

TCP/IP performance test 2

y.ma@riken.jp

20170517

Outline

- ❖ Motivation:
 - ❖ Measure the real performance and use it as P.D.F. input for more integrated study
- ❖ Setup: (more details in slides)
 - ❖ Software: mini-DAQ
 - ❖ Hardware: direct 10gbps connection
 - ❖ Hardware: Cisco 6120XP
- ❖ Results
- ❖ Summary & todo

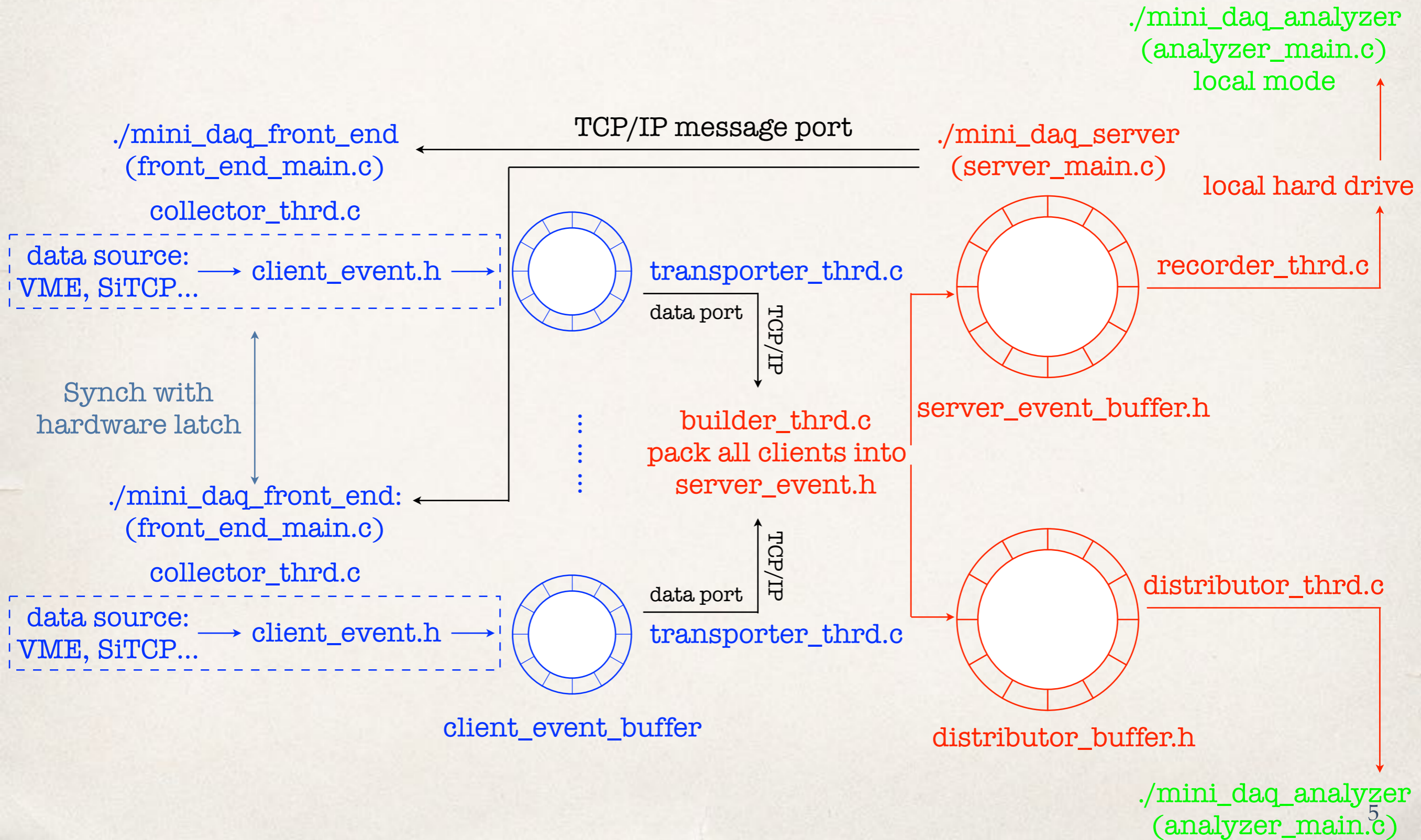
Motivation

- ❖ Sophisticated simulation for TCP/IP performance takes months to simulate a few seconds of Alice data
- ❖ Reliability of simulation is always an open question
- ❖ A direct measurement is indispensable
- ❖ Results from direct measurement can be used a probability distribution function (P.D.F.) for a more integrated simulation

Software: a minimalism DAQ

- ❖ A tiny DAQ package with basic functionalities for data taking in physics experiment
- ❖ programmed by Y. Ma from scratch with pure C and Linux system calls
- ❖ slightly overhead caused by data header filtering and so on shows real TCP/IP performance for DAQ task

Software: a minimalism DAQ



Software: default CentOS7 configuration

- ❖ Default CentOS 7.0 x86_64:

- ❖ [oper@e50_server0 ~]\$ cat /proc/sys/net/ipv4/tcp_rmem

- ❖ 4096 (min [Byte]) 87380(default [Byte]) 6291456(max [Byte])

- ❖ [oper@e50_server0 ~]\$ cat /proc/sys/net/ipv4/tcp_wmem

- ❖ 4096(min [Byte]) 16384(default [Byte]) 4194304(max [Byte])

- ❖ mini-DAQ:

- ❖ minimum DAQ functionality from scratch in C

- ❖ data source (ring buffer) —> *TCP/IP socket* —> data sink (ring buffer)



default C library function,
no customization

Hardware: server & switch



Two sets of Alice O2 compatible server;
40 threads / server;
4 ports of 10Gbps, X710, Intel;
2 ports of 40 Gbps, MT27700, Mellanox
Purchased with E50 budget of Prof. Noumi

Cisco managed switch 6120XP;
RIKEN blade server controller;
Picked up by Itahashi-san before recycle...

Hardware: Cisco 6120XP configuration

- ❖ Preparation: RJ45-DB9 serial cable; install minicom; Ctrl+A Z to select /dev/ttyS0 as serial device
- ❖ Turn on Cisco 6120XP power; Ctrl+l to choose boot image; loader > boot /installables/switch/ucs-6100-k9-kickstart.5.2.3.N2.2.22c.bin; Fabric(boot)# config terminal; Fabric(boot)(config)# **admin-password***password*; Exit config terminal mode and return to the boot prompt; Boot the system firmware version on the fabric interconnect: Fabric(boot)# **load** /installables/switch/ucs-6100-k9-system.5.2.3.n(?)
- ❖ set password as: himitsu-301
- ❖ from serial terminal (minicom), enter /system manual; scope fabric-interconnect a; show detail; obtain management port IP address



```
ucs-2811-A /system # console buffer
ucs-2811-A /system #
ucs-2811-A /system #
ucs-2811-A /system #
ucs-2811-A /system # console
ucs-2811-A /system #
ucs-2811-A /system #
ucs-2811-A /system # scope fabric-interconnect
ucs-2811-A /system #
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a show detail
ucs-2811-A /system # scope fabric-interconnect a show detail
Fabric Interconnect:
ID: A
Product Name: Cisco UCS 6120XP
PID: 838-54188
Vendor: Cisco Systems, Inc.
Serial ID: 55113378730
Management IP: 172.27.42.234
OOB Network: 255.255.255.0
OOB IPv6 Address: ::
OOB IPv6 Gateway: ::
Prefix: 64
Operability: Operable
Thermal Status: OK
Current Task 1:
Current Task 2:
Current Task 3:
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
ucs-2811-A /system # scope fabric-interconnect a
```


Hardware: Cisco 6120XP configuration

- ❖ from serial terminal (minicom): enter /system manual; scope fabric -interconnect a; show detail; to obtain management port IP address
- ❖ configure Fujitsu laptop with proper IP and mask to access Cisco management port
- ❖ Add certificate exception to java data base

```
e50-A /system # scope fabric-interconnect a
e50-A /fabric-interconnect #
e50-A /fabric-interconnect #
e50-A /fabric-interconnect #
e50-A /fabric-interconnect # show detail

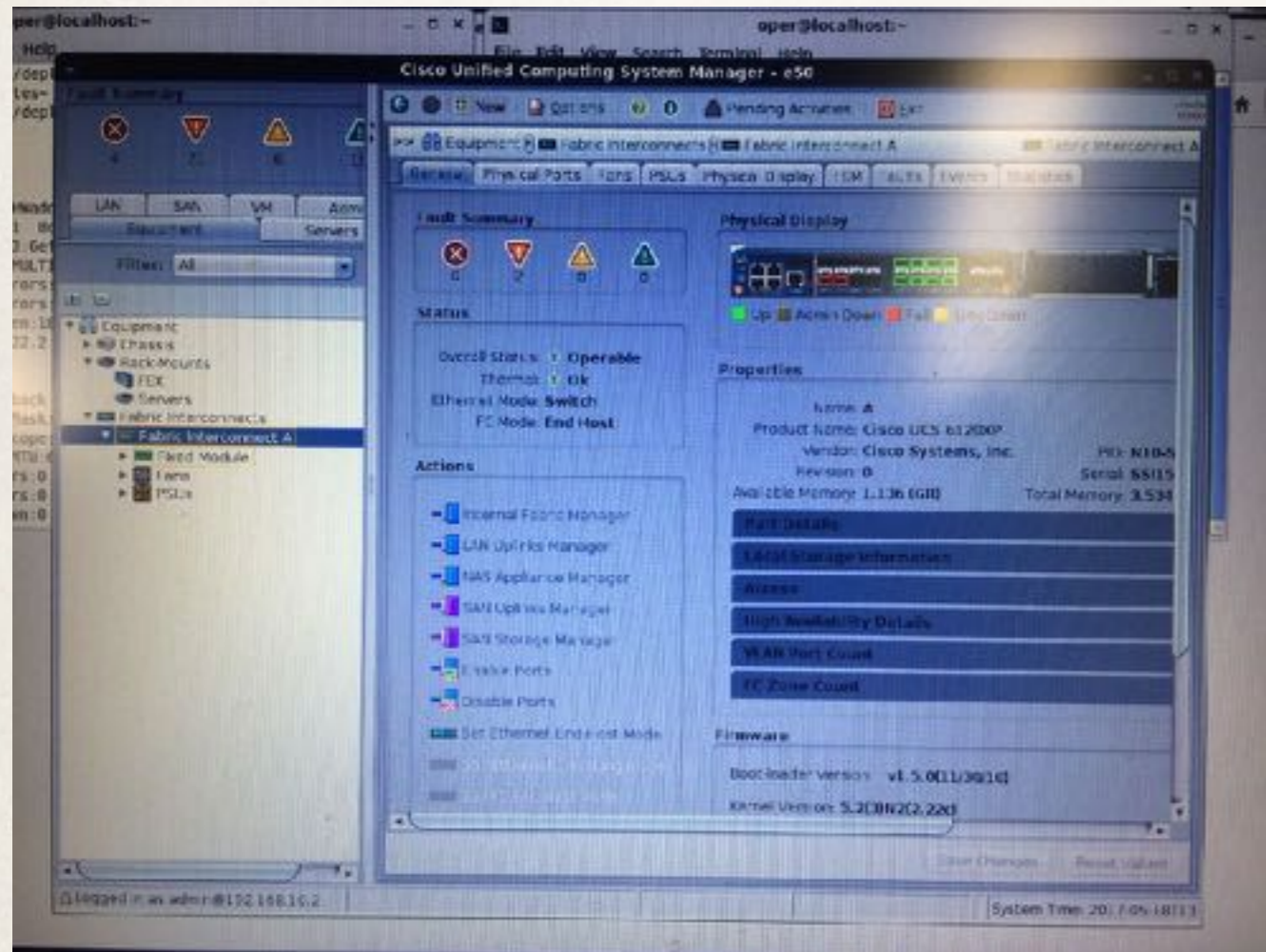
Fabric Interconnect:
  ID: A
  Product Name: Cisco UCS 6120XP
  PID: N10-S6100
  VID: V01
  Vendor: Cisco Systems, Inc.
  Serial (SN): SSI15370F36
  HW Revision: 0
  Total Memory (MB): 3534
  00B IP Addr: 172.27.42.155
  00B Gateway: 172.27.42.254
  00B Netmask: 255.255.255.0
```

```
DEVICE=eth0
HWADDR=8C:73:6E:7A:BC:F4
ONBOOT="yes"
#following config for hosting Kalliope
BOOTPROTO="none"
IPADDR=192.168.10.1
#IPADDR=172.27.42.150
NETMASK=255.255.255.0
TYPE="Ethernet"
USERCTL="no"
IPV6INIT="no"
#GATEWAY=192.168.10.1
/etc/sysconfig/network-scripts/ifcfg-eth0 (END)
```

```
[oper@localhost ~]$ emacs .java/deployment/security/
exception.sites      exception.sites~  trusted.jssecerts
[oper@localhost ~]$ emacs .java/deployment/security/exception.sites
```

Hardware: Cisco 6120XP configuration

❖ Finally, it works!

