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

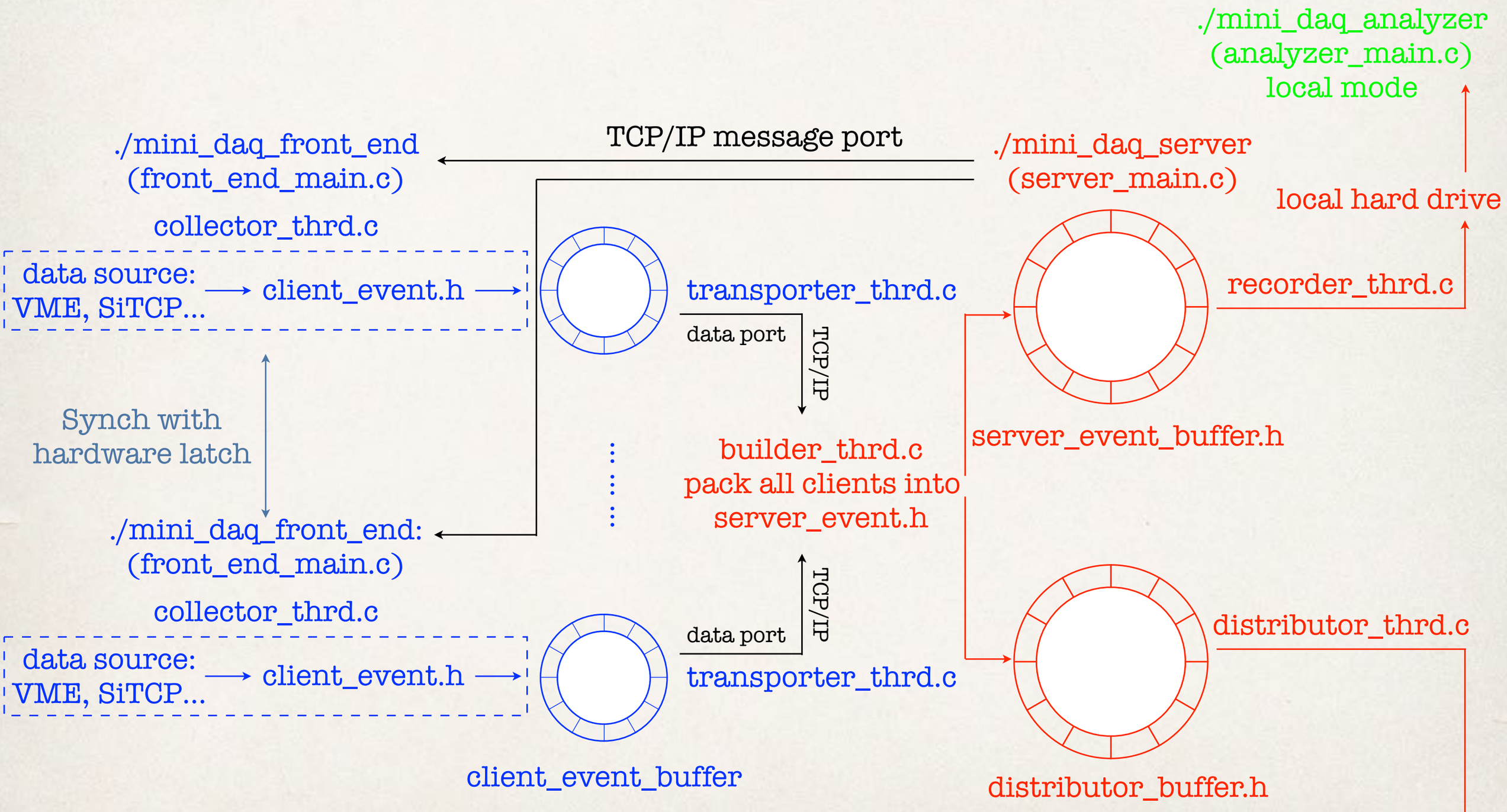
Software: a minimalism DAQ

- A tiny DAQ package with basic functionalities for data taking in physics experiment

- programed 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

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

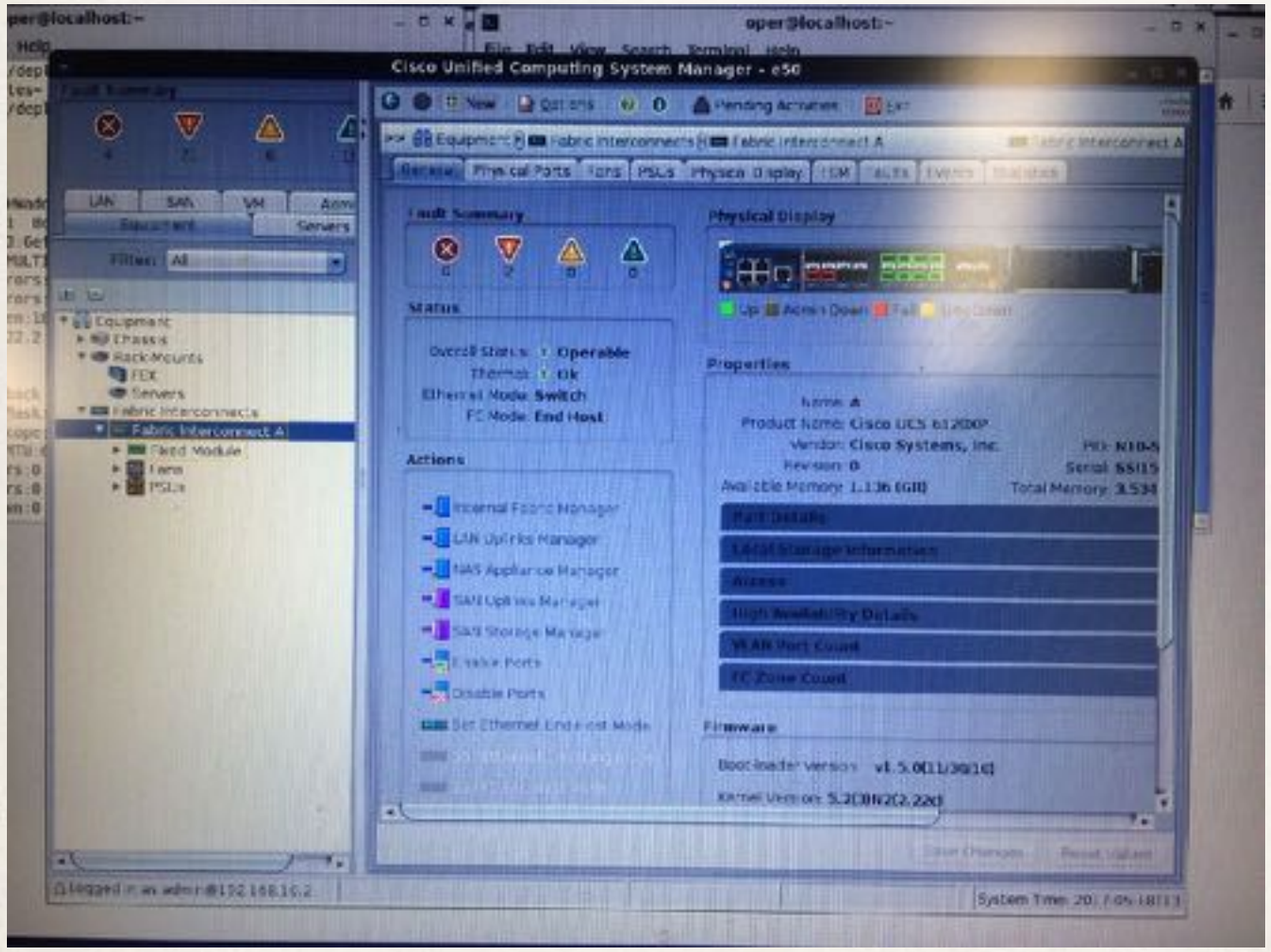
```
e50-A /system # scope fabric-interconnect a
e50-A /fabric-interconnect #
e50-A /fabric-interconnect #
e50-A /fabric-interconnect #
e50-A /fabric-interconnect # show detail

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
    OOB IP Addr: 172.27.42.155
    OOB Gateway: 172.27.42.254
    OOB 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
```

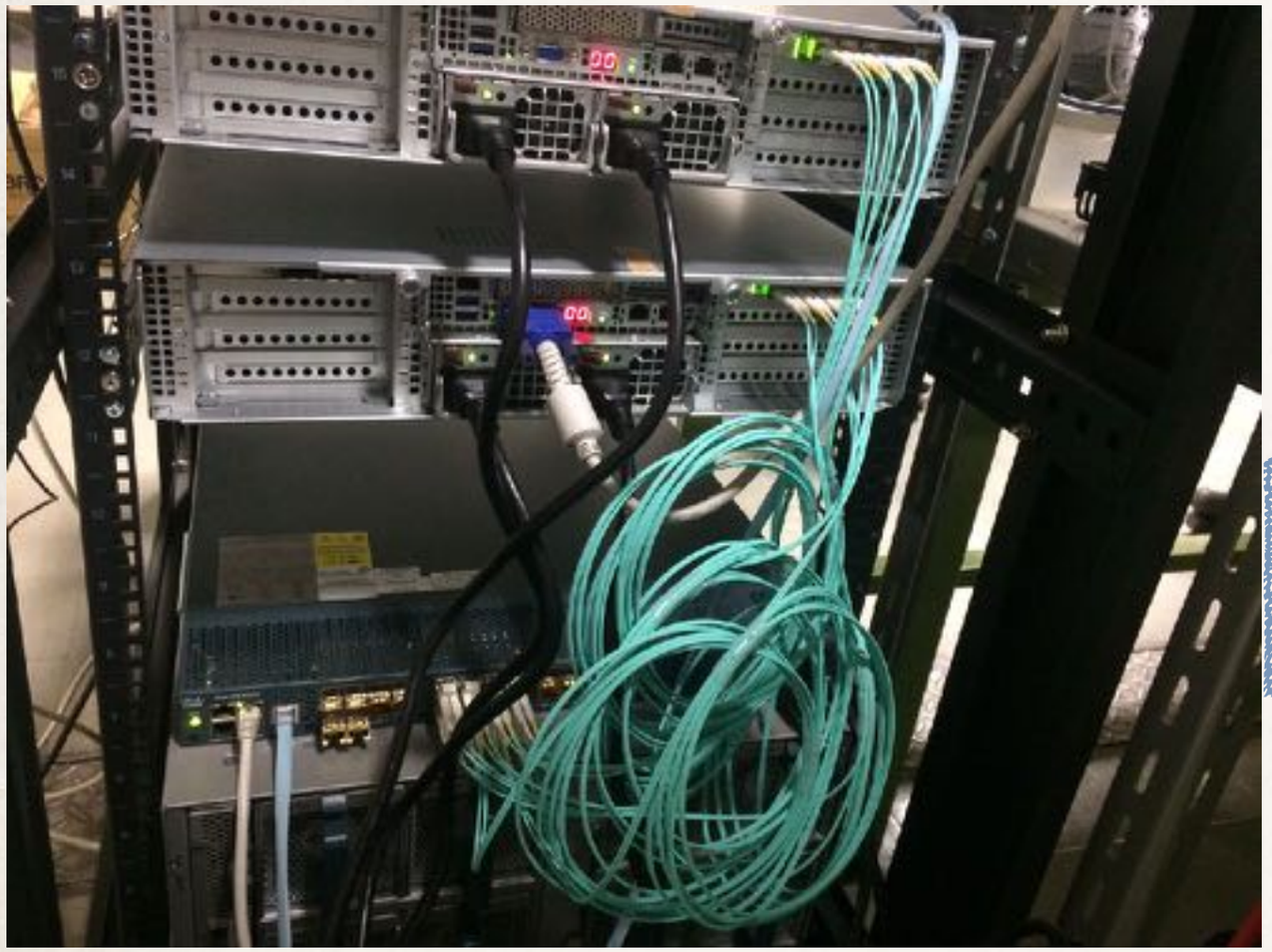
Hardware: Cisco 6120XP configuration

- Finally, it works!

