
- Since my computer is M1 chip, I faced some network problems in part1 when I was setting the network config between client and server. It seems like the IP setting didn't apply to my VMs and two VMs cannot connect with each other. So I went to TA Time and the TAs assisted me to set the IP of both client and server manually.

- Client IP



```
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

To change this login announcement, see motd(5).
root@root:~ # ifconfig
em0: flags=8863<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=481009b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,VLAN_H
WFILTER,NOMAP>
        ether 08:00:27:8a:db:a0
        inet 10.0.2.15 netmask 0xffffff00 broadcast 10.0.2.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
        inet 127.0.0.1 netmask 0xff000000
        groups: lo
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@root:~ #
```

- Server IP & The result of mounting



```
FreeBSD directory layout:      man hier

To change this login announcement, see motd(5).
root@root:~ # ifconfig
em0: flags=8863<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=481009b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,VLAN_H
WFILTER,NOMAP>
        ether 08:00:27:97:5b:dd
        inet 10.0.2.5 netmask 0xffffff00 broadcast 10.0.2.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
        inet 127.0.0.1 netmask 0xff000000
        groups: lo
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@root:~ # ls /mnt/shared_folder
root@root:~ # mount_vboxvfs -w shared_data /mnt/shared_folder
VBOXVFS[1]: sfprov_mount: Enter
VBOXVFS[1]: sfprov_mount: path: [shared_data]
sfprov_mount(shared_data): error=0 rc=0
root@root:~ #
```

Part 2: TCP Data Transmission

- Problems

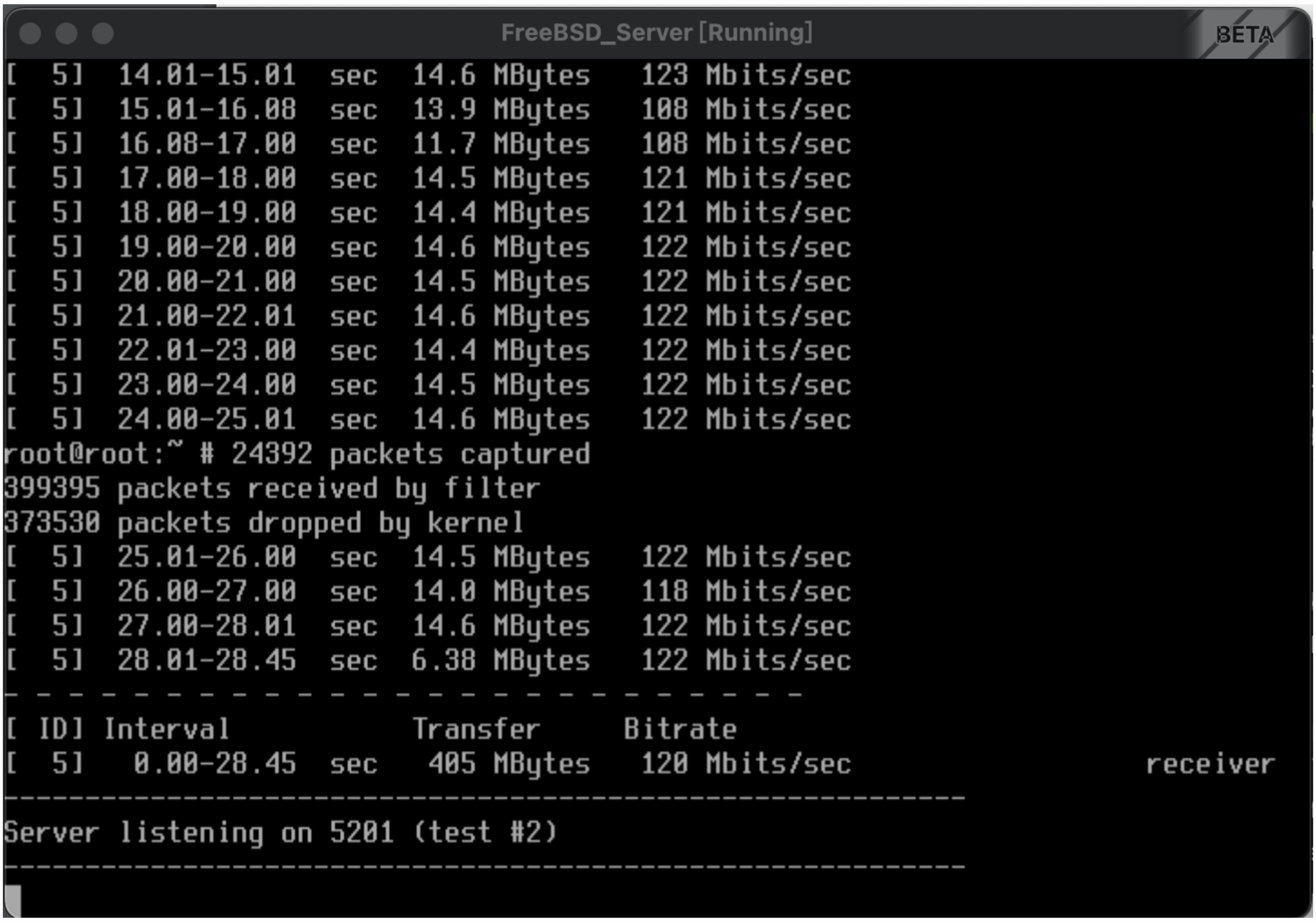
- Since I didn’t use any network monitoring tools before this course, I was unfamiliar with iperf, tcpdump and wireshark in the beginning. After searching some useful websites, I realized the meaning of each command and the parameters.