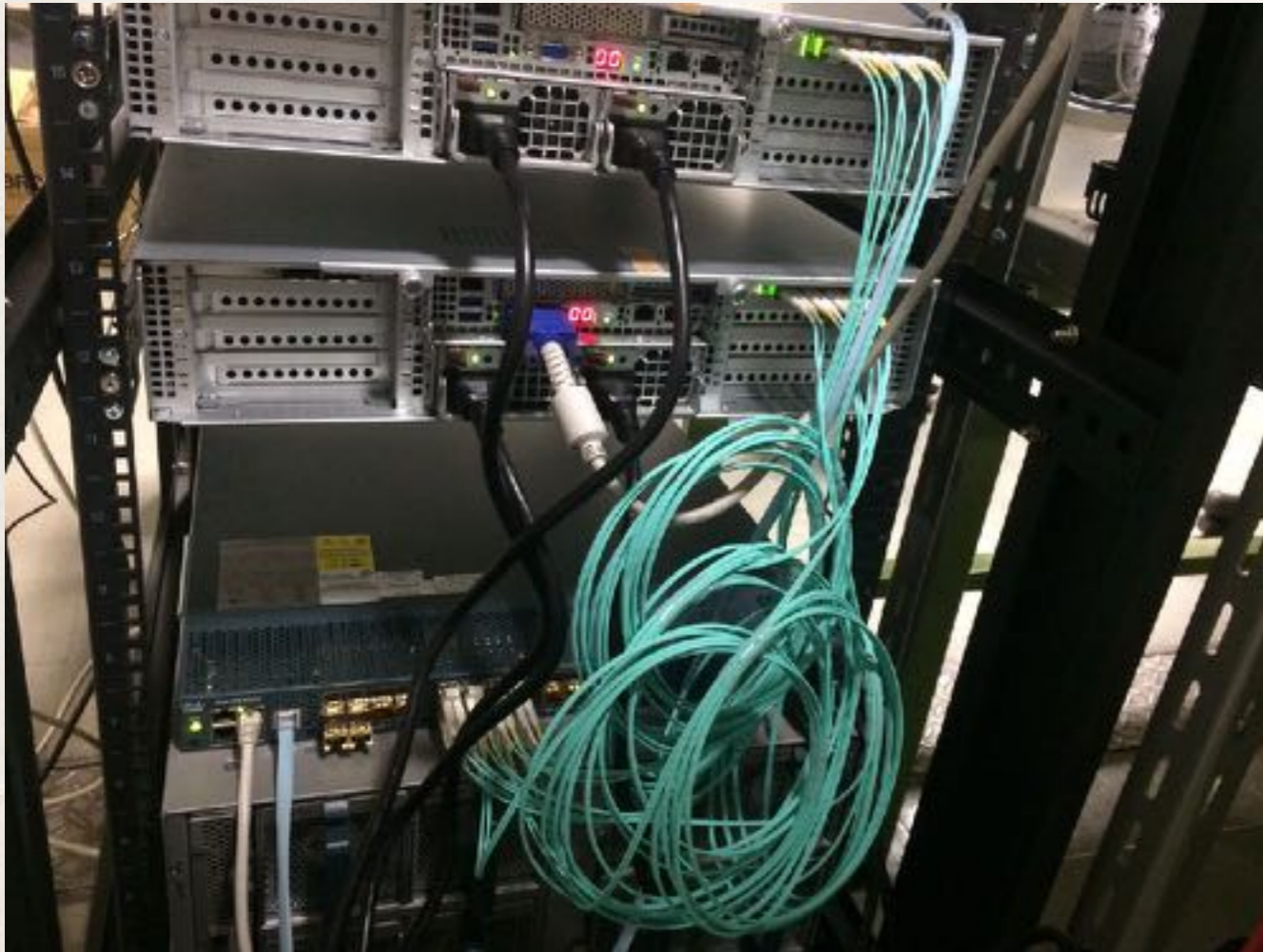


❖ `iperf -s -B 10.0.0.5`

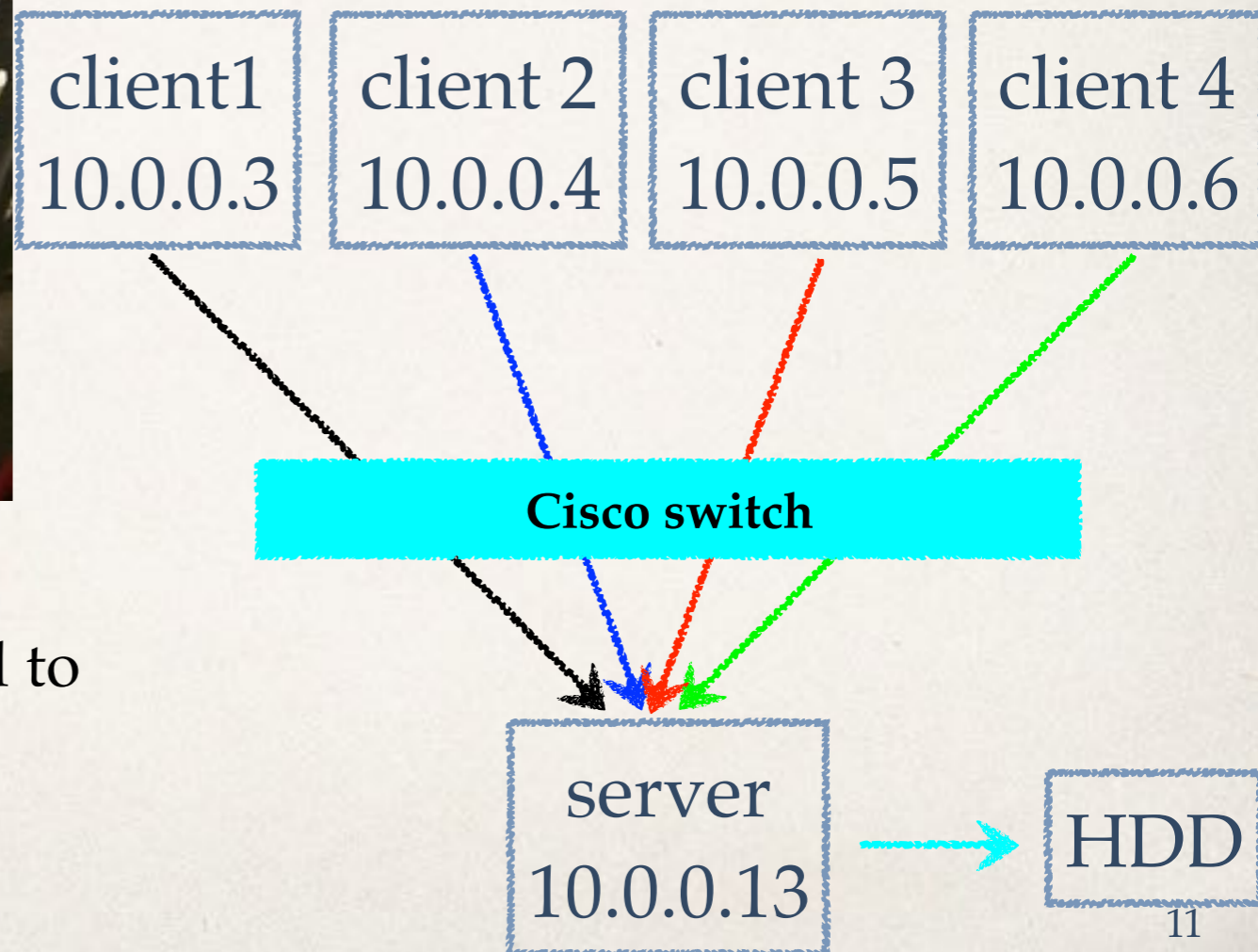
❖ `iperf -c 10.0.0.5 -B 10.0.0.15`

❖ 10Gbps performance as expected

Test configuration 1

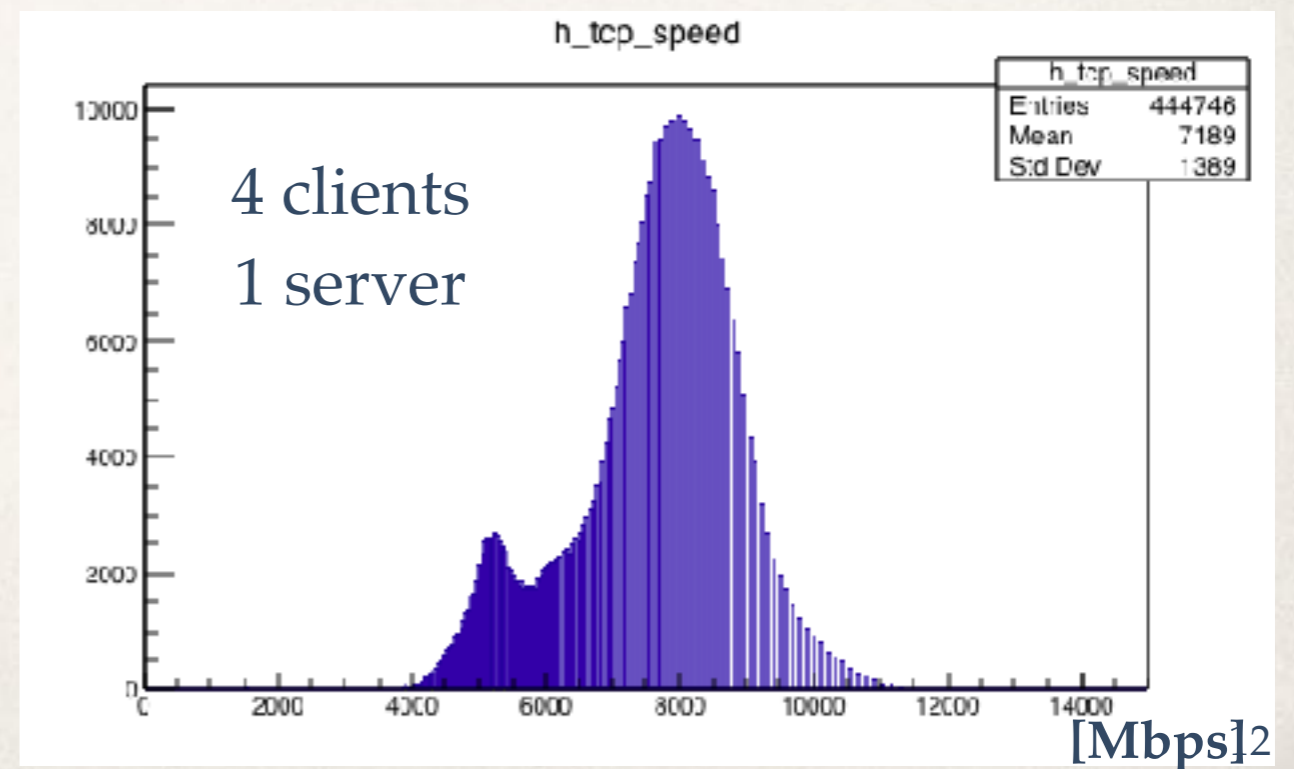
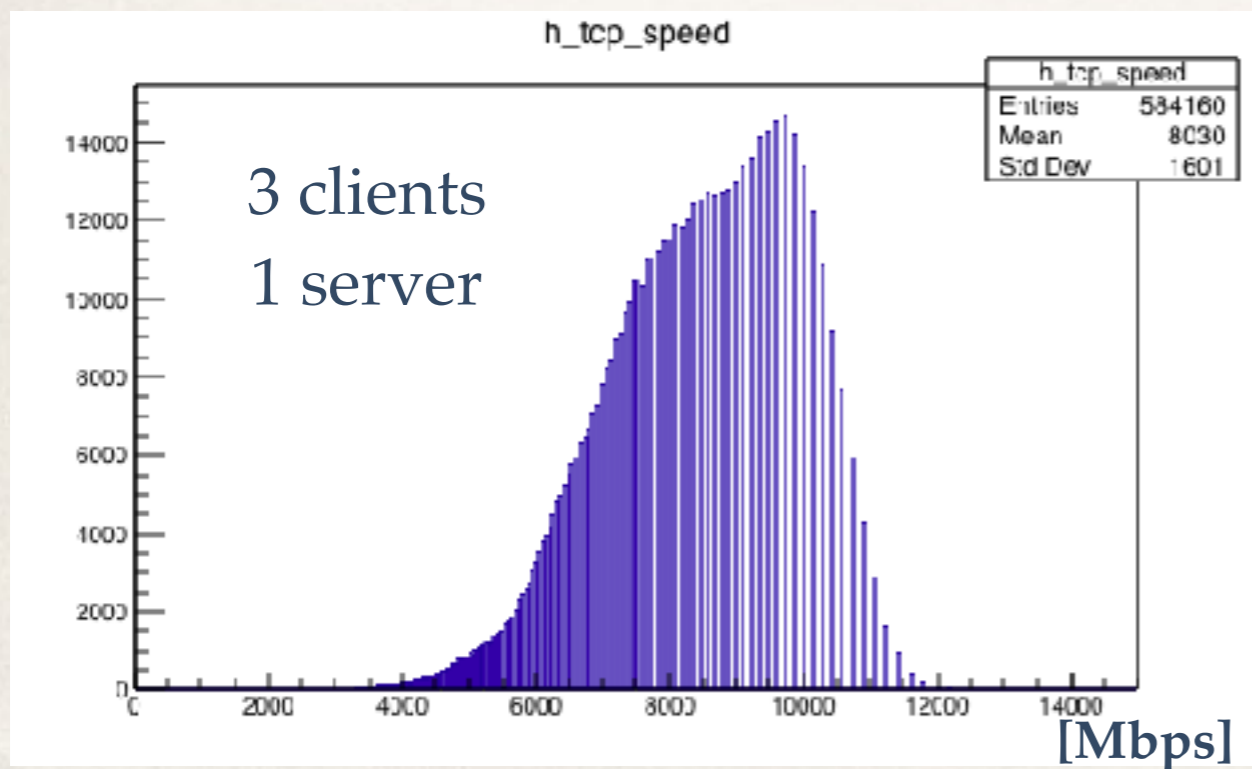
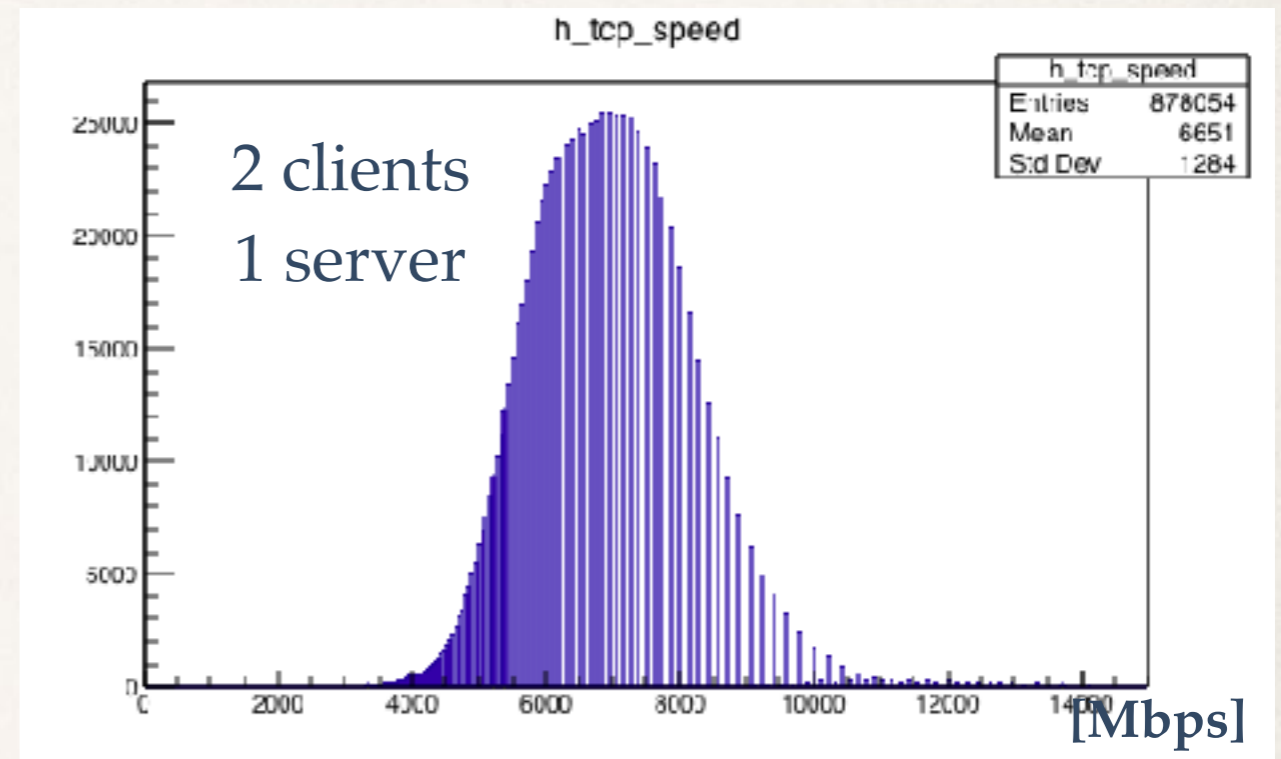
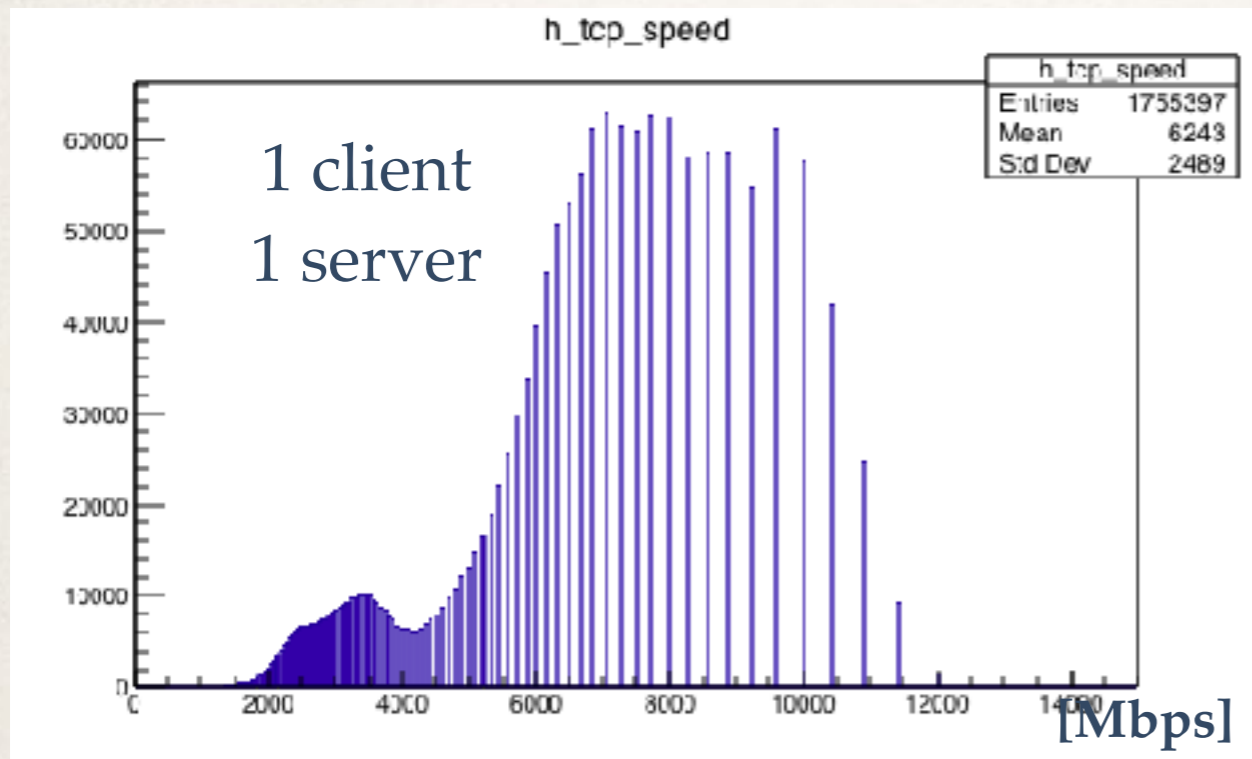


- ❖ data saved into hard drive
- ❖ server packs all clients data and create a time stamp to calculate the speed