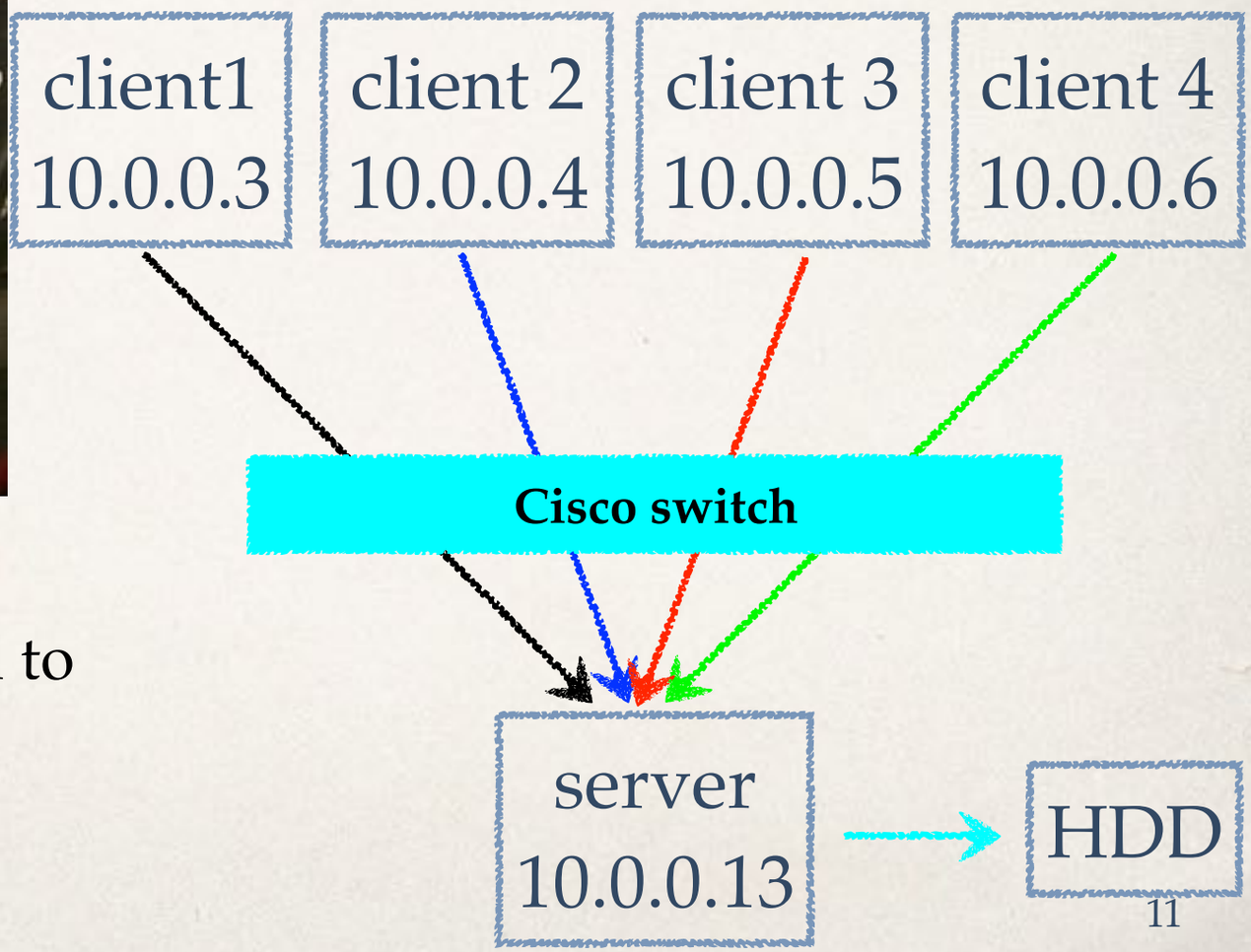


- iperf -s -B 10.0.0.5
- iperf -c 10.0.0.5 -B 10.0.0.15
- 10Gbps performance as expected

Test configuration 1

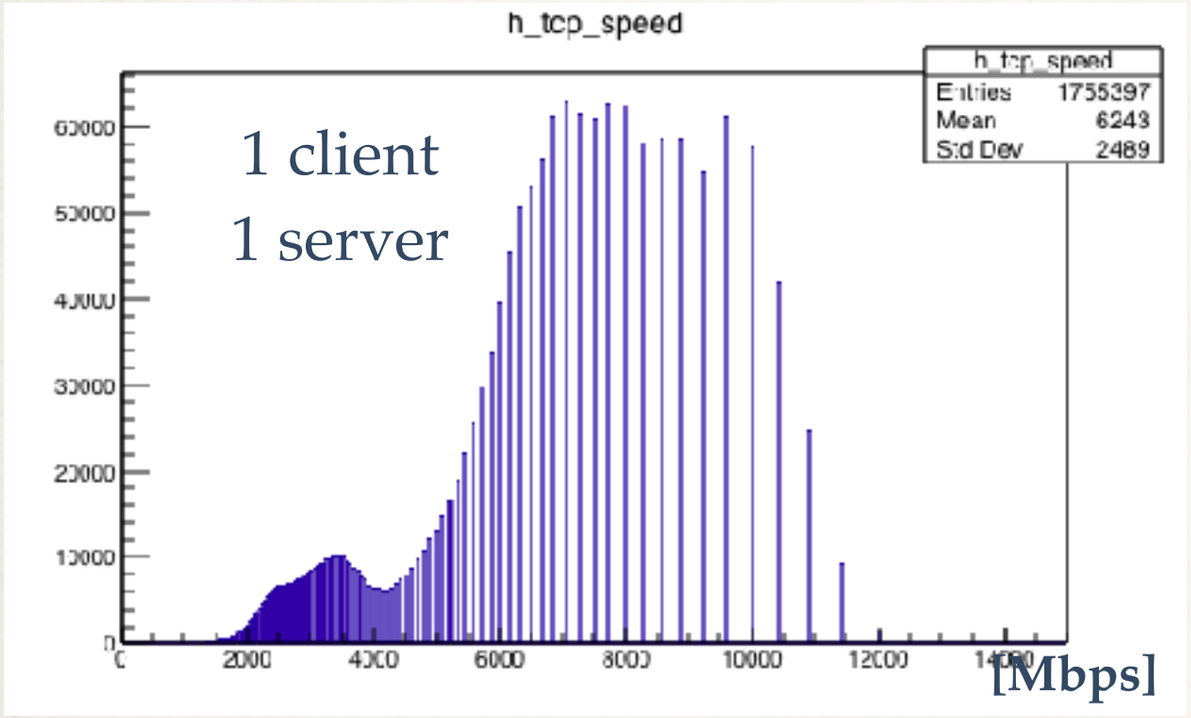


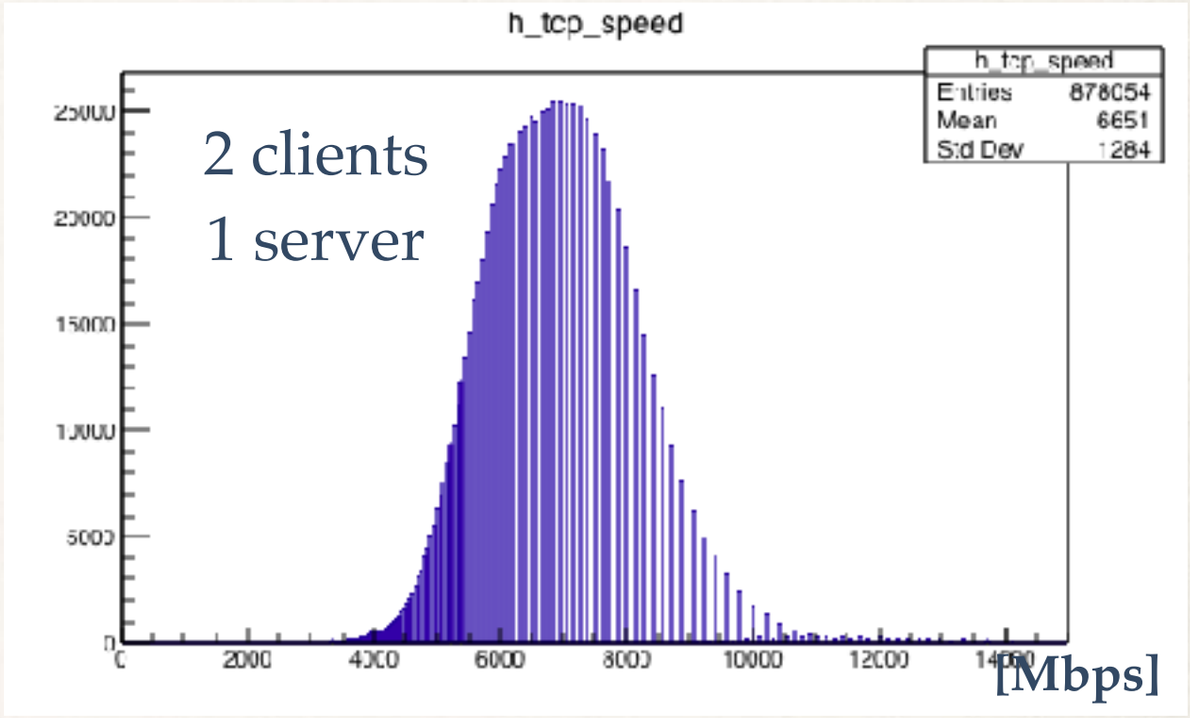
- data saved into hard drive
- server packs all clients data and create a time stamp to calculate the speed