- Command

```
iperf3 -c 10.0.2.5 -t 30 -i 1 -w 1M -C newreno
```

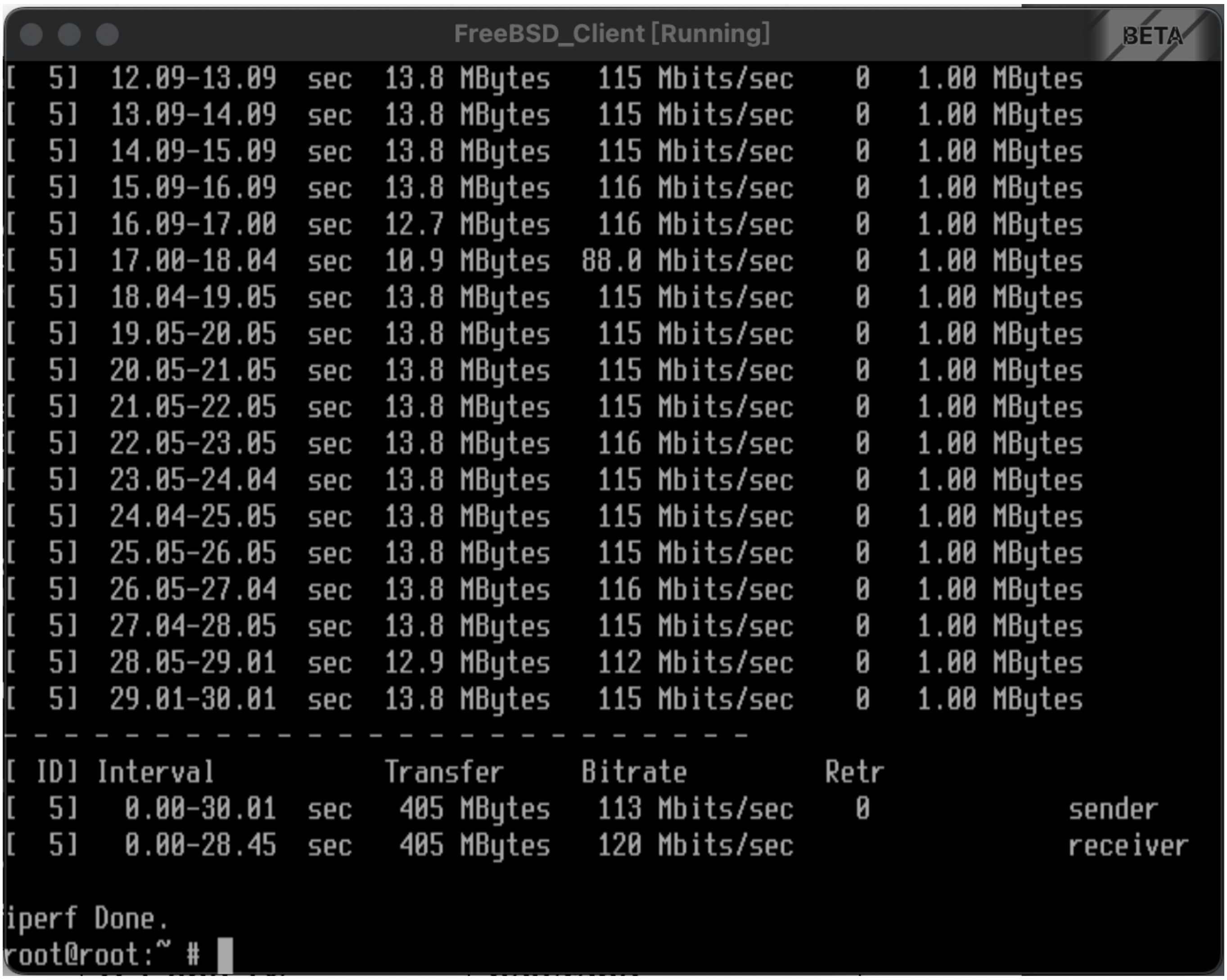
- `-c 10.0.2.5`: start an iperf client which is going to send data to 10.0.2.5 (server).
- `-t 30`: transmits the data for 30 seconds.
- `-i 1`: enable periodic bandwidth, jitter, and loss reports for each second.