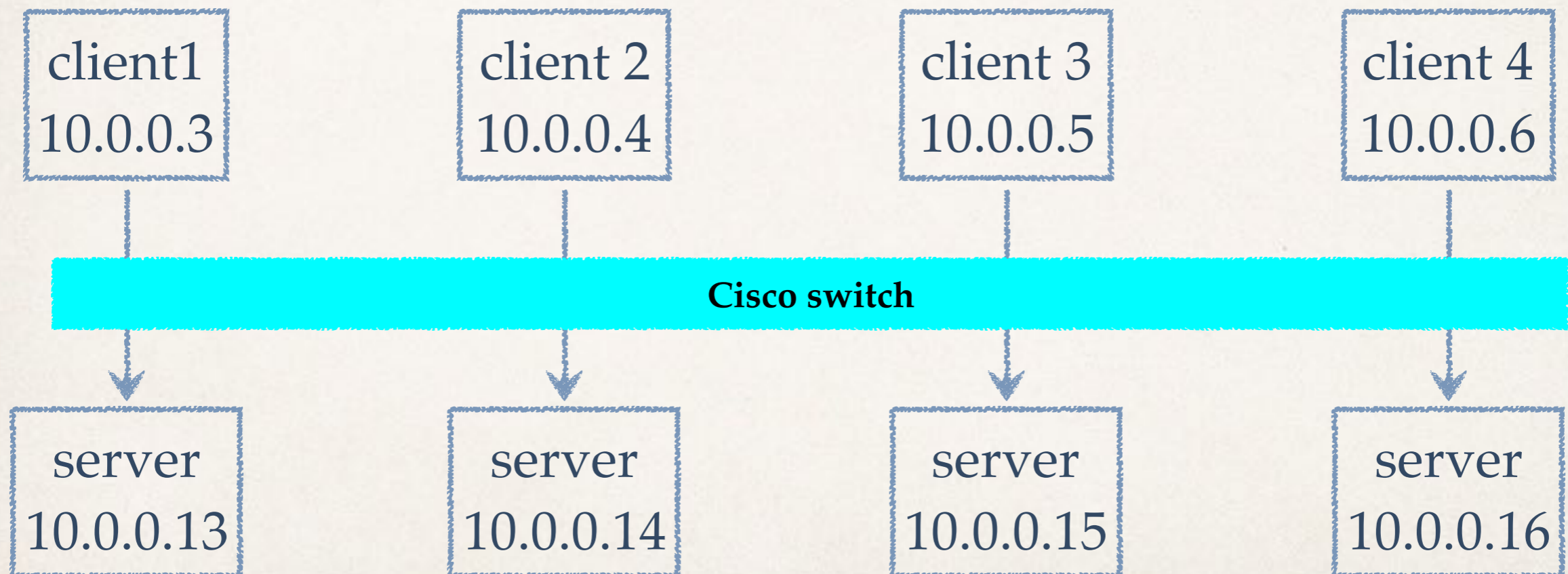
- ❖ All 8 ports from two servers are connected to Cisco 6120XP switch with SFP+ and fibers

Preliminary results (30kB event size/client)

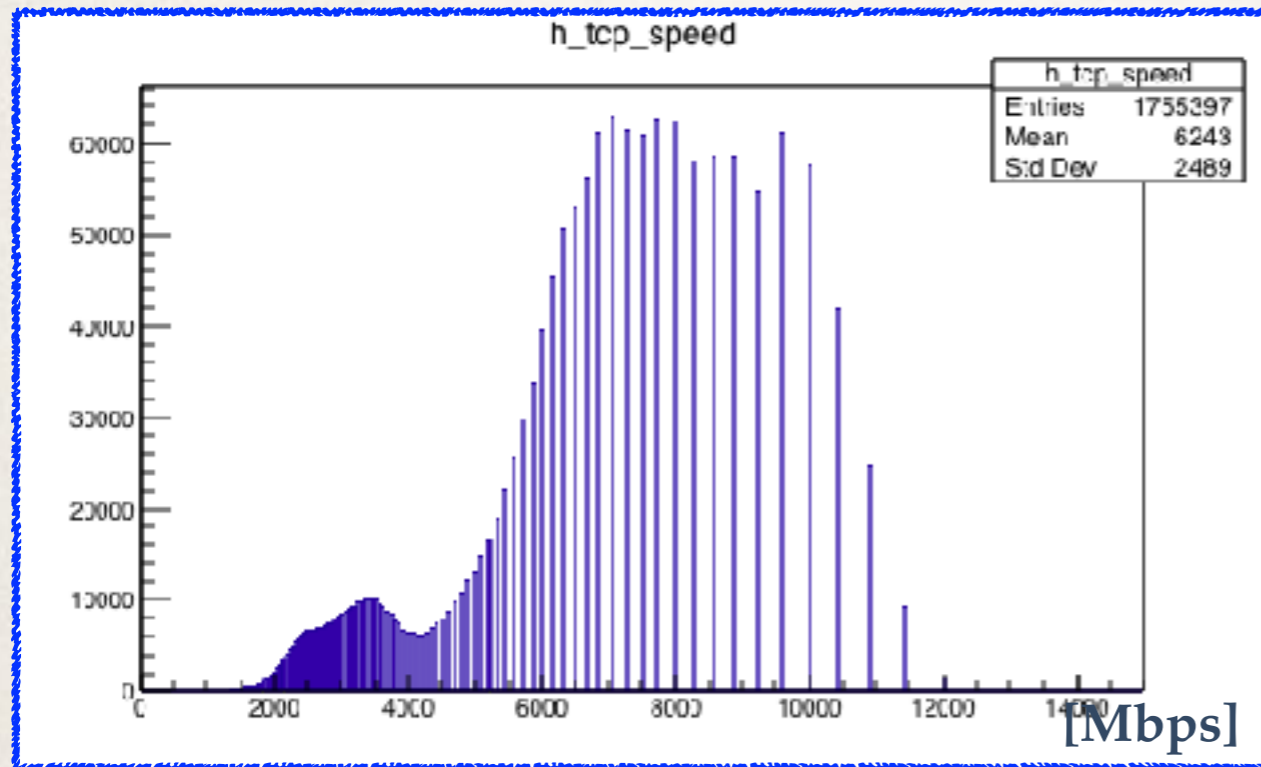


Test configuration 2

- ❖ All ports are connected to Cisco switch
- ❖ Run in parallel between two servers
- ❖ HDD becomes bottle neck —> **run online analyzer** —> **an offset due to total throughput**