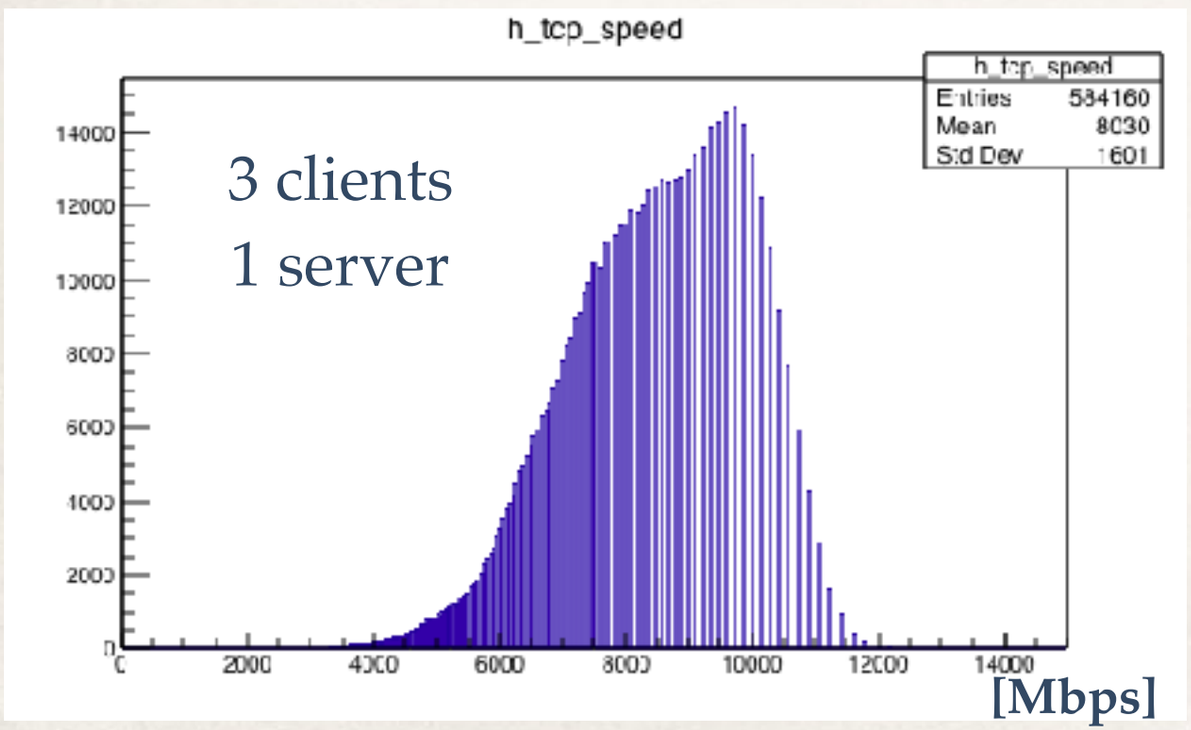


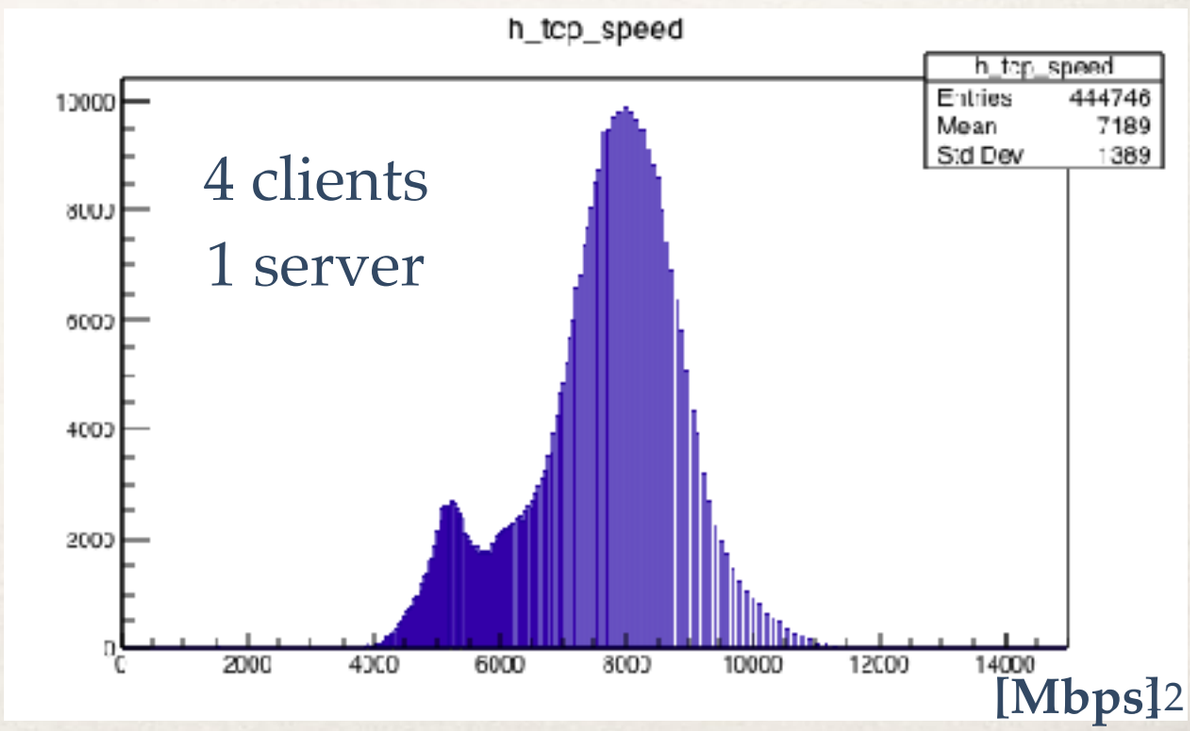
- All 8 ports from two servers are connected to Cisco 6120XP switch with SFP+ and fibers

Preliminary results (30kB event size/client)