- Server Result



```
[  5]  14.01-15.01  sec  14.6 MBytes   123 Mbits/sec
[  5]  15.01-16.08  sec  13.9 MBytes   108 Mbits/sec
[  5]  16.08-17.00  sec  11.7 MBytes   108 Mbits/sec
[  5]  17.00-18.00  sec  14.5 MBytes   121 Mbits/sec
[  5]  18.00-19.00  sec  14.4 MBytes   121 Mbits/sec
[  5]  19.00-20.00  sec  14.6 MBytes   122 Mbits/sec
[  5]  20.00-21.00  sec  14.5 MBytes   122 Mbits/sec
[  5]  21.00-22.01  sec  14.6 MBytes   122 Mbits/sec
[  5]  22.01-23.00  sec  14.4 MBytes   122 Mbits/sec
[  5]  23.00-24.00  sec  14.5 MBytes   122 Mbits/sec
[  5]  24.00-25.01  sec  14.6 MBytes   122 Mbits/sec
root@root:~ # 24392 packets captured
399395 packets received by filter
373530 packets dropped by kernel
[  5]  25.01-26.00  sec  14.5 MBytes   122 Mbits/sec
[  5]  26.00-27.00  sec  14.0 MBytes   118 Mbits/sec
[  5]  27.00-28.01  sec  14.6 MBytes   122 Mbits/sec
[  5]  28.01-28.45  sec  6.38 MBytes   122 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-28.45  sec   405 MBytes   120 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201 (test #2)
-----------------------------------------------------------
```

- Client Result



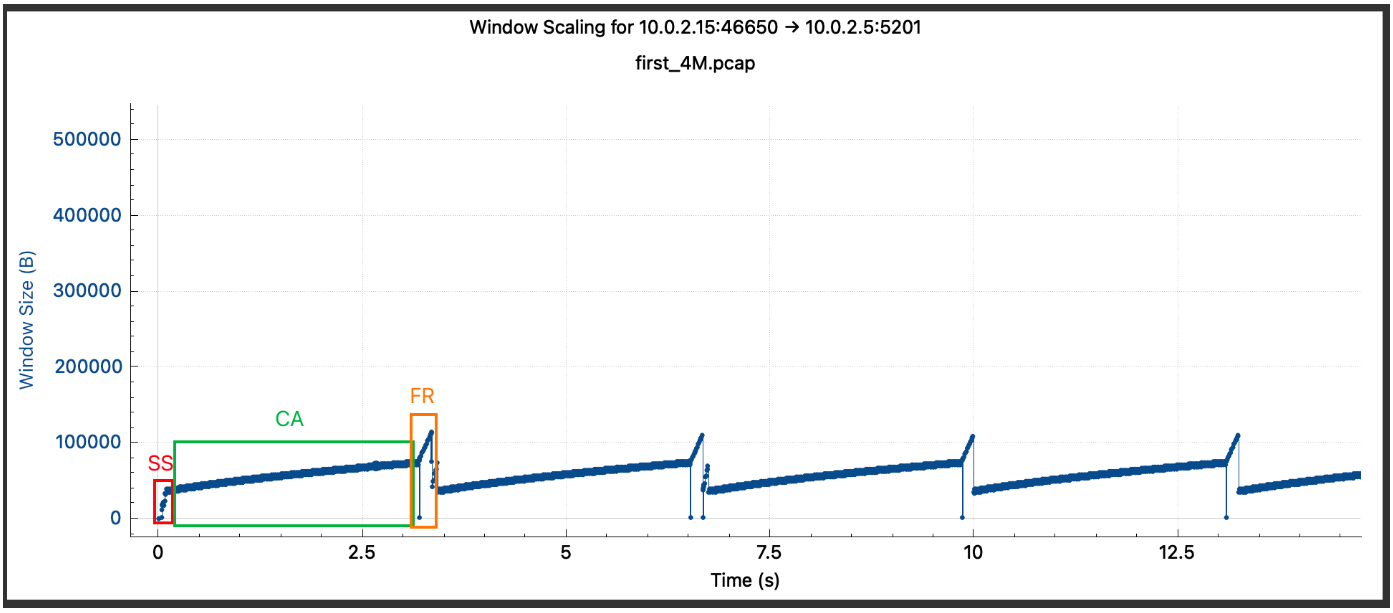
```
[  5]  12.09-13.09  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  13.09-14.09  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  14.09-15.09  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  15.09-16.09  sec  13.8 MBytes   116 Mbits/sec    0   1.00 MBytes
[  5]  16.09-17.00  sec  12.7 MBytes   116 Mbits/sec    0   1.00 MBytes
[  5]  17.00-18.04  sec  10.9 MBytes  88.0 Mbits/sec    0   1.00 MBytes
[  5]  18.04-19.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  19.05-20.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  20.05-21.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  21.05-22.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  22.05-23.05  sec  13.8 MBytes   116 Mbits/sec    0   1.00 MBytes
[  5]  23.05-24.04  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  24.04-25.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  25.05-26.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  26.05-27.04  sec  13.8 MBytes   116 Mbits/sec    0   1.00 MBytes
[  5]  27.04-28.05  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
[  5]  28.05-29.01  sec  12.9 MBytes   112 Mbits/sec    0   1.00 MBytes
[  5]  29.01-30.01  sec  13.8 MBytes   115 Mbits/sec    0   1.00 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-30.01  sec   405 MBytes   113 Mbits/sec    0             sender
[  5]   0.00-28.45  sec   405 MBytes   120 Mbits/sec                  receiver

iperf Done.
root@root:~ #
```