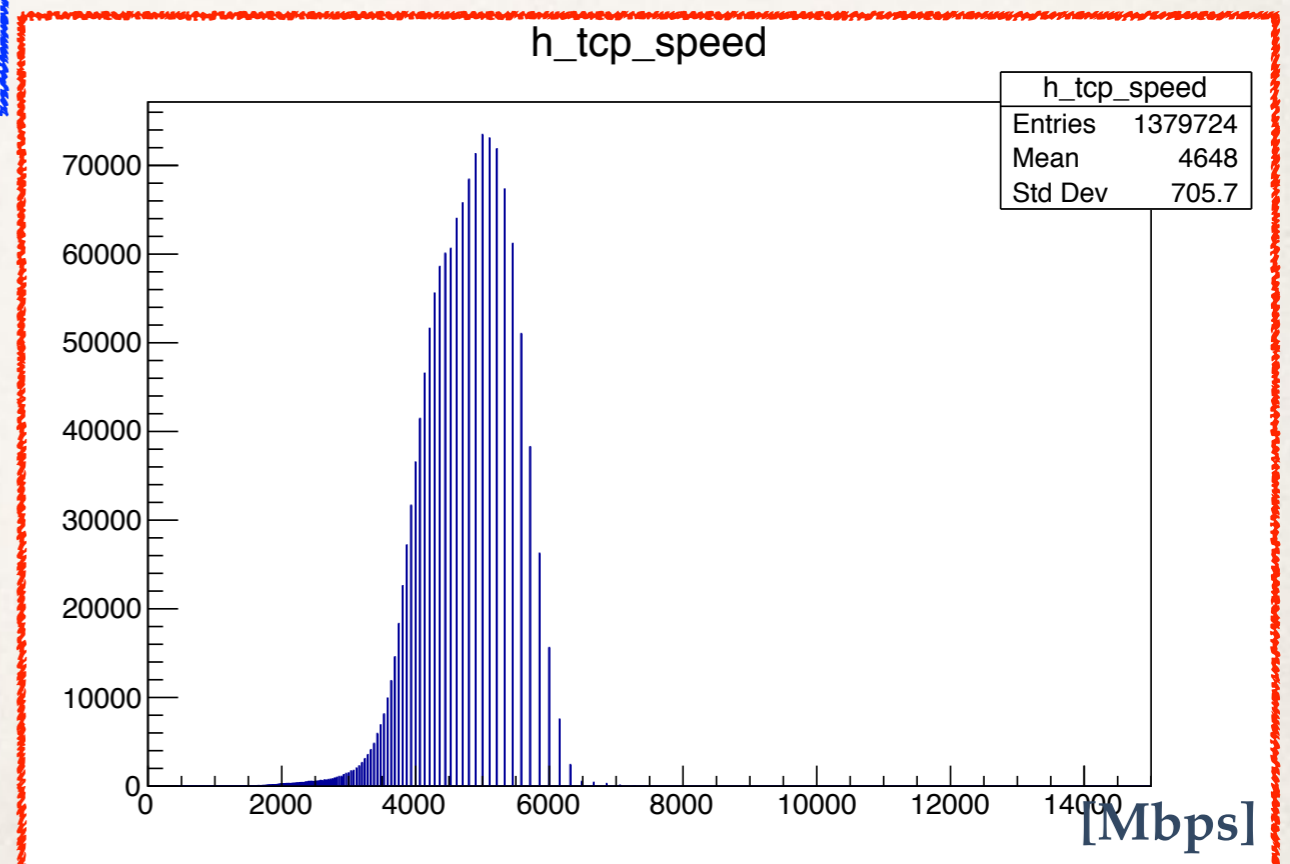
Test configuration 2



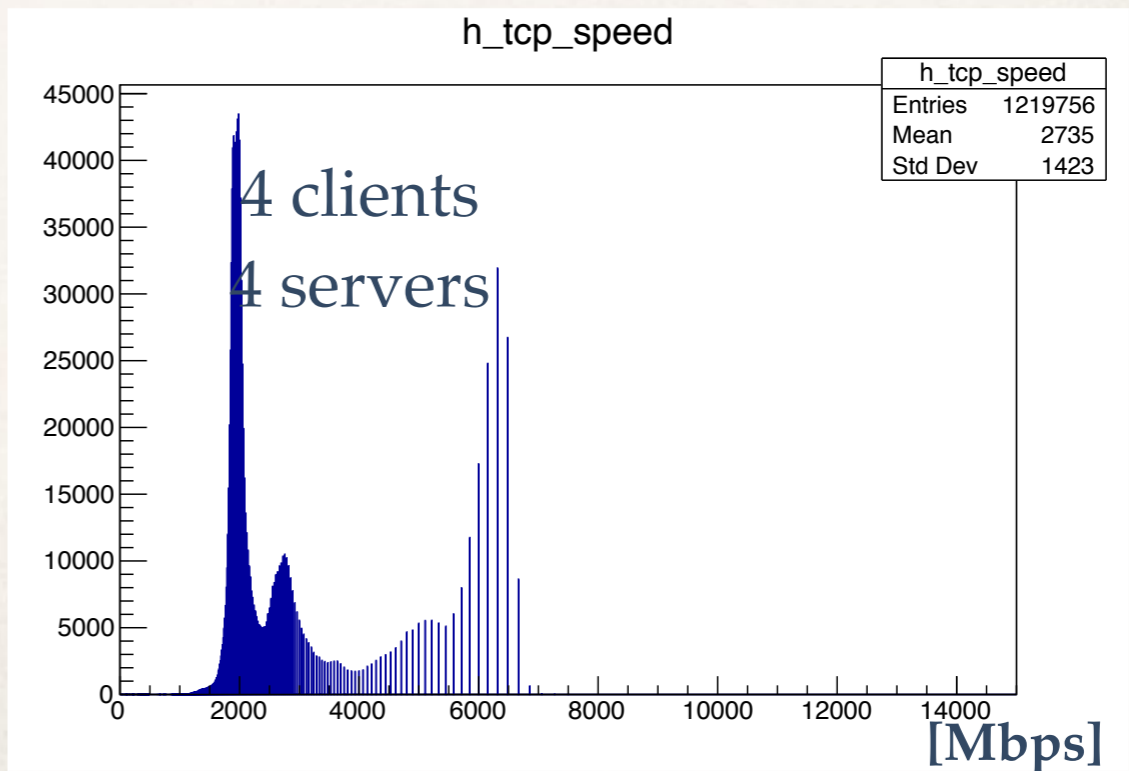
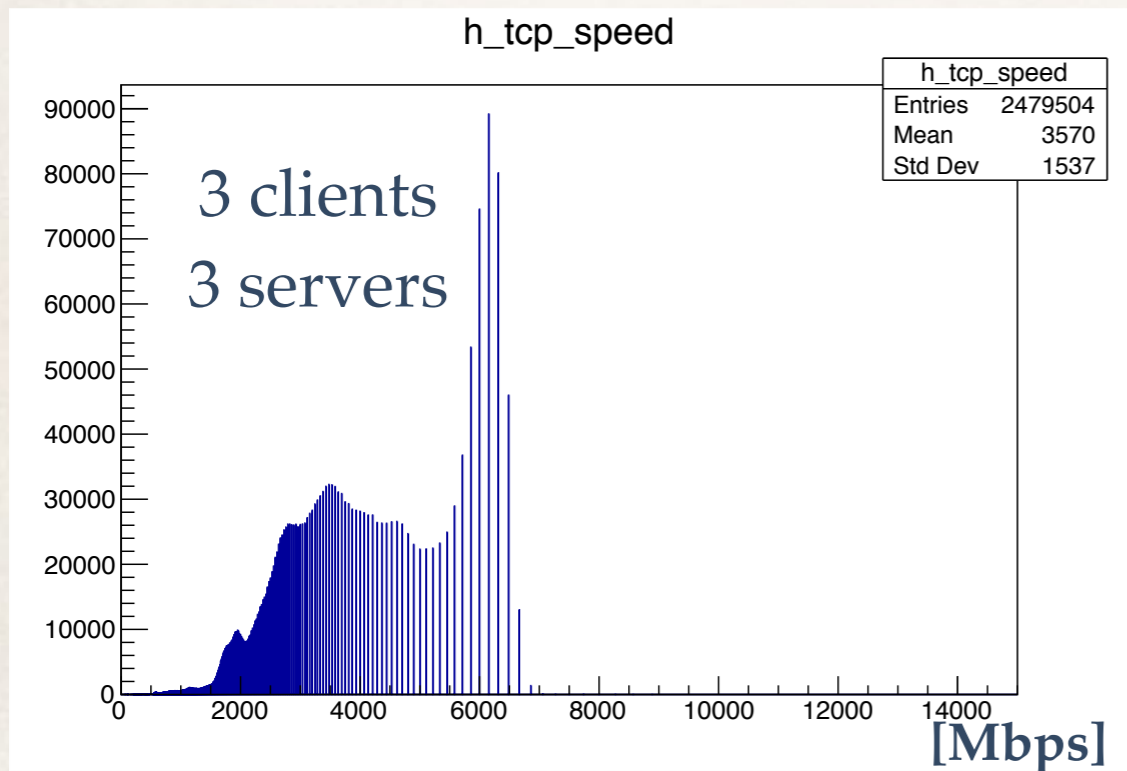
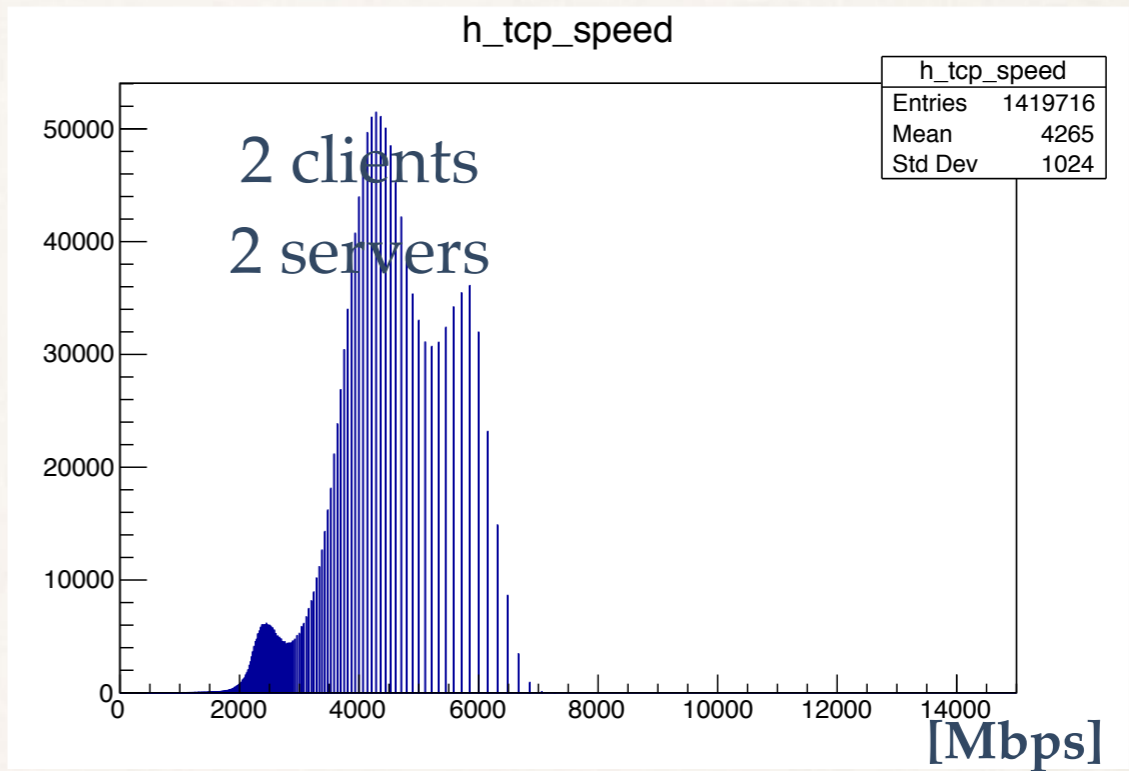
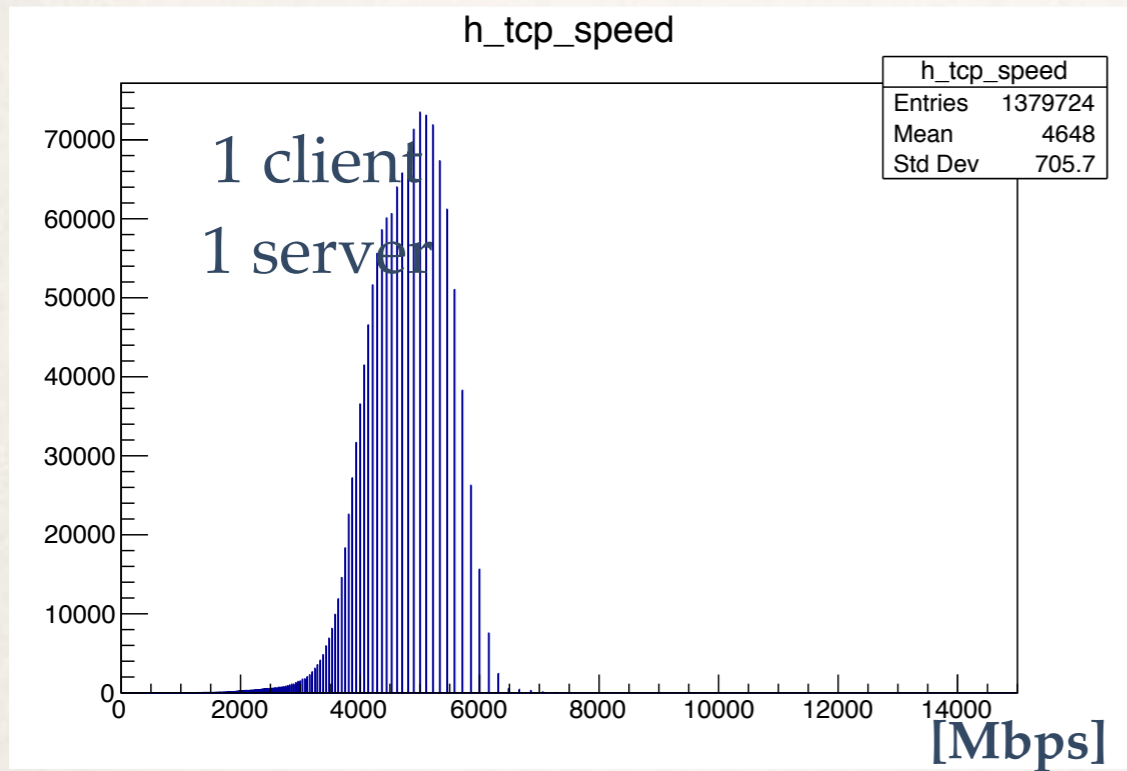
W/ HDD + offline
analysis

W/O HDD + online
analysis

The same one server + one
client setup;
keep this "offset" in mind.



Test configuration 2



iperf gives similar speed drops

The image shows three terminal windows from a Linux system, each displaying the output of an iperf test. The windows are titled 'oper@e50_server1:-', 'oper@e50_server1:- (on e50_server1)', and 'oper@e50_server1:- (on e50_server1)'. Each window shows a client connecting to a server on port 5001 and performing a throughput test. The results are summarized in a table with columns for Interval, Transfer, and Bandwidth. The first window shows a bandwidth of 3.54 Gbits/sec. The second window shows a bandwidth of 3.29 Gbits/sec. The third window shows a bandwidth of 4.92 Gbits/sec. Red dashed boxes highlight the bandwidth results in each window.

```
[ 3] local 10.0.0.13 port 54576 connected with 10.0.0.3 port 5001
[ 10] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  4.12 GBytes  3.54 Gbits/sec
[oper@e50_server1 mini_daq v2.0 10.0.0.13]$ iperf -c 10.0.0.3 -B 10.0.0.13^C
[oper@e50_server1 mini_daq v2.0 10.0.0.13]$ cd
[oper@e50_server1 ~]$ iperf -c 10.0.0.3 -B 10.0.0.13
-----
Client connecting to 10.0.0.3, TCP port 5001
Binding to local address 10.0.0.13
TCP window size: 1.06 MByte (default)
-----
[ 3] local 10.0.0.13 port 5001 connected with 10.0.0.3 port 5001
[ 10] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  3.22 GBytes  2.77 Gbits/sec
[oper@e50_server1 ~]$

oper@e50_server1:- (on e50_server1)
File Edit View Search Terminal Help
TCP window size: 85.0 KByte (default)
-----
[ 3] local 10.0.0.14 port 5001 connected with 10.0.0.4 port 5001
[ 10] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  3.83 GBytes  3.29 Gbits/sec
[oper@e50_server1 mini_daq v2.0 10.0.0.14]$ iperf -c 10.0.0.4 -B 10.0.0.14^C
[oper@e50_server1 mini_daq v2.0 10.0.0.14]$ cd
[oper@e50_server1 ~]$ iperf -c 10.0.0.4 -B 10.0.0.14
bind failed: Address already in use
-----
Client connecting to 10.0.0.4, TCP port 5001
Binding to local address 10.0.0.14
TCP window size: 85.0 KByte (default)
-----
[ 3] local 10.0.0.13 port 57694 connected with 10.0.0.4 port 5001
[ 10] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  2.97 GBytes  2.55 Gbits/sec
[oper@e50_server1 ~]$

oper@e50_server1:- (on e50_server1)
File Edit View Search Terminal Help
TCP window size: 85.0 KByte (default)
-----
[ 3] local 10.0.0.15 port 5001 connected with 10.0.0.5 port 5001
[ 10] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  4.10 GBytes  3.52 Gbits/sec
[oper@e50_server1 mini_daq v2.0 10.0.0.15]$ iperf -c 10.0.0.5 -B 10.0.0.15^C
[oper@e50_server1 mini_daq v2.0 10.0.0.15]$ cd
[oper@e50_server1 ~]$ iperf -c 10.0.0.5 -B 10.0.0.15
bind failed: Address already in use
-----
Client connecting to 10.0.0.5, TCP port 5001
Binding to local address 10.0.0.15
TCP window size: 85.0 KByte (default)
-----
[ 3] local 10.0.0.13 port 56276 connected with 10.0.0.5 port 5001
[ 10] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  5.73 GBytes  4.92 Gbits/sec
[oper@e50_server1 ~]$
```