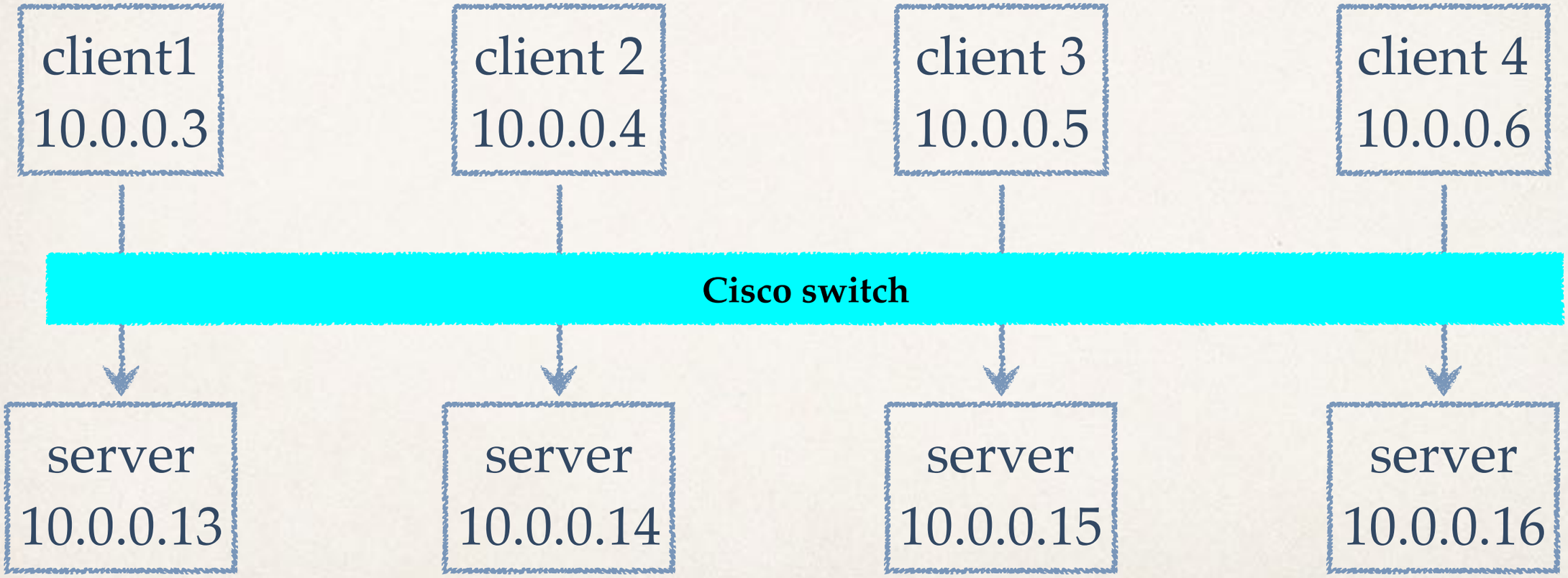




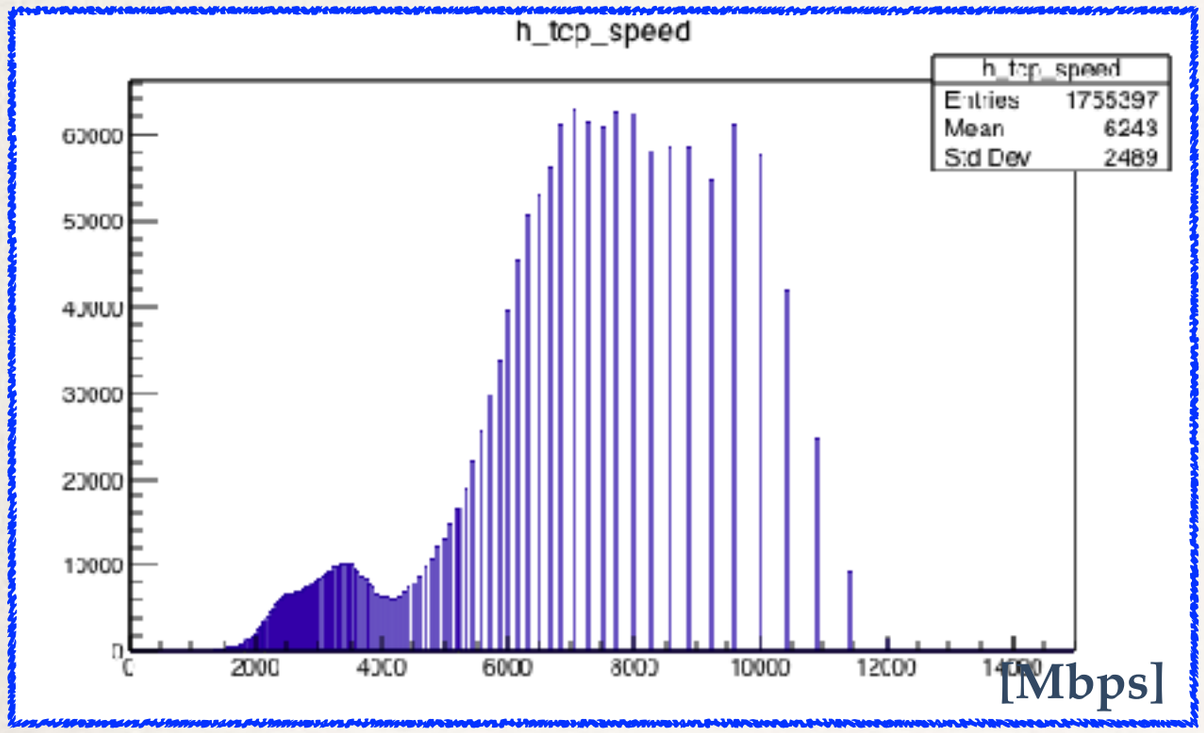


Test configuration 2

- 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



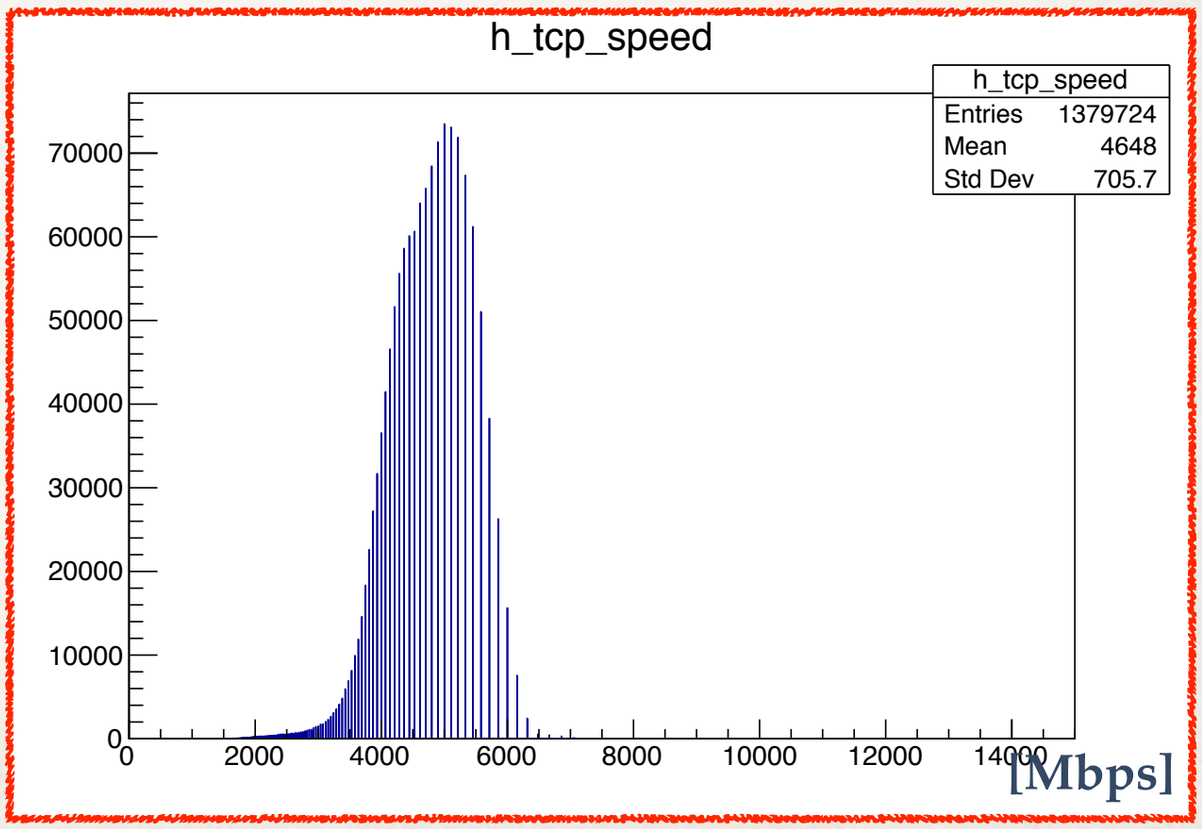
Test configuration 2



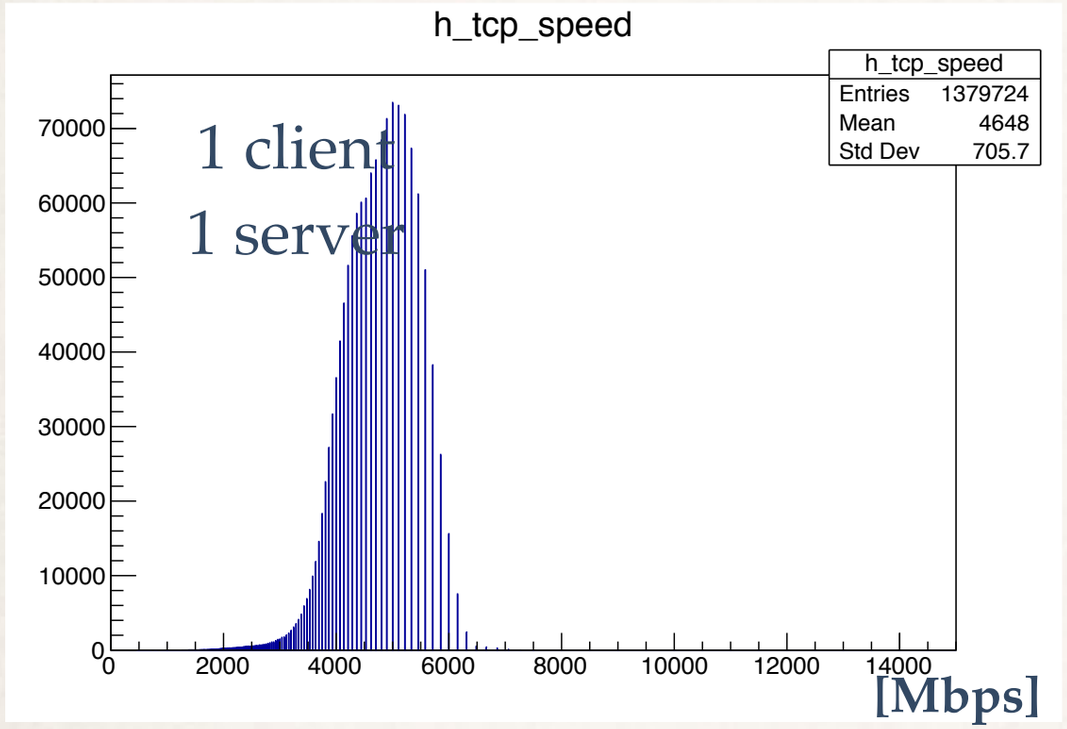
W/ HDD + offline analysis

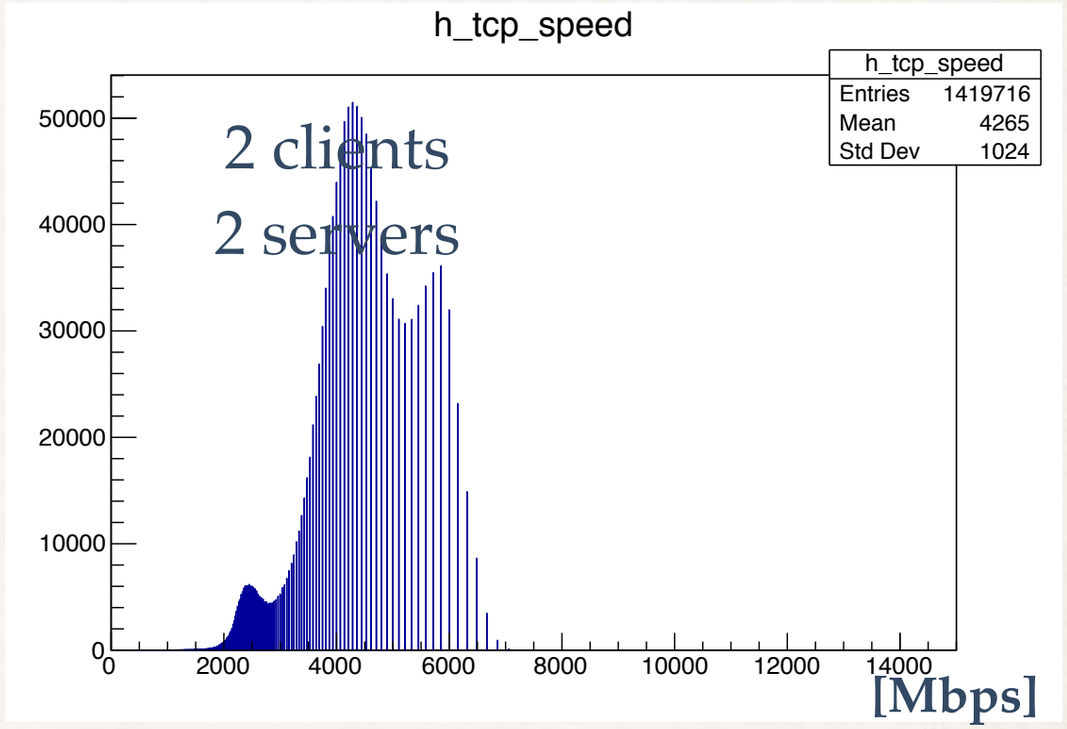
W/O HDD + online analysis

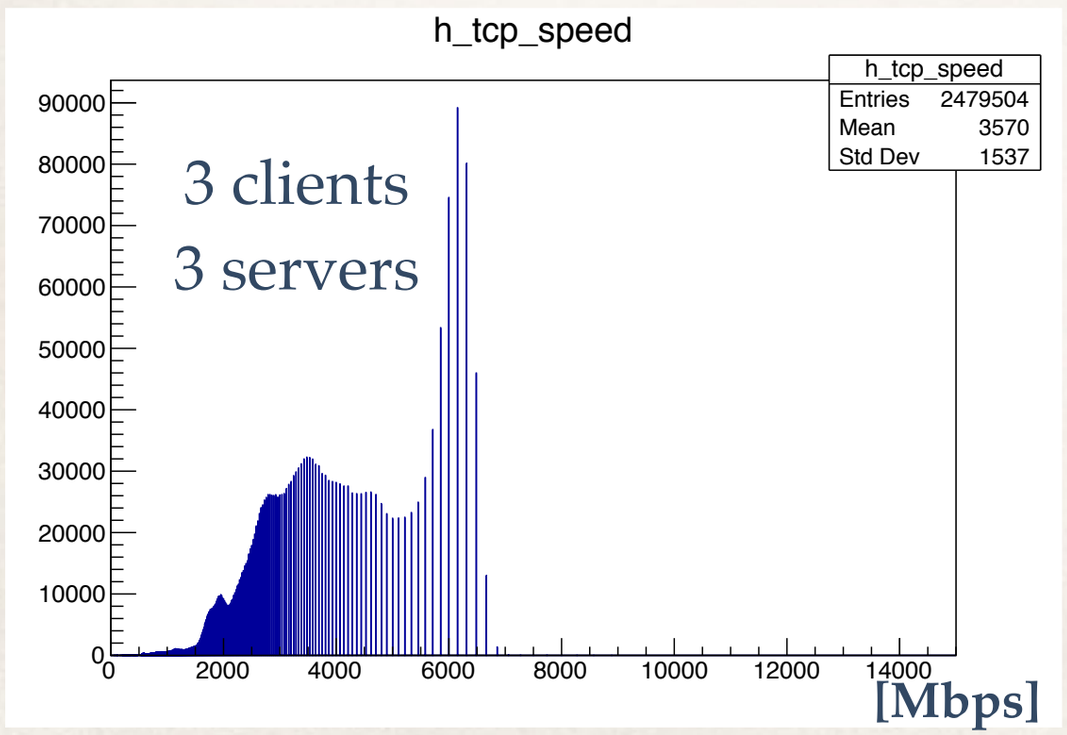
The same one server + one client setup;
keep this “offset” in mind.