Part3: TCP Congestion Control

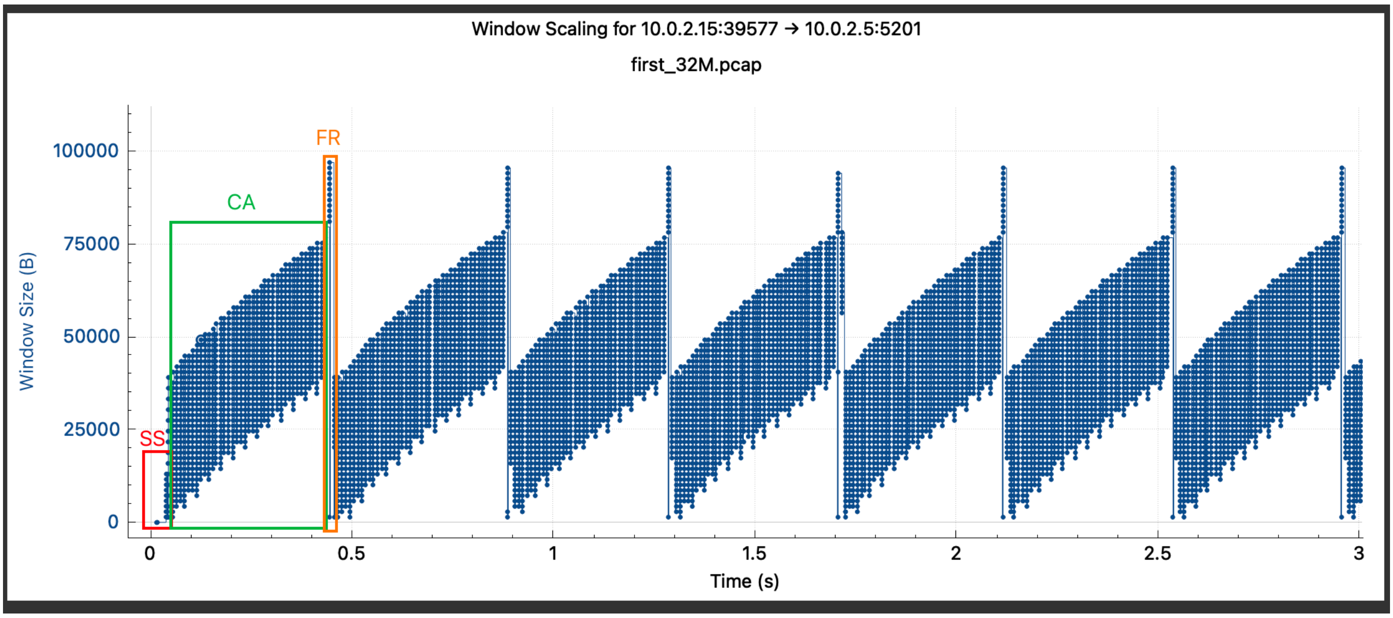
- **Red(SS):** Slow Start, which grows exponentially in ideal and ends when `cwnd` reach the `ssthresh`. `(cwnd < ssthresh)`
- **Green(CA):** Congestion Avoidance, which grows linearly right after the Slow Start in order to avoid the congestion and ends when any congestion being detected. `(cwnd > ssthresh)`
- **Orange(FR):** Fast Recovery, entered after receiving what seems to be the missing segment sent during Fast-Retransmit.

Scenario1: (Bandwidth)