- ❖ 3 servers: iperf -s -B 10.0.0.3; ...
- ❖ 3 clients: iperf -c 10.0.0.3 -B 10.0.0.13; ...
- ❖ total throughput of 10Gbps

Update: solution found

- ❖ Default route policy isn't smart enough for multiple ports on the same machine;
- ❖ Problem solved by assigning a static route to each pair of IP address

```
#!/bin/bash
# run as SU

#ip address for server0, 50Gbps, port 1
#ip addr add 10.0.0.1/24 dev ens2f0
#ip r add 10.0.0.0/8 dev ens2f0
#ip address for server0, 50Gbps, port 2
#ip addr add 10.0.0.2/24 dev ens2f1
#ip r add 10.0.0.0/8 dev ens2f1
#ip address for server0, 10Gbps, port 1
ip addr add 10.0.0.3/24 dev ens6f0
ip route add 10.0.0.13 via 10.0.0.3 dev ens6f0
#ip r add 10.0.0.3/32 dev ens6f0
#ip address for server0, 10Gbps, port 2
ip addr add 10.0.0.4/24 dev ens6f1
ip route add 10.0.0.14 via 10.0.0.4 dev ens6f1
#ip r add 10.0.0.4/32 dev ens6f1
#ip address for server0, 10Gbps, port 3
ip addr add 10.0.0.5/24 dev ens6f2
ip route add 10.0.0.15 via 10.0.0.5 dev ens6f2
#ip r add 10.0.0.5/32 dev ens6f2
#ip address for server0, 10Gbps, port 4
ip addr add 10.0.0.6/24 dev ens6f3
ip route add 10.0.0.16 via 10.0.0.6 dev ens6f3
#ip r add 10.0.0.6/32 dev ens6f3

#ip r add 10.0.0.2/32 dev eth0 # only 10.0.0.2
#ip r add 10.0.0.0/8 dev eth0 # 10.0.0.0 - 10
```

```
#!/bin/bash
# run as SU

#ip address for server0, 50Gbps, port 1
#ip addr add 10.0.0.11/24 dev ens2f0
#ip r add 10.0.0.0/8 dev ens2f0
#ip address for server0, 50Gbps, port 2
#ip addr add 10.0.0.12/24 dev ens2f1
#ip r add 10.0.0.0/8 dev ens2f1
#ip address for server0, 10Gbps, port 1
ip addr add 10.0.0.13/24 dev ens6f0
ip route add 10.0.0.3 via 10.0.0.13 dev ens6f0
#ip route add 10.0.0.3 via 10.0.0.13 dev ens6f0
#ip address for server0, 10Gbps, port 2
ip addr add 10.0.0.14/24 dev ens6f1
ip route add 10.0.0.4 via 10.0.0.14 dev ens6f1
#ip route add 10.0.0.4 via 10.0.0.14 dev ens6f1
#ip address for server0, 10Gbps, port 3
ip addr add 10.0.0.15/24 dev ens6f2
ip route add 10.0.0.5 via 10.0.0.15 dev ens6f2
#ip route add 10.0.0.5 via 10.0.0.15 dev ens6f2
#ip address for server0, 10Gbps, port 4
ip addr add 10.0.0.16/24 dev ens6f3
ip route add 10.0.0.6 via 10.0.0.16 dev ens6f3
#ip route add 10.0.0.6 via 10.0.0.16 dev ens6f3

#ip r add 10.0.0.2/32 dev eth0 # only 10.0.0.2
#ip r add 10.0.0.0/8 dev eth0 # 10.0.0.0 - 10
```

Update: solution found

iperf results from
direct cable connection

```
oper@e50_server1:~$ iperf -c 10.0.0.3 -B 10.0.0.13
bind failed: Address already in use
-----
Client connecting to 10.0.0.3, TCP port 5001
Binding to local address 10.0.0.13
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.13 port 54638 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  18.4 GBytes  8.96 Gbits/sec

oper@e50_server1:~$ iperf -c 10.0.0.4 -B 10.0.0.14
bind failed: Address already in use
-----
Client connecting to 10.0.0.4, TCP port 5001
Binding to local address 10.0.0.14
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.14 port 34072 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  18.7 GBytes  9.23 Gbits/sec

oper@e50_server1:~$ iperf -c 10.0.0.5 -B 10.0.0.15
bind failed: Address already in use
-----
Client connecting to 10.0.0.5, TCP port 5001
Binding to local address 10.0.0.15
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.15 port 45504 connected with 10.0.0.5 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  10.0 GBytes  8.62 Gbits/sec

oper@e50_server1:~$ iperf -c 10.0.0.6 -B 10.0.0.16
bind failed: Address already in use
-----
Client connecting to 10.0.0.6, TCP port 5001
Binding to local address 10.0.0.16
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.16 port 36832 connected with 10.0.0.6 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  10.4 GBytes  8.96 Gbits/sec
```