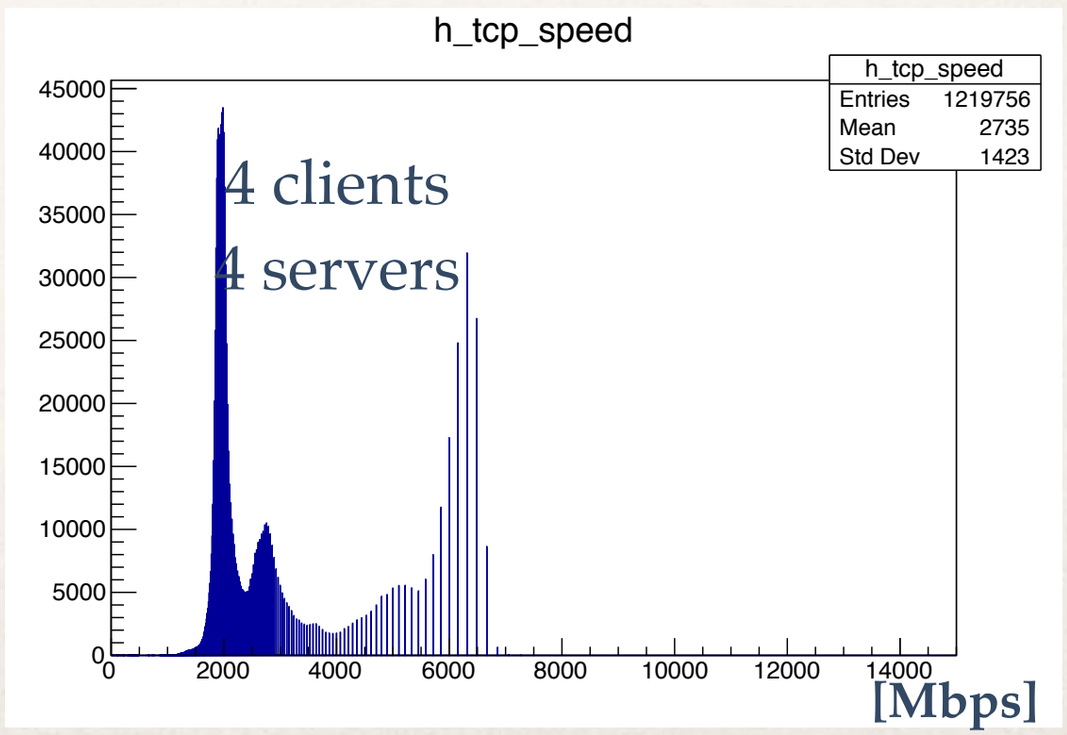


Test configuration 2

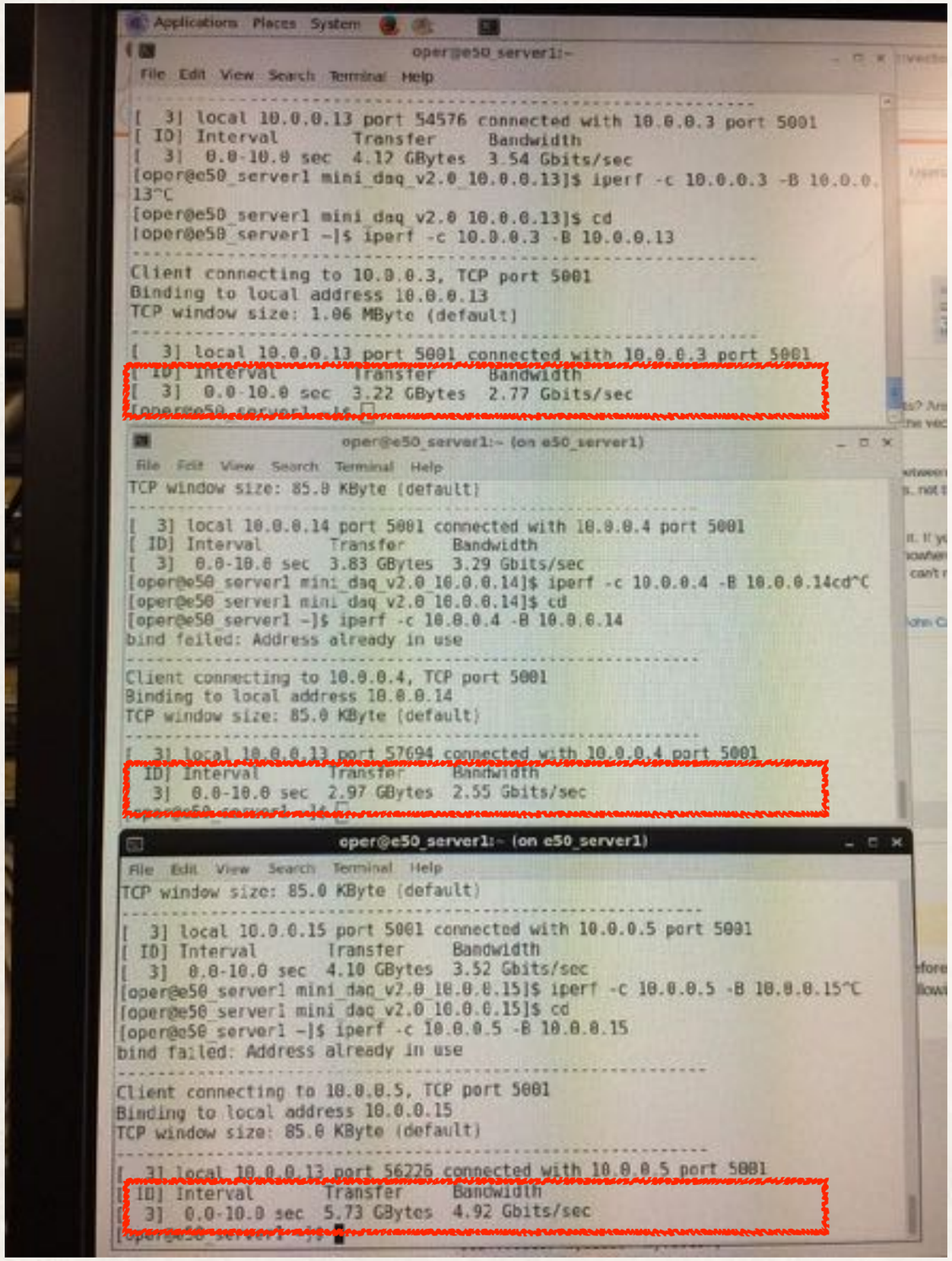








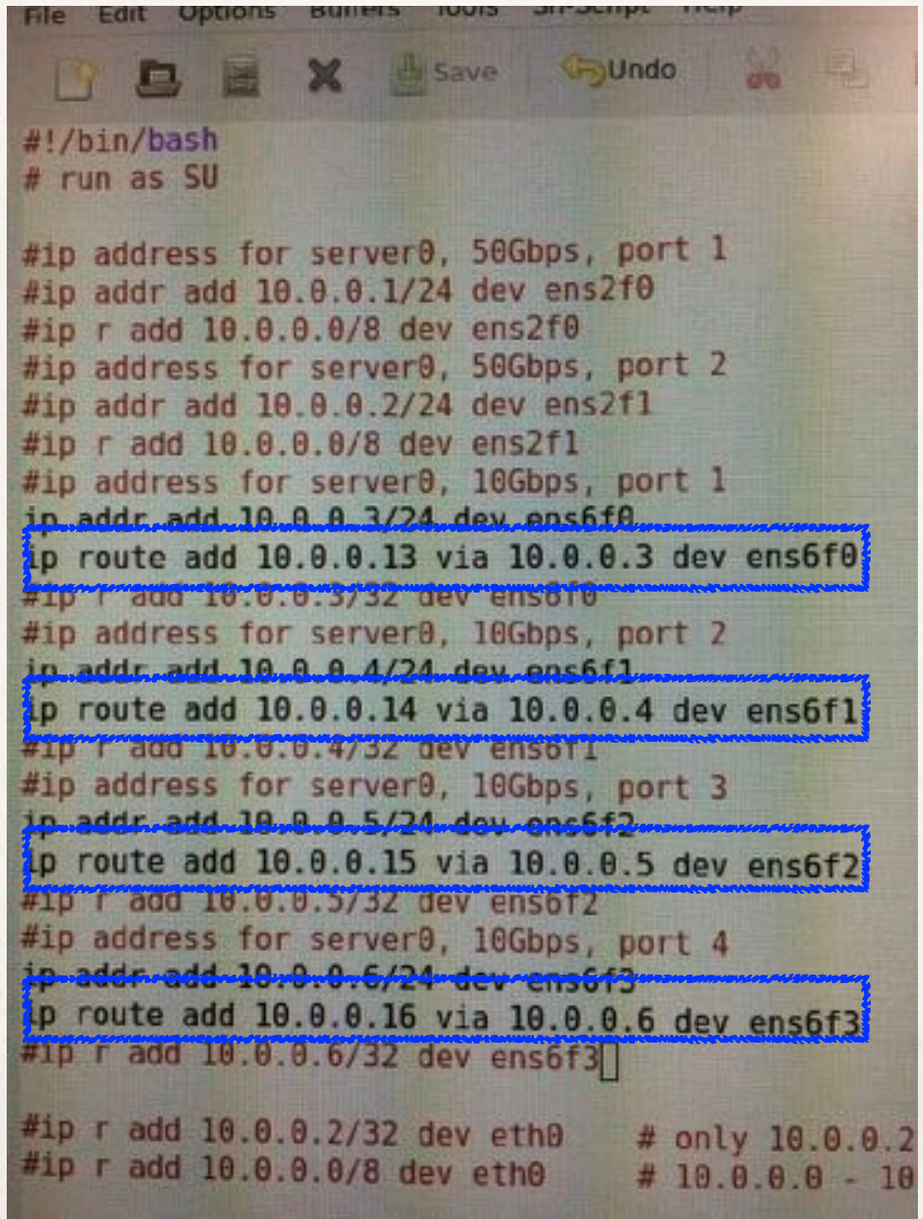
iperf gives similar speed drops

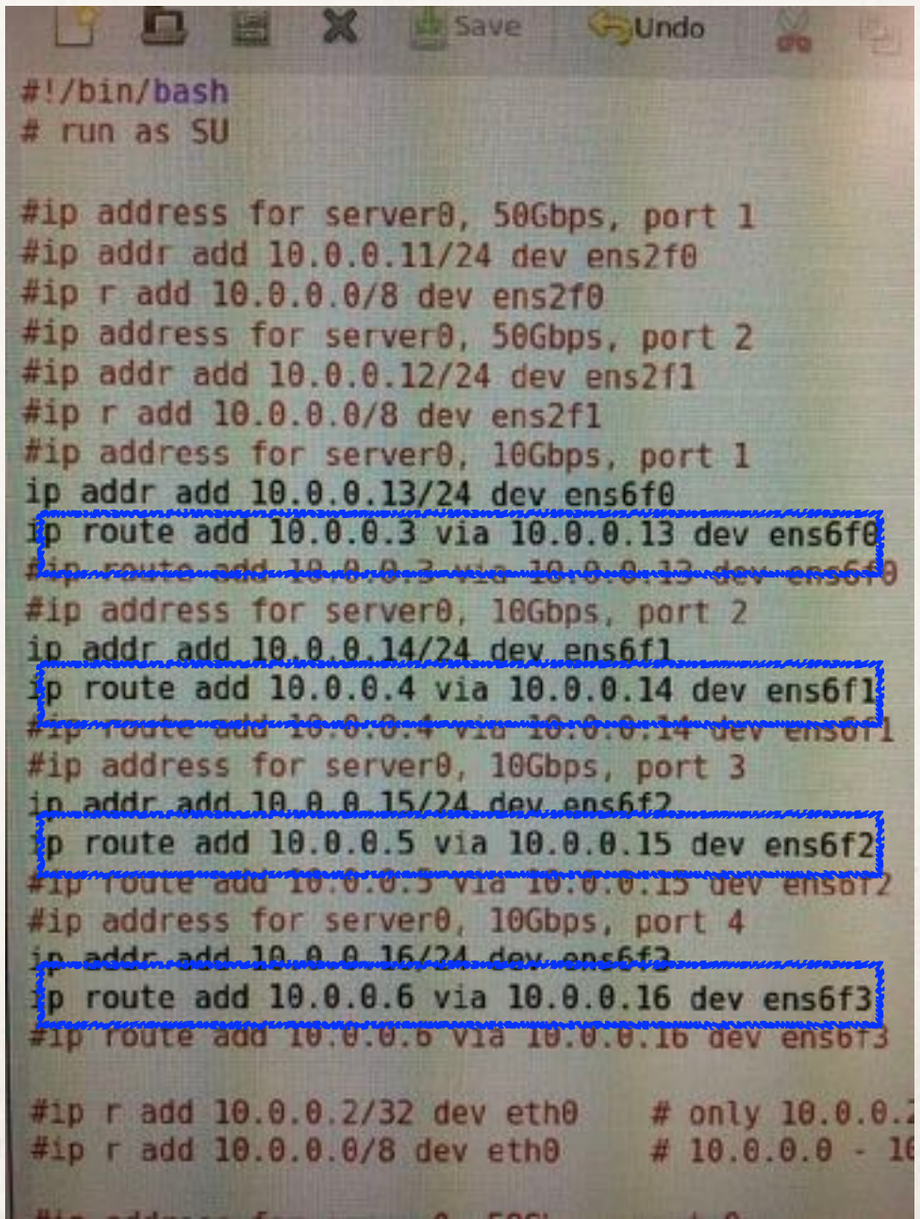


- 3 servers: iperf -s -B 10.0.0.3; …