- 4 Mbit/s



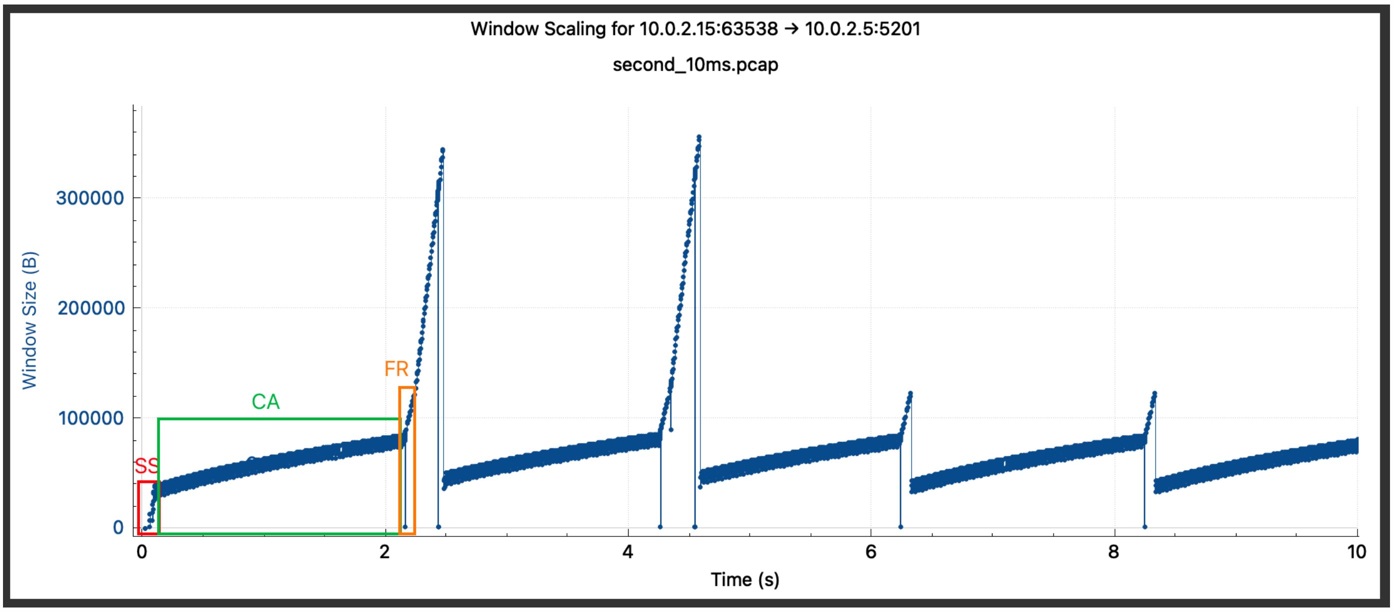
- 32 Mbit/s



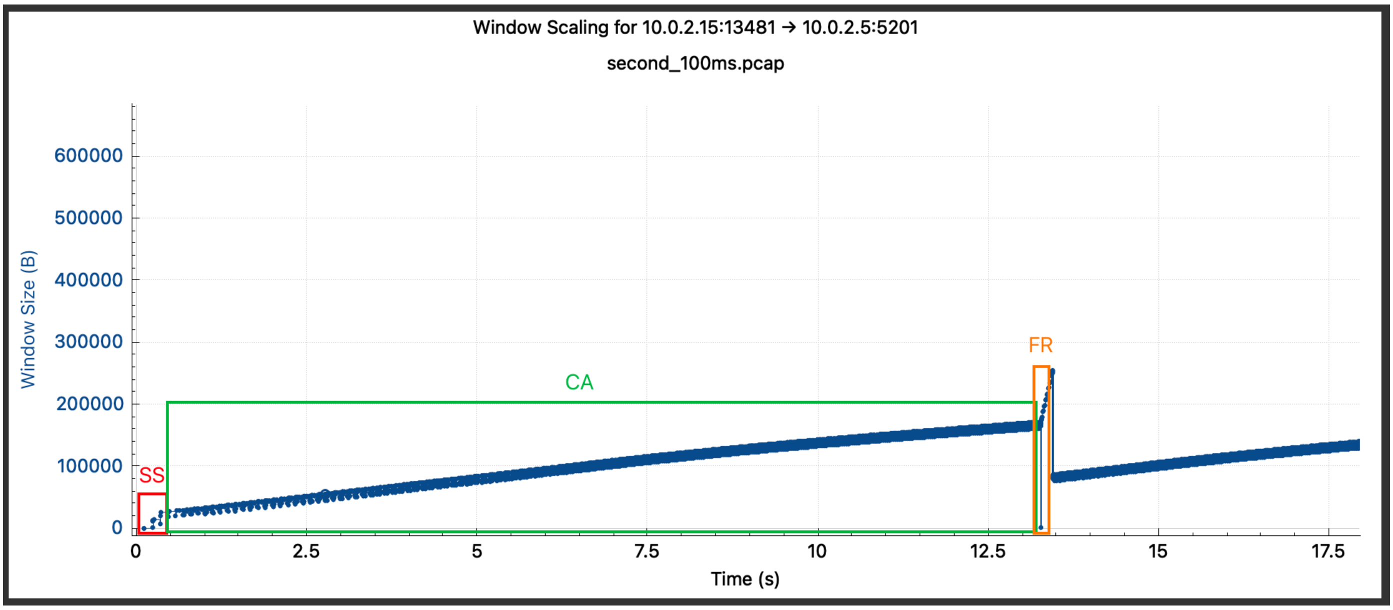
- Since the bandwidth is different between these two cases, the Congestion Avoidance period of `4 Mbit/s` is obviously longer than the Congestion Avoidance period of `32 Mbit/s` as the figure shown.
- In my opnion, I guess the smaller bandwidth will take longer to detect the congestion.

Scenario2: (Delay)