iperf results from
Cisco switch

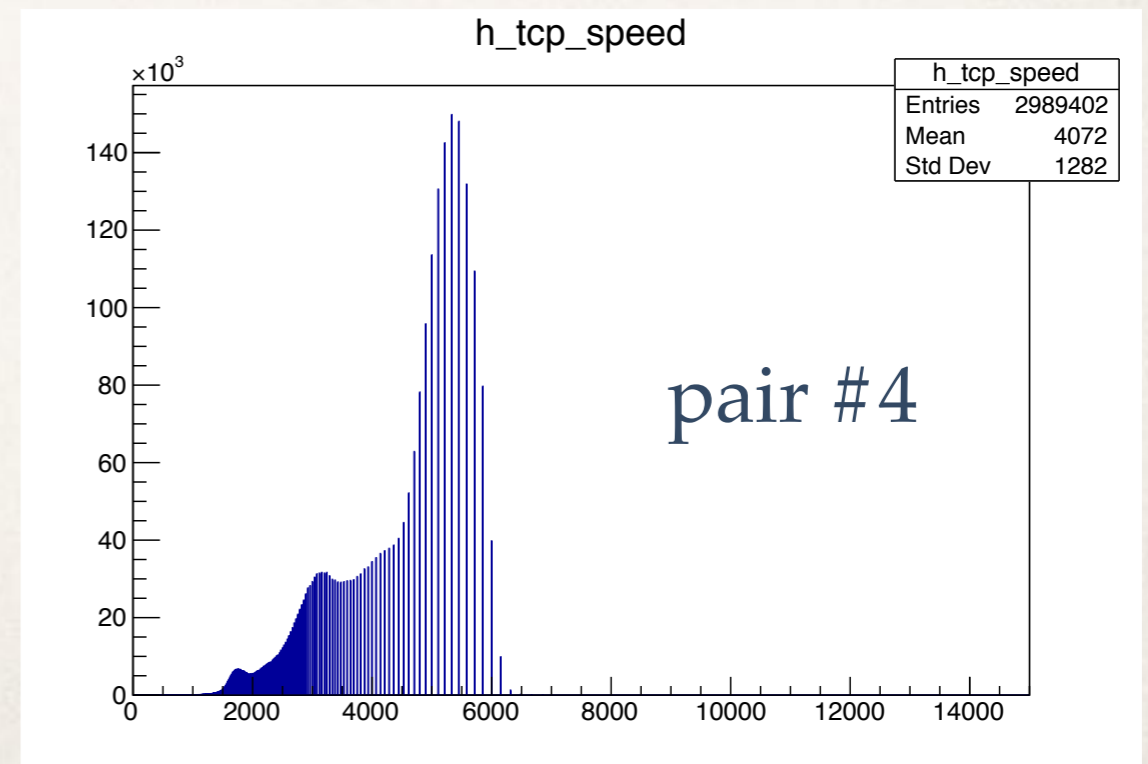
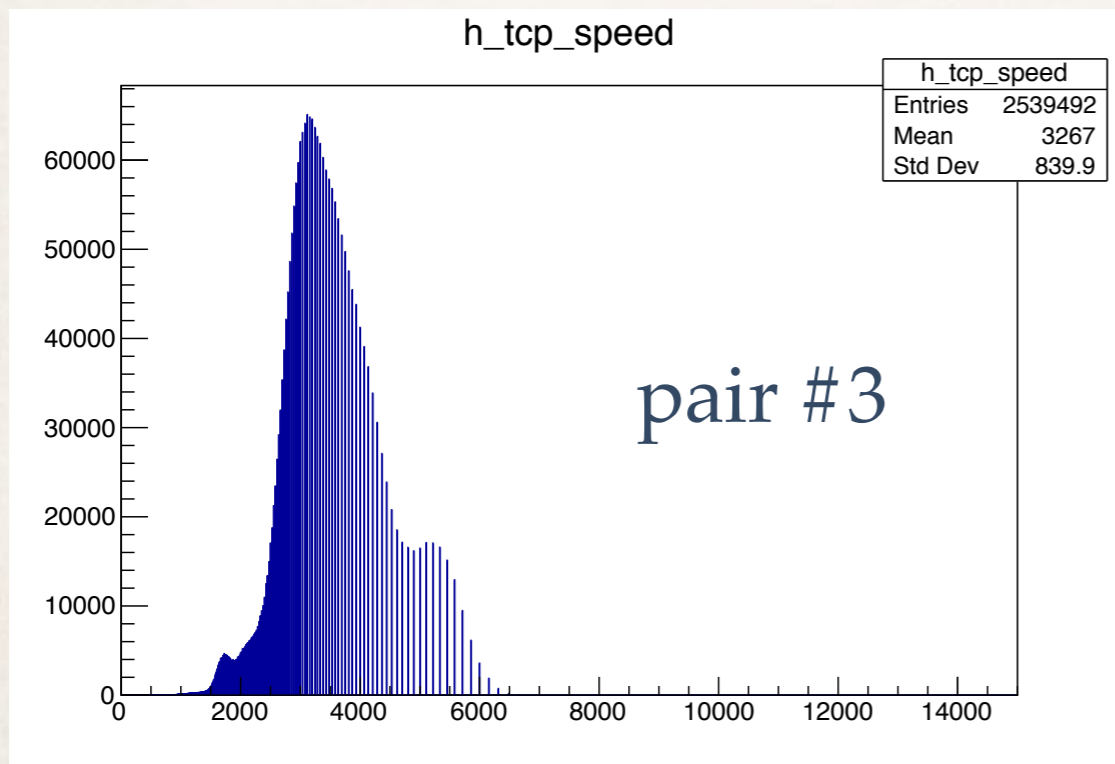
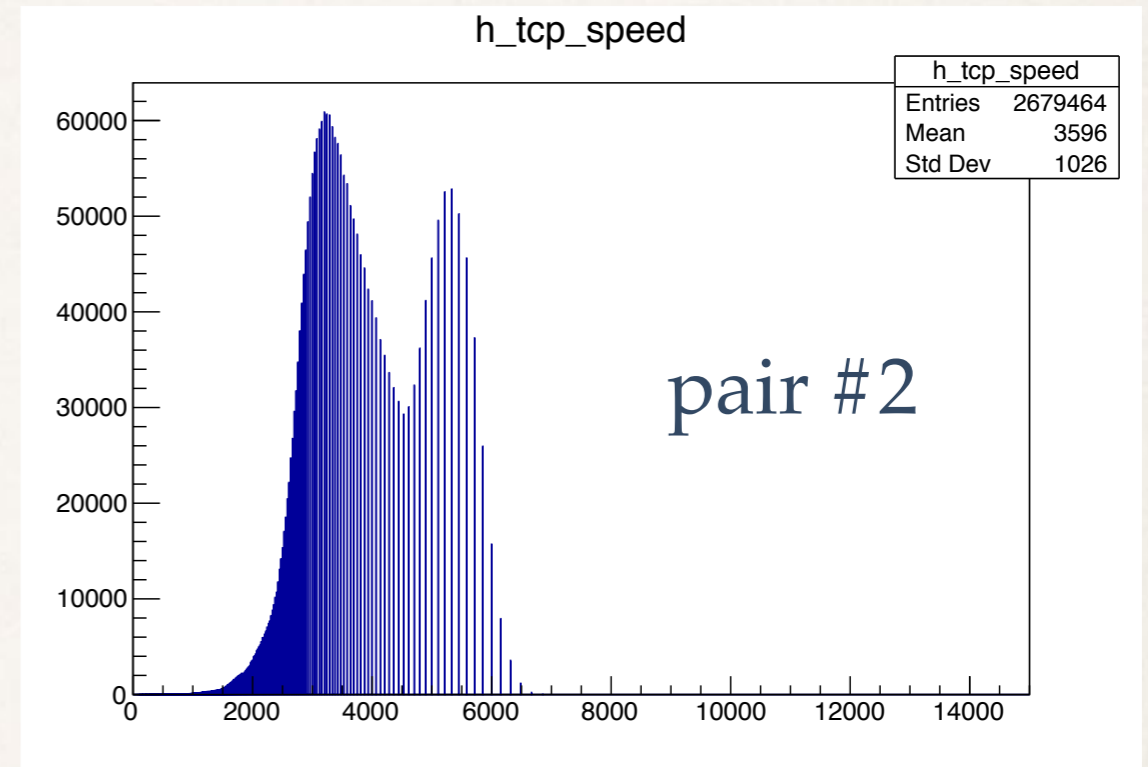
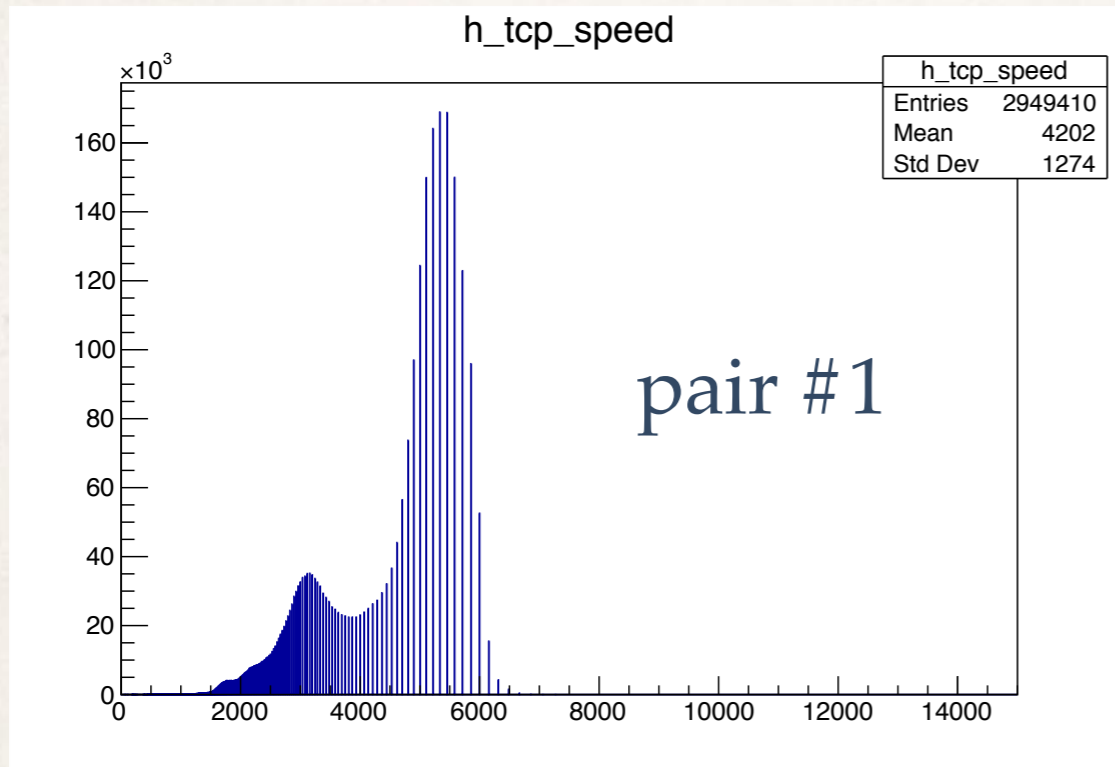
```
oper@e50_server1:~/mini_daq v2.0 10.0.0.13 (on e50_server1)
[ 3] 0.0-10.0 sec  18.9 GBytes  9.34 Gbits/sec
oper@e50_server1:~/mini_daq v2.0 10.0.0.13$ iperf -c 10.0.0.3 -B 10.0.0.13
-----
Client connecting to 10.0.0.3, TCP port 5001
Binding to local address 10.0.0.13
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.13 port 5001 connected with 10.0.0.3 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  18.8 GBytes  9.30 Gbits/sec

oper@e50_server1:~/mini_daq v2.0 10.0.0.14 (on e50_server1)
[ 3] 0.0-10.0 sec  18.8 GBytes  9.31 Gbits/sec
oper@e50_server1:~/mini_daq v2.0 10.0.0.14$ iperf -c 10.0.0.4 -B 10.0.0.14
-----
Client connecting to 10.0.0.4, TCP port 5001
Binding to local address 10.0.0.14
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.14 port 5001 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  18.3 GBytes  8.83 Gbits/sec