- 3 clients: iperf -c 10.0.0.3 -B 10.0.0.13; …
- total throughput of 10Gbps

Update: solution found

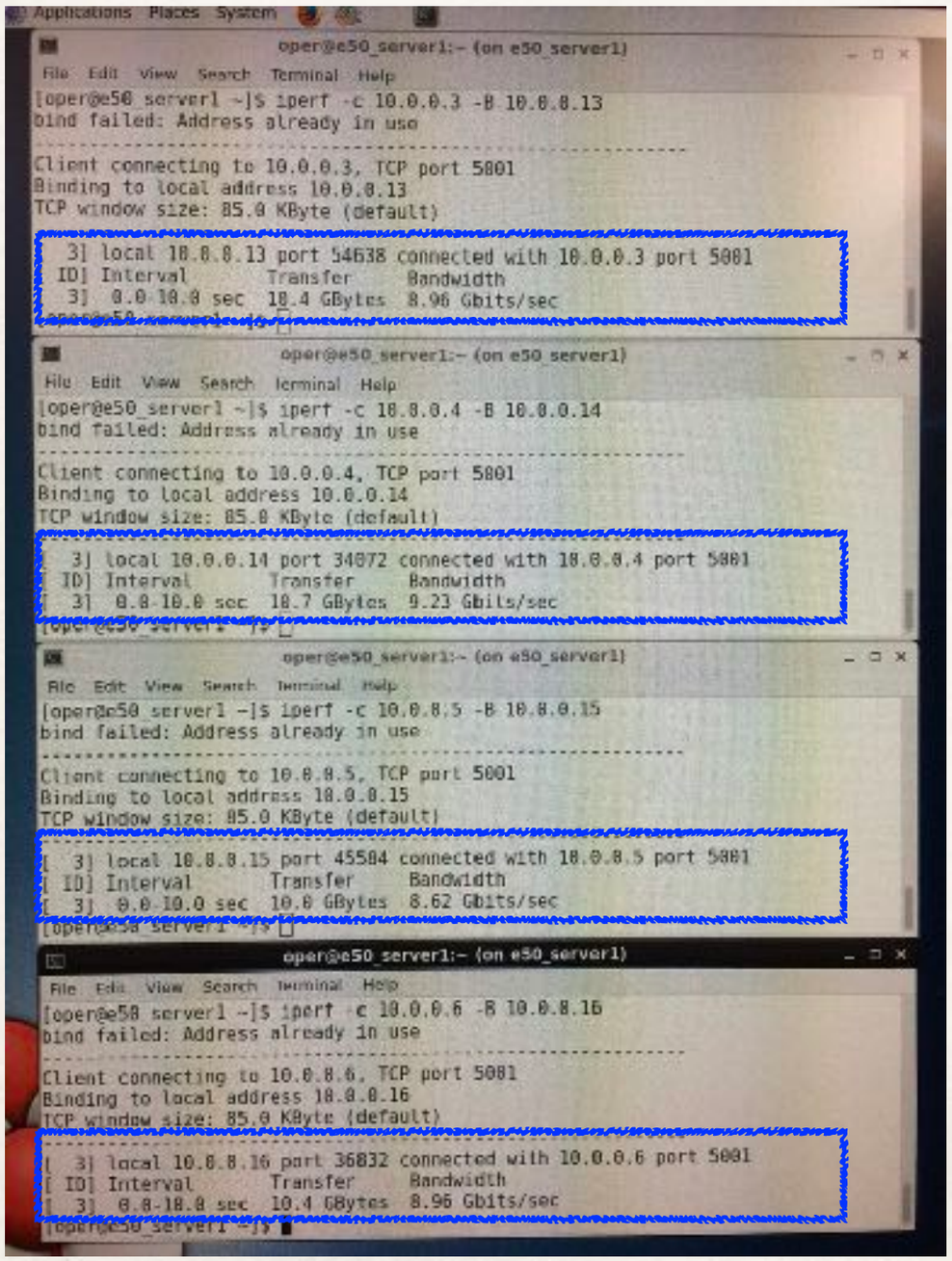
- Default route policy isn't smart enough for multiple ports on the same machine;
- Problem solved by assigning a static route to each pair of IP address