- 10 ms



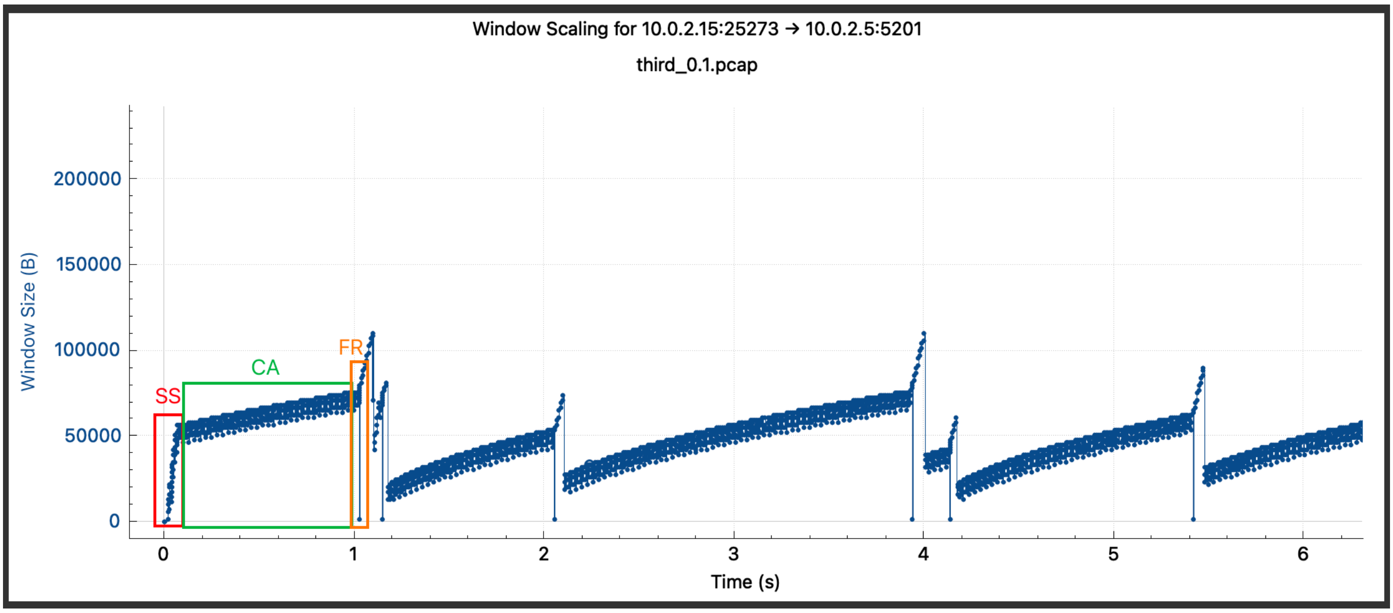
- 100 ms



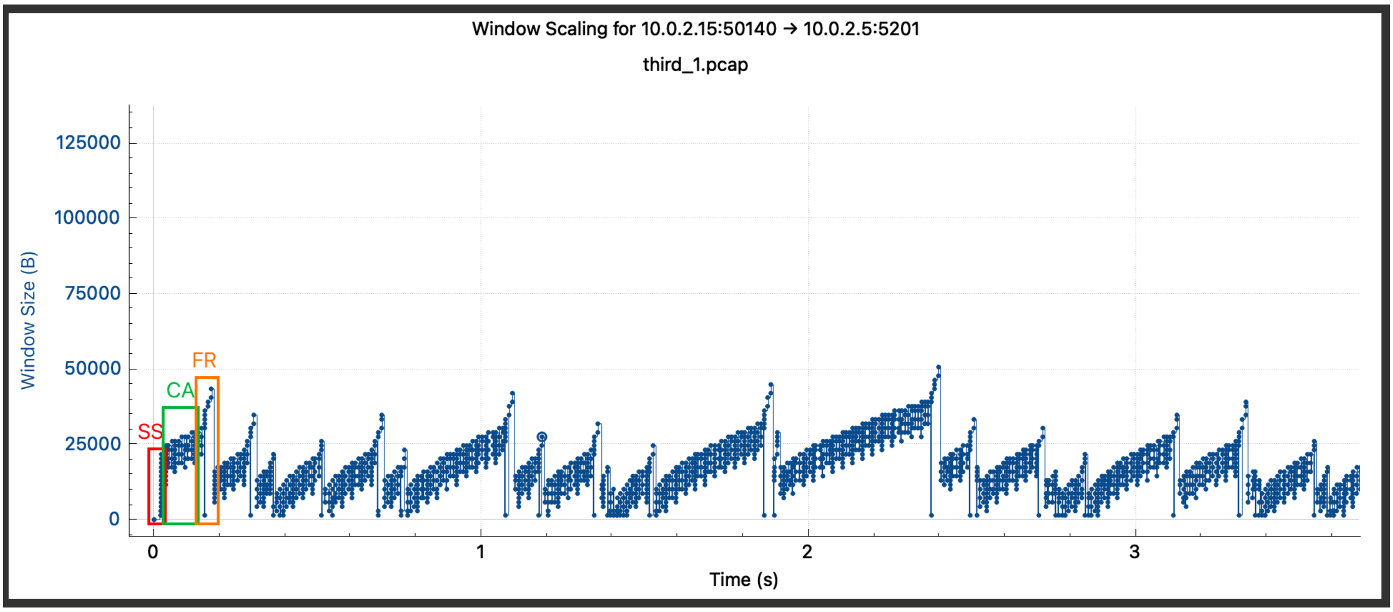
- Due to the difference of delay in these two cases, we know that the period of Congestion Avoidance will take much longer if we increase the delay time.
- Besides, the congestion window size will also be larger if we have larger delay.

Scenario3: (Packet Loss Rate)