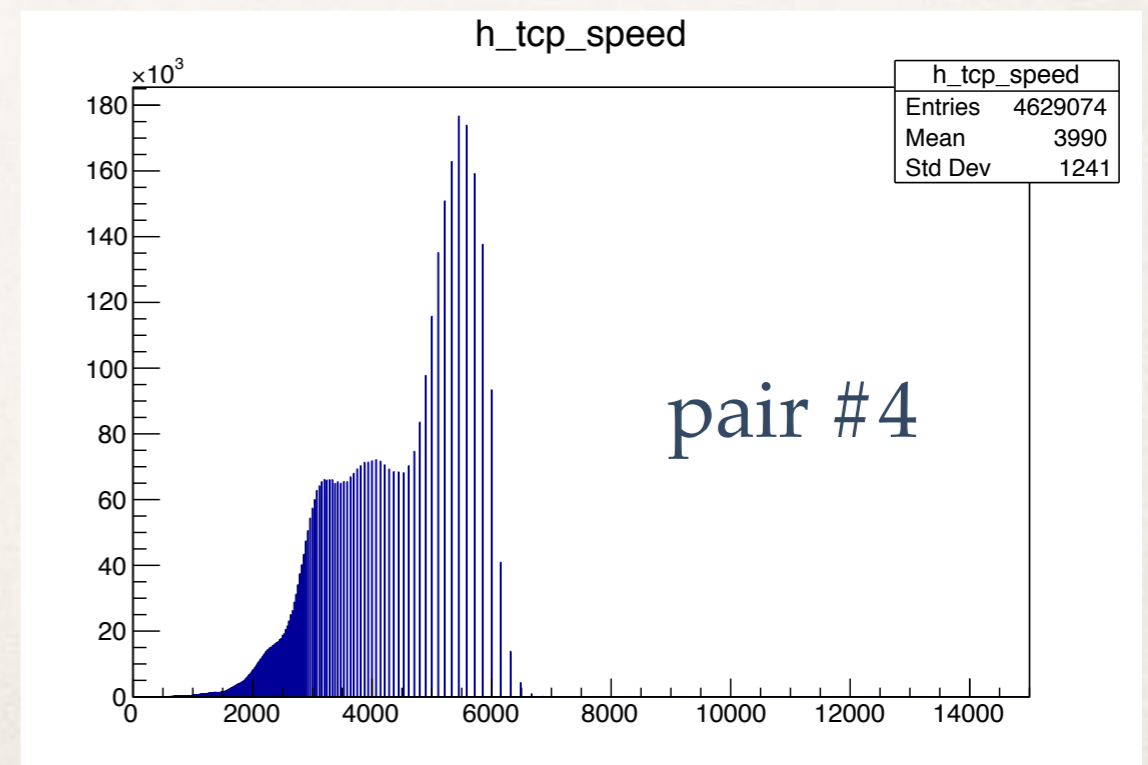
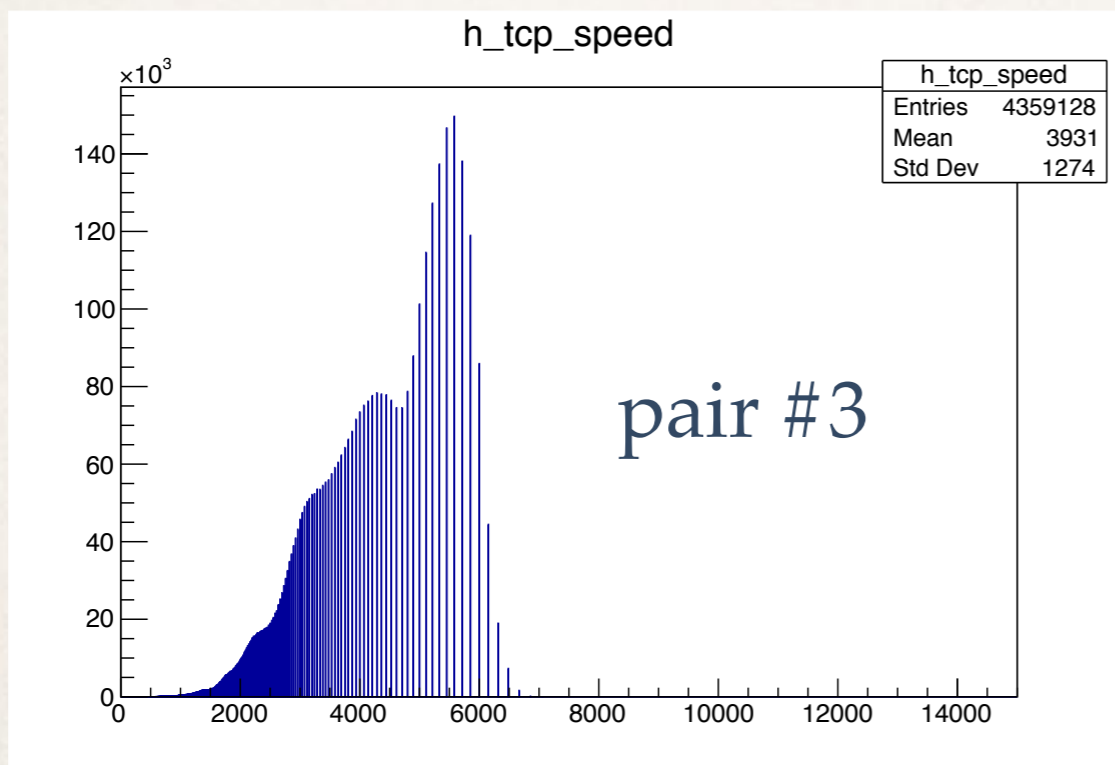
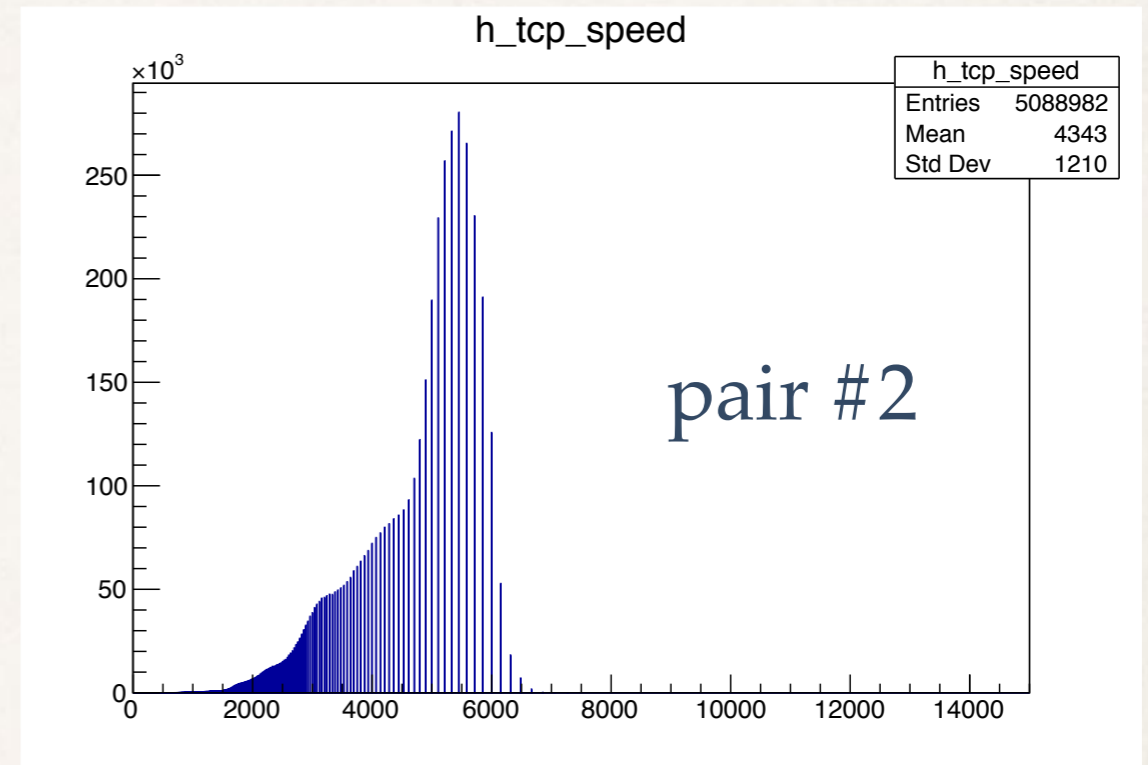
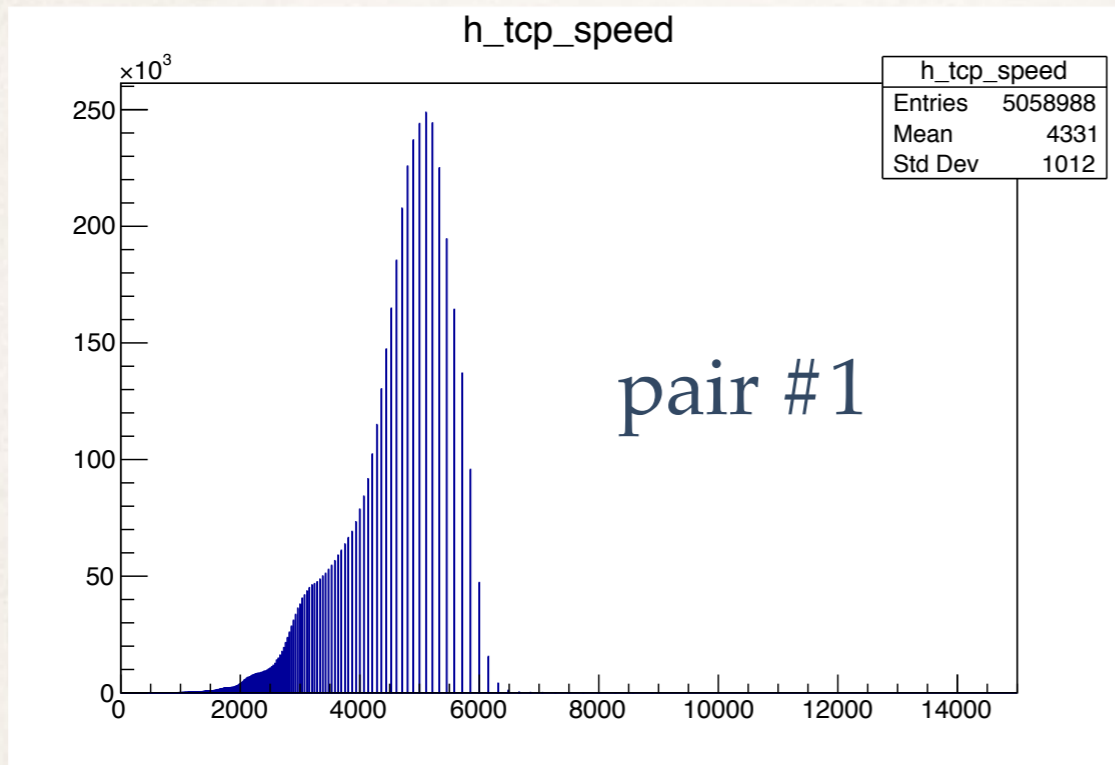
oper@e50_server1:~/mini_daq v2.0 10.0.0.15 (on e50_server1)
connect failed: Connection refused
oper@e50_server1:~/mini_daq v2.0 10.0.0.15$ iperf -c 10.0.0.5 -B 10.0.0.15
-----
Client connecting to 10.0.0.5, TCP port 5001
Binding to local address 10.0.0.15
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.15 port 5001 connected with 10.0.0.5 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  10.3 GBytes  8.81 Gbits/sec

oper@e50_server1:~/mini_daq v2.0 10.0.0.16 (on e50_server1)
connect failed: Connection refused
oper@e50_server1:~/mini_daq v2.0 10.0.0.16$ iperf -c 10.0.0.6 -B 10.0.0.16
-----
Client connecting to 10.0.0.6, TCP port 5001
Binding to local address 10.0.0.16
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.0.16 port 5001 connected with 10.0.0.6 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  9.74 GBytes  8.36 Gbits/sec
```