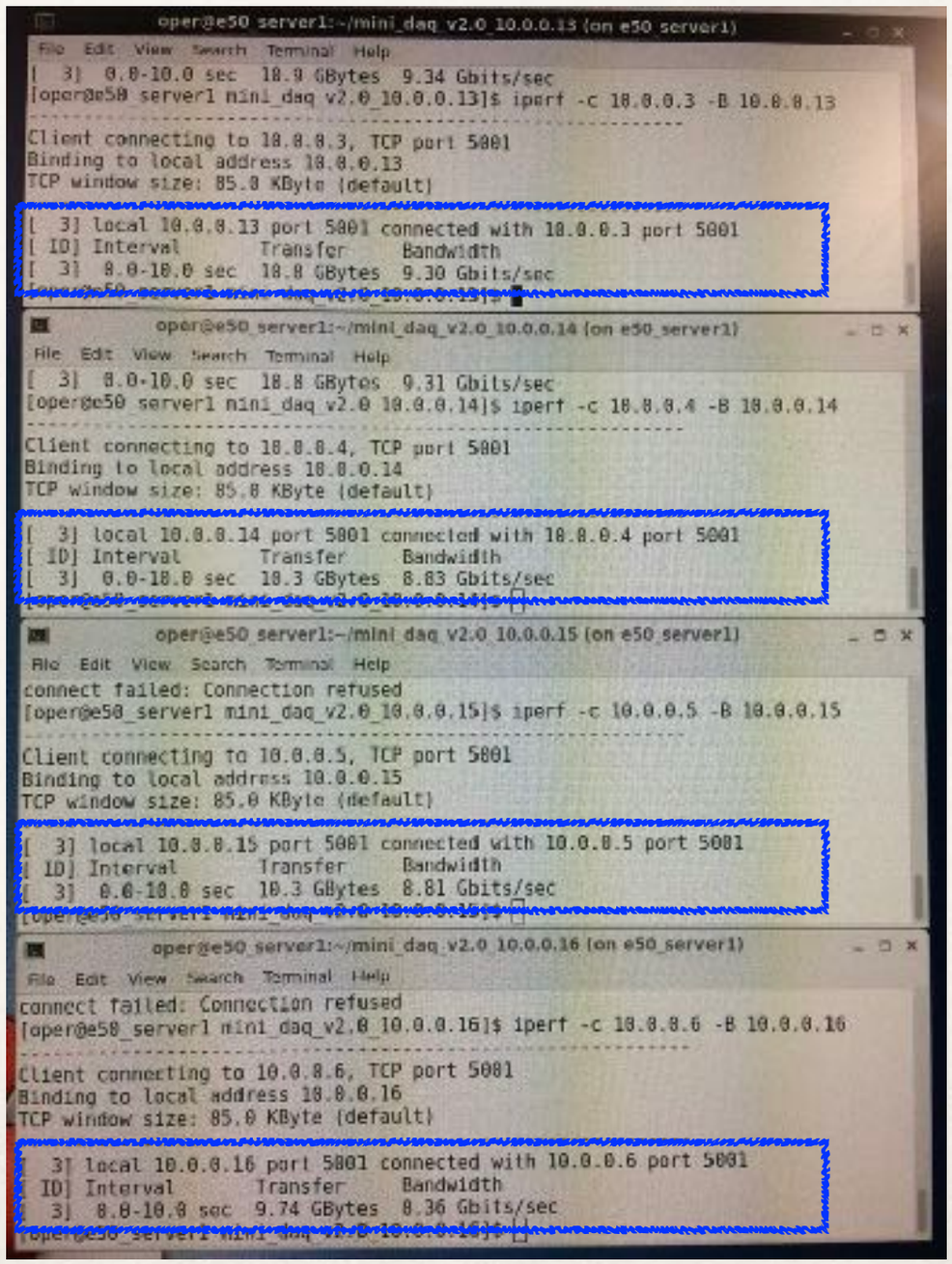


Update: solution found

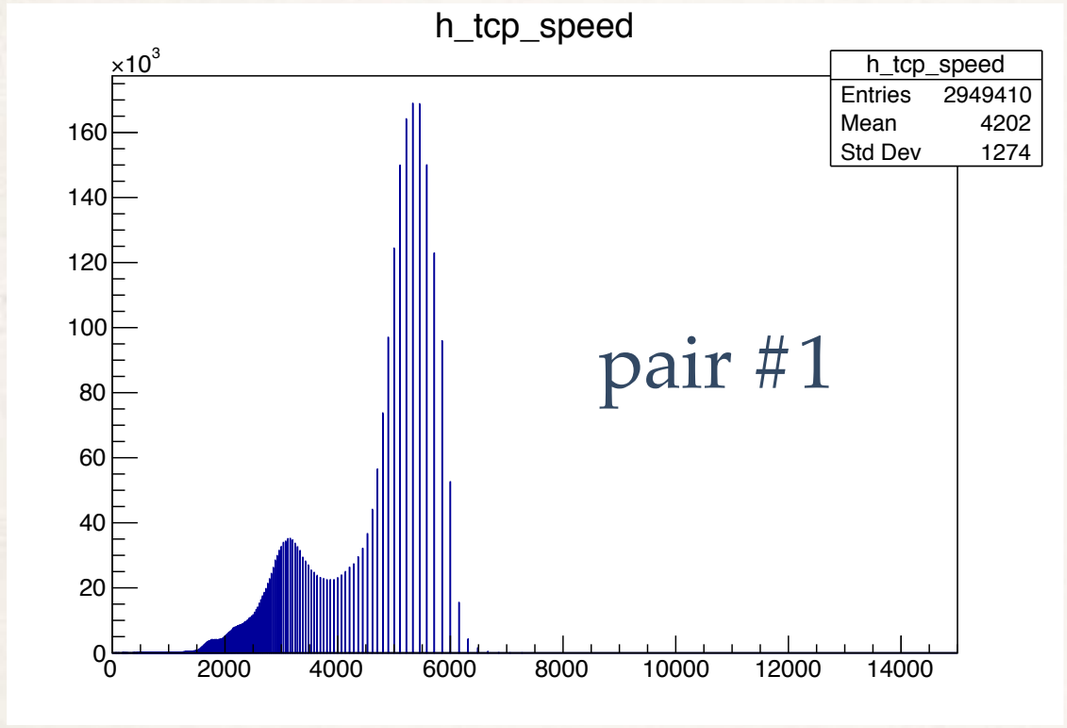
iperf results from
direct cable connection

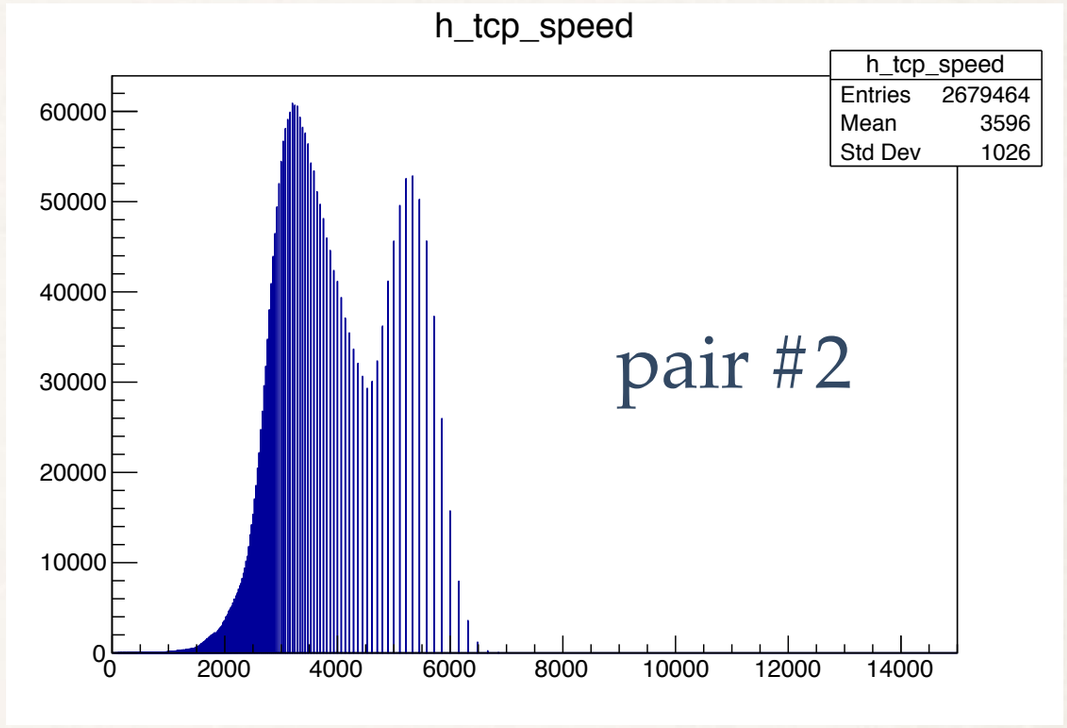


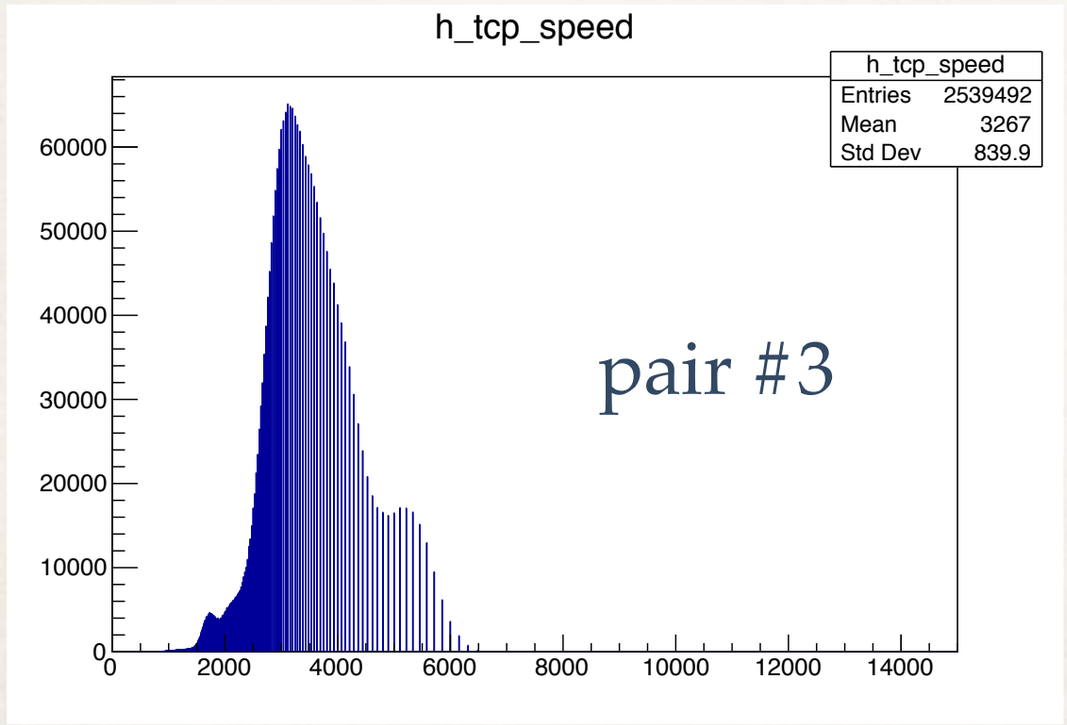
iperf results from
Cisco switch

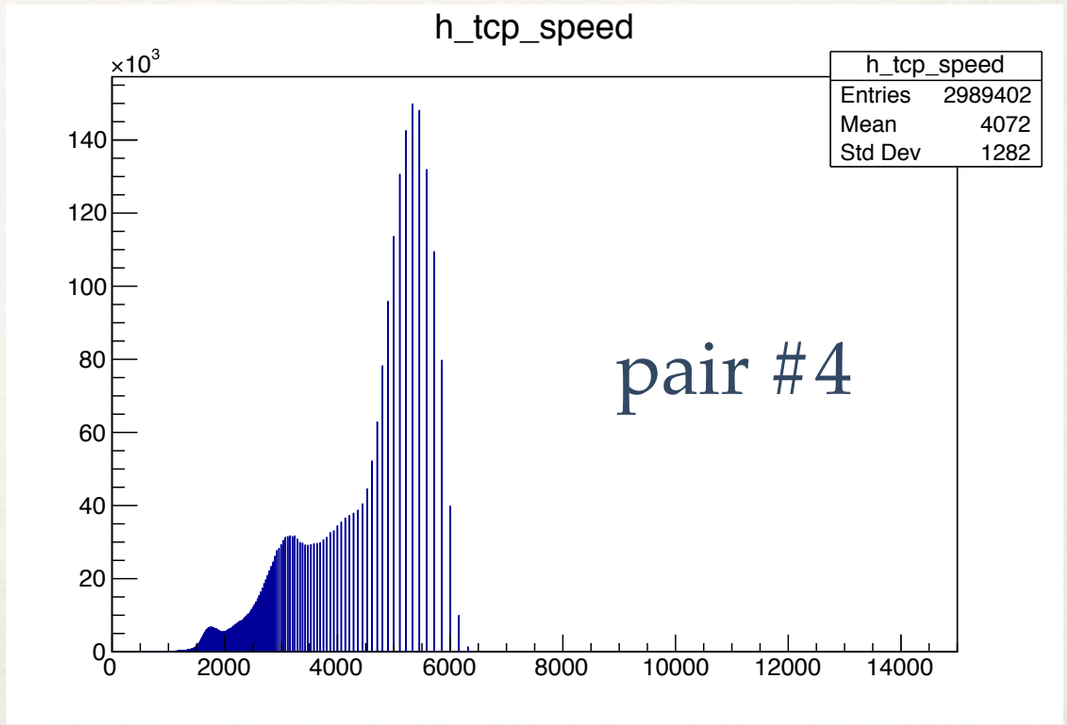


Update: solution found (direct connection)