- 0.1 %



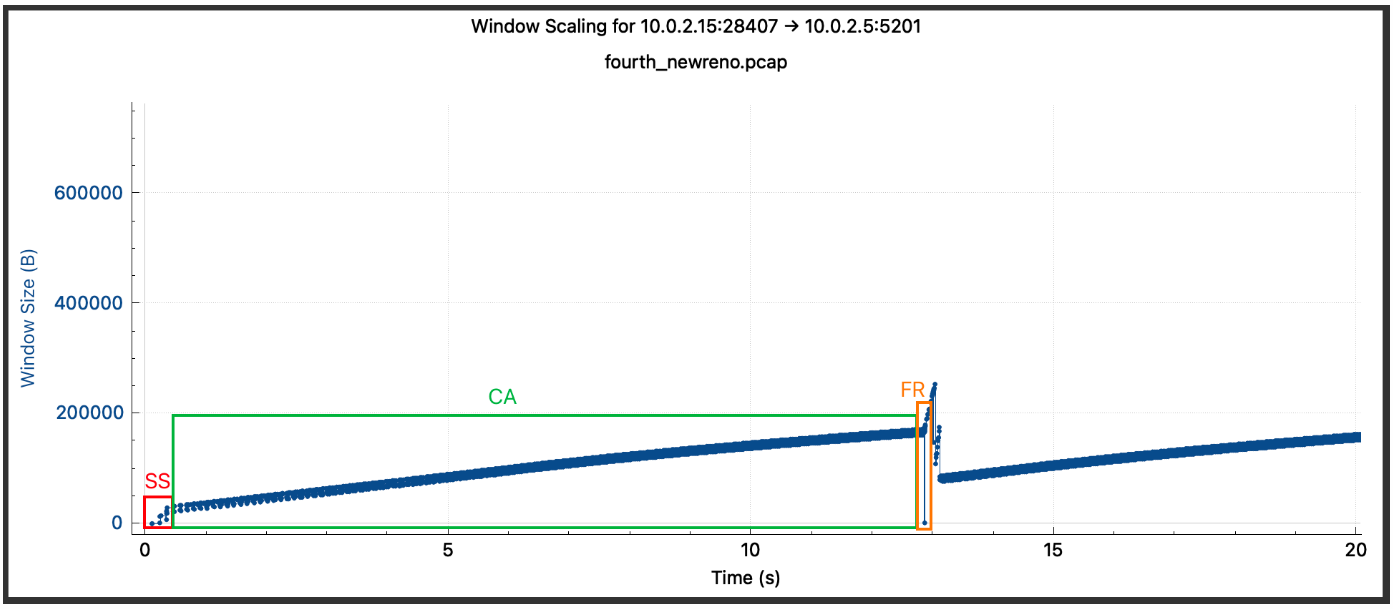
- 1 %



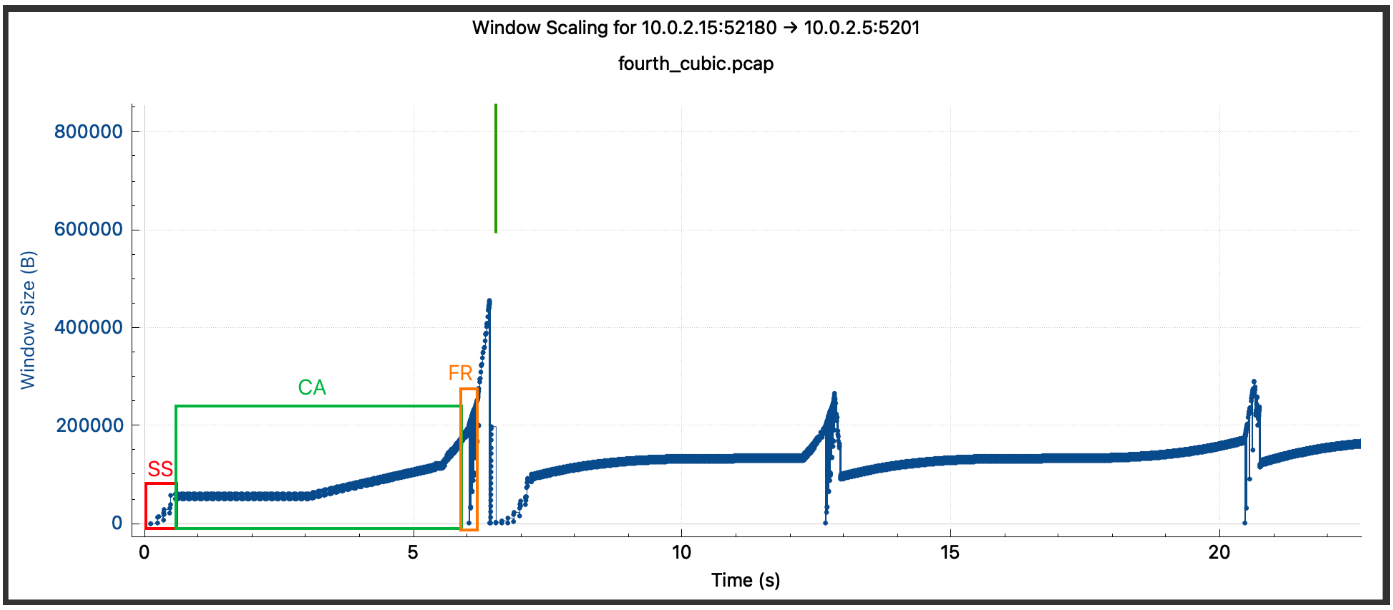
- Since the packet loss rate is different between these two cases, the congestion window size and the frequency of transmission may have little differences.
- In my opnion, I guess the case with higher packet loss rate (`1%`) will re-transmit the data more often, so the frequency is higher than the other one (`0.1%`).

Scenario4: (Different Algorithm with 100ms Delay)