Update: solution found (direct connection)



Update: solution found (Cisco switch)



CPU usage

- ❖ a rough CPU usage obtained with linux “top” command
- ❖ each thread is consuming the full power of CPU -> quite heavy

```
oper@e50_server0:~/mini_daq_v2.0/data (on e50_server0)
top - 11:17:36 up 31 days, 34 min, 4 users, load average: 1.36, 0.59, 0.29
Tasks: 515 total, 1 running, 514 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.6 us, 2.1 sy, 0.0 ni, 93.3 id, 1.5 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 26387265+total, 22402718+free, 4086116 used, 35759364 buff/cache
KiB Swap: 4194300 total, 4194300 free, 0 used, 25804961+avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  %CPU  %MEM    TIME+  COMMAND
 19657 oper      20   0 615220 385440 9700  177.2  0.1   1:19.05 mini_daq s+
 38209 root       20   0     0     0     0    0.0  0.0   0:00.00 kworker/0:0
 40138 root       20   0     0     0     0    1.3  0.0   0:05.10 kworker/13+
 22526 root       20   0     0     0     0    0.7  0.0   0:02.44 kworker/10+
   75 root       20   0     0     0     0    0.3  0.0   0:01.04 ksoftirqd/+
  153 root       rt   0     0     0     0    0.3  0.0   0:06.25 watchdog/29
 2833 root       20   0     0     0     0    0.3  0.0   0:19.35 kworker/14+
 19572 root       20   0     0     0     0    0.3  0.0   0:00.06 kworker/u8+
 19675 oper      20   0 158076 2632 1560  0.3  0.0   0:00.13 top
 31675 oper      20   0 630120 20528 12244  0.3  0.0   0:06.59 gnome-tera+
   1 root       20   0 193624 6580 2396  0.0  0.0   1:06.02 systemd
   2 root       20   0     0     0     0    0.0  0.0   0:00.38 kthreadd
   3 root       20   0     0     0     0    0.0  0.0   0:00.13 ksoftirqd/0
   5 root       0 -20     0     0     0    0.0  0.0   0:00.00 kworker/0:0H
   8 root       rt   0     0     0     0    0.0  0.0   0:00.29 migration/0
   9 root       20   0     0     0     0    0.0  0.0   0:00.00 rcu_bh
  10 root       20   0     0     0     0    0.0  0.0   1:58.68 rcu sched
```

configuration 1;
at server side

```
oper@e50_server0:~
top - 15:21:02 up 33 days, 4:38, 12 users load average: 8.10, 3.84, 1.83
Tasks: 531 total, 1 running, 530 sleeping, 0 stopped, 0 zombie
%Cpu(s): 19.4 us, 3.8 sy, 0.0 ni, 75.3 id, 0.0 wa, 0.0 hi, 1.4 si, 0.0 st
KiB Mem : 26387265+total, 19646854+free, 5164148 used, 62239972 buff/cache
KiB Swap: 4194300 total, 4194300 free, 0 used, 25802596+avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  %CPU  %MEM    TIME+  COMMAND
 29678 oper      20   0 615216 387432 9696  222.3  0.1   5:22.73 mini_daq_server
 29679 oper      20   0 615216 387440 9696  221.6  0.1   5:16.50 mini_daq_server
 29686 oper      20   0 623412 385592 9712  221.3  0.1   9:09.09 mini_daq_server
 29682 oper      20   0 615216 387428 9696  221.3  0.1   5:05.01 mini_daq_server
 29648 oper      20   0 523292 88780 53730  25.6  0.0   0:34.18 mini_daq_analyz
 29625 oper      20   0 523292 88780 53730  25.6  0.0   0:34.18 mini_daq_analyz
 29698 oper      20   0 523312 88804 53730  24.9  0.0   0:23.24 mini_daq_analyz
 29663 oper      20   0 523308 88796 53730  24.6  0.0   0:31.93 mini_daq_analyz
  1316 root       20   0     4368    680    524  1.3  0.0   8:17.13 rngd
 29846 oper      20   0 158136 2700 1550  0.7  0.0   0:00.13 top
   94 root       rt   0     0     0     0    0.3  0.0   0:00.22 migration/17
 22111 root       20   0     0     0     0    0.3  0.0   0:09.16 kworker/0:2
 26185 root       20   0     0     0     0    0.3  0.0   0:01.66 kworker/u80:2
 27760 root       20   0     0     0     0    0.3  0.0   0:00.58 kworker/u80:4
   1 root       20   0 193624 6580 2396  0.0  0.0   1:09.68 systemd
   1 root       20   0     0     0     0    0.0  0.0   0:00.40 kthreadd
   2 root       20   0     0     0     0    0.0  0.0   0:00.14 ksoftirqd/0
   3 root       20   0     0     0     0    0.0  0.0   0:00.00 kworker/0:0H
   5 root       0 -20     0     0     0    0.0  0.0   0:00.29 migration/0
   8 root       rt   0     0     0     0    0.0  0.0   0:00.00 rcu_bh
   9 root       20   0     0     0     0    0.0  0.0   1:58.43 rcu sched
  10 root       20   0     0     0     0    0.0  0.0   0:07.58 watchdog/0
  11 root       rt   0     0     0     0    0.0  0.0   0:05.64 watchdog/1
  12 root       rt   0     0     0     0    0.0  0.0   0:00.21 migration/1
```

configuration 2;
at server side

Summary & todo

- ❖ Spikes are from digits handling of time stamp (64bit beast!)?
- ❖ Configuration #1:
 - ❖ More clients improve the performance (parallel TCP buffer?)
- ❖ Configuration #2:
 - ❖ Need to be careful for the route table configuration;
 - ❖ Degraded performance compared to iperf is due to the CPU consumption of mini_daq on a SMP system??
- ❖ Total throughput with Cisco switch seems fine! (no apparent “data collision effects” observed for 4 pairs of client-server connections)
- ❖ Results in Page 20 can be used as a “primitive” P.D.F. (probability distribution function) for a more integrated simulation study

Thank you for your attention!