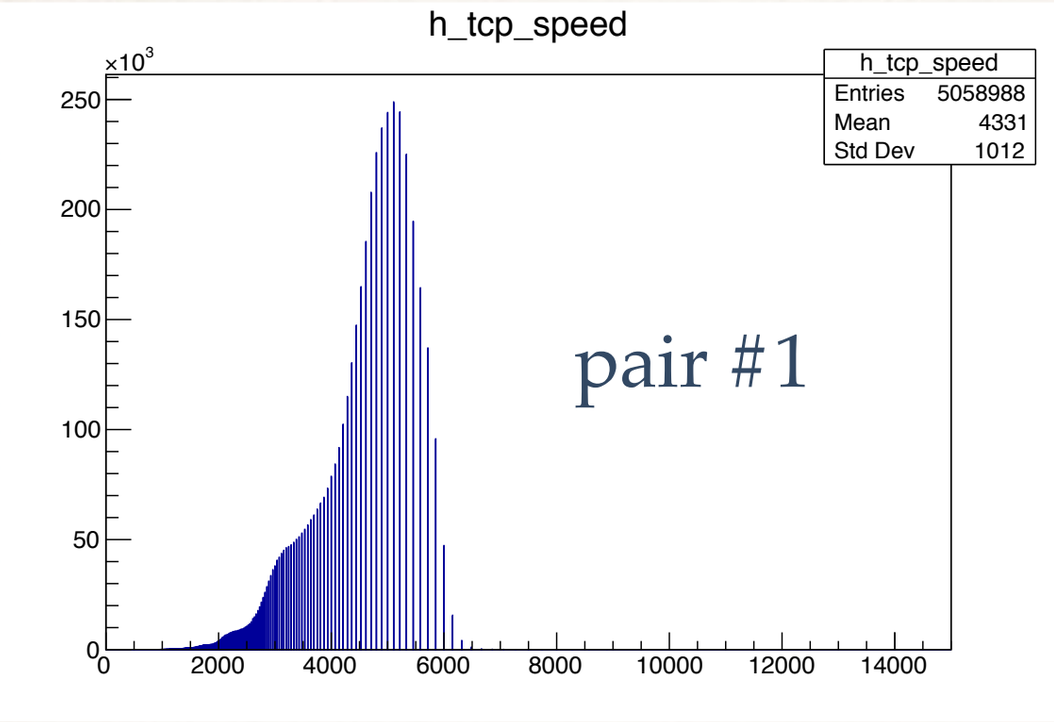


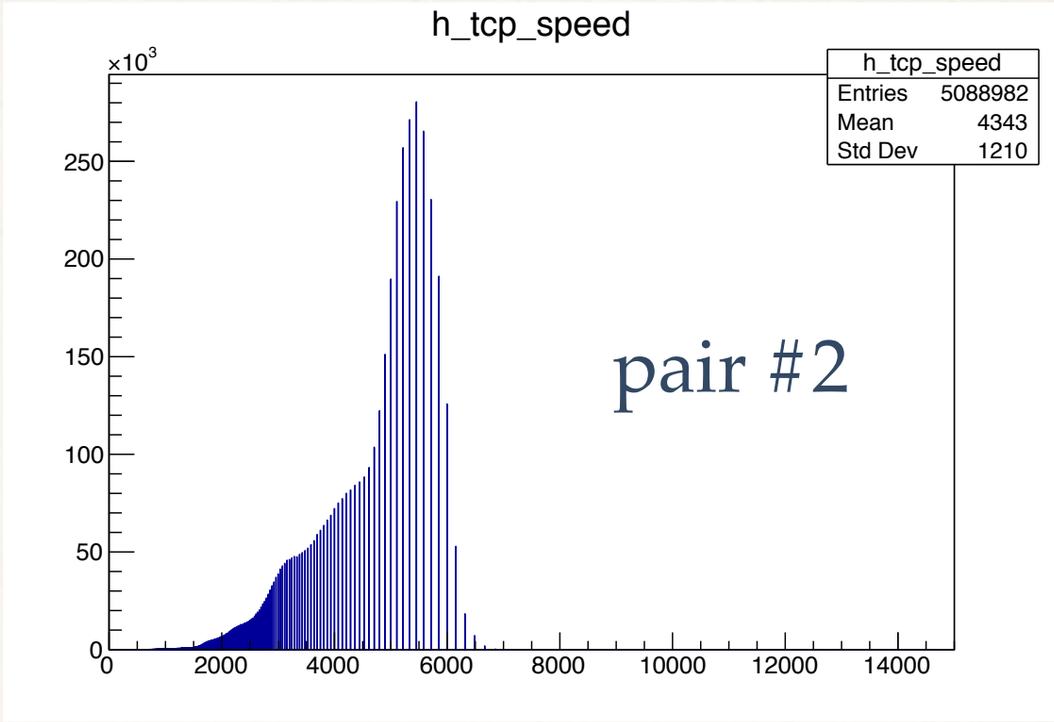


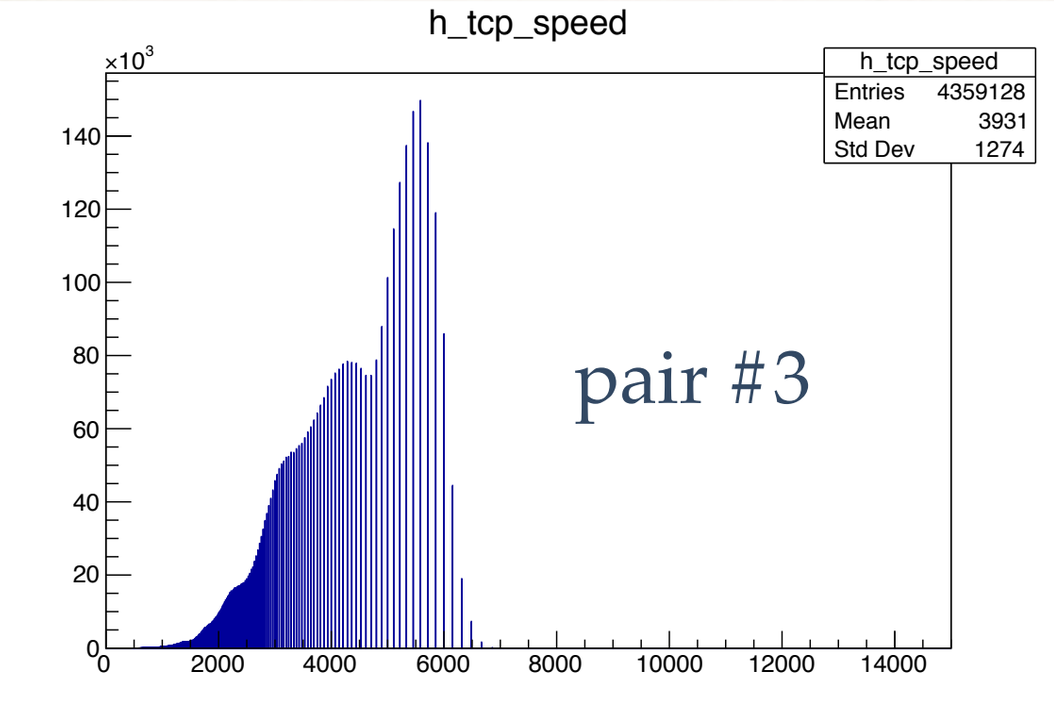


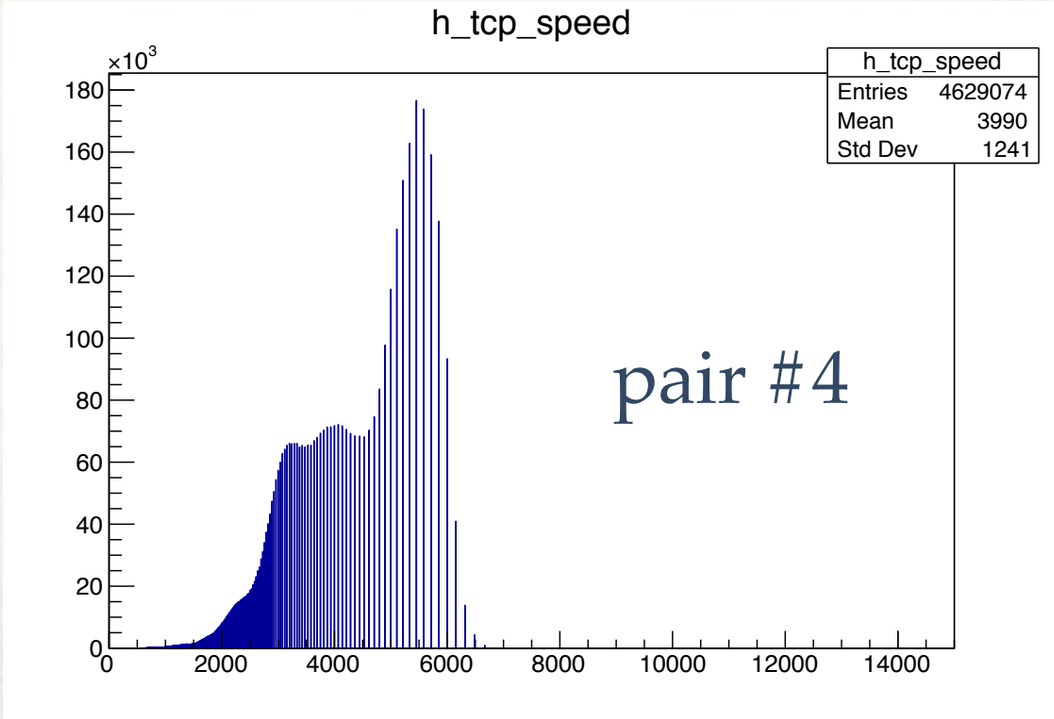


Update: solution found (Cisco switch)



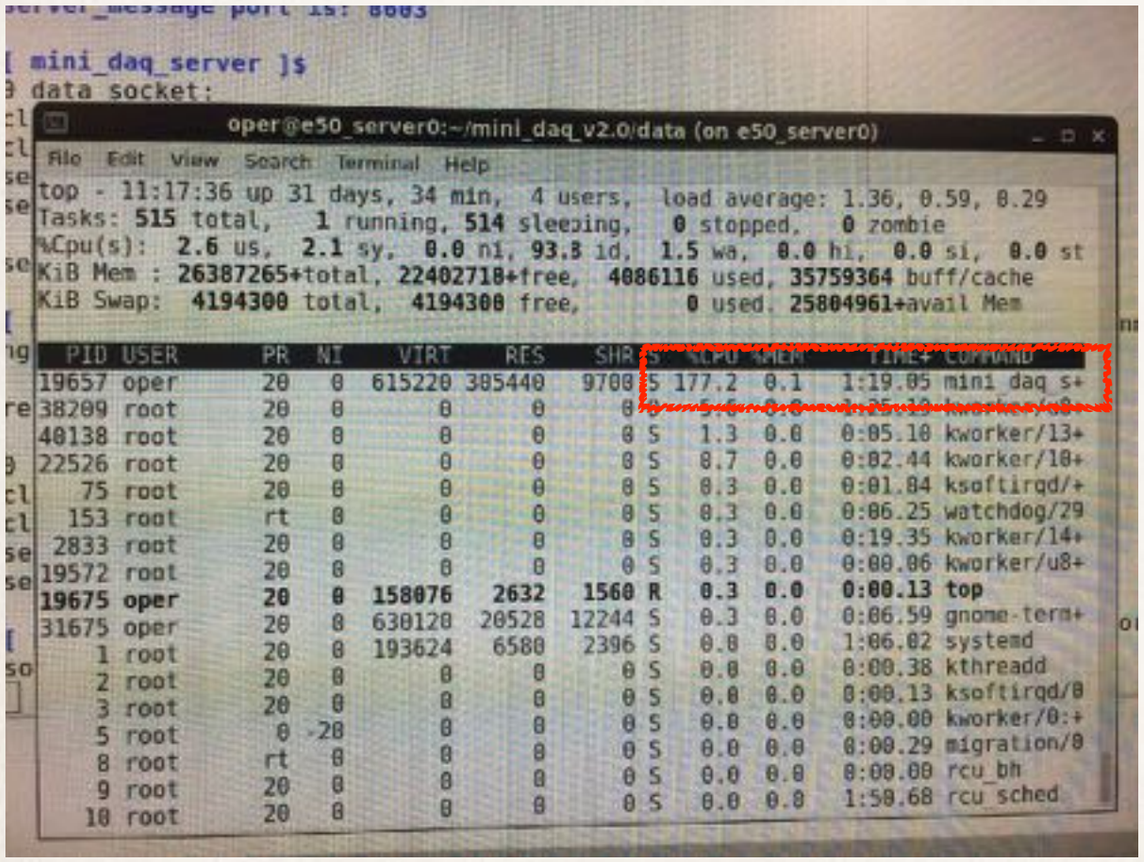