- newreno



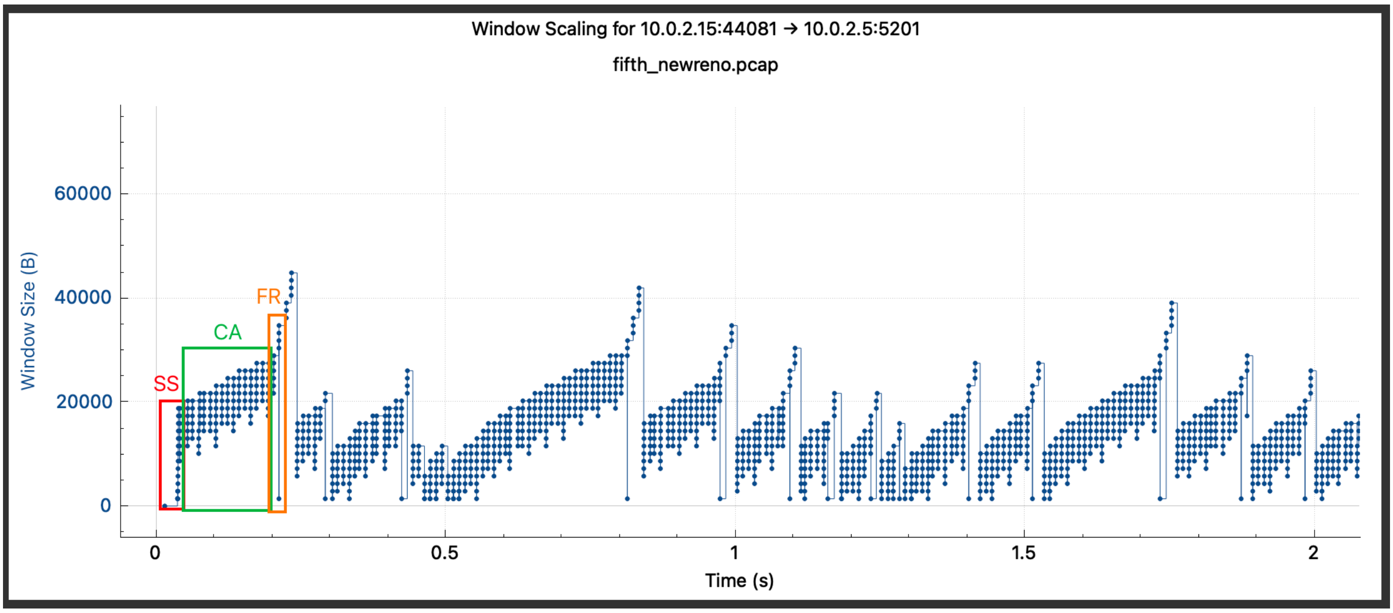
- cubic



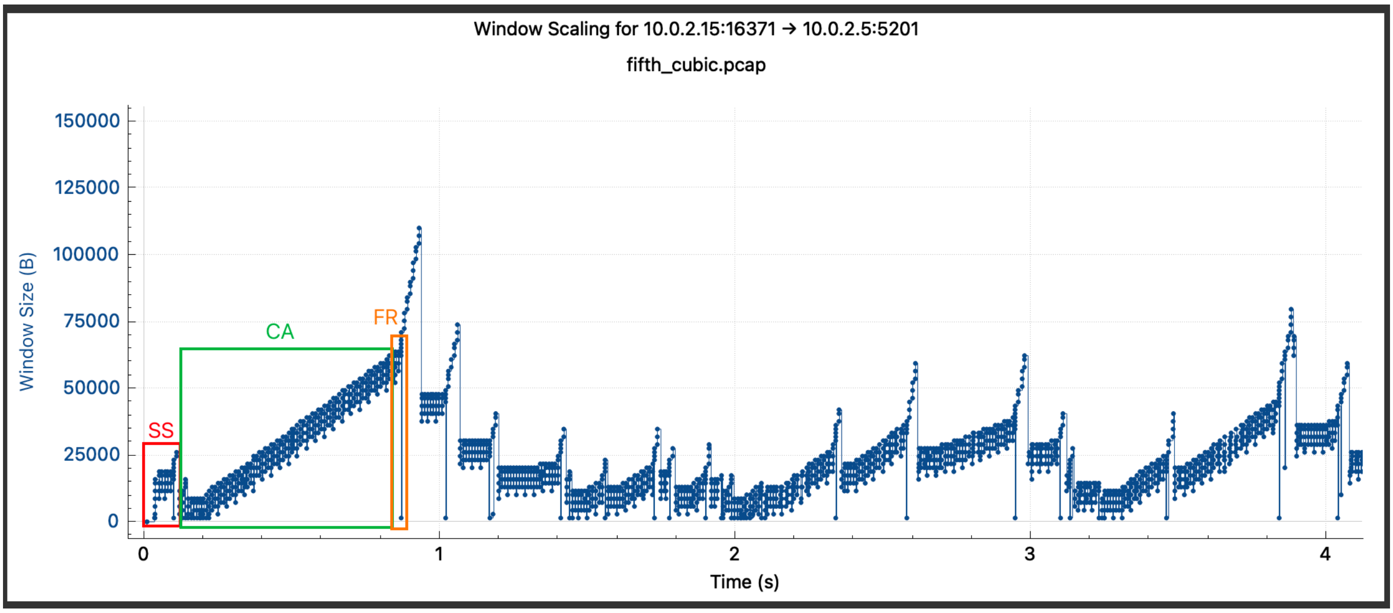
- In the same delay time, we can observe that the period of Congestion Avoidance with `newreno` is obviously larger than the one with `cubic`.
- Since the congestion window size increasing function is different between these two algorithm, the Congestion Avoidance length may be different.

Scenario5: (Different Algorithm with 1% Packet Loss Rate)

- newreno



- cubic



- In the same Packet Loss Rate, we can observe that the period and the increasing speed of Congestion Avoidance is quite different.
- In my opnion, I think the reason which makes the period different is the difference of congestion window size increasing function between `cubic` and `newreno`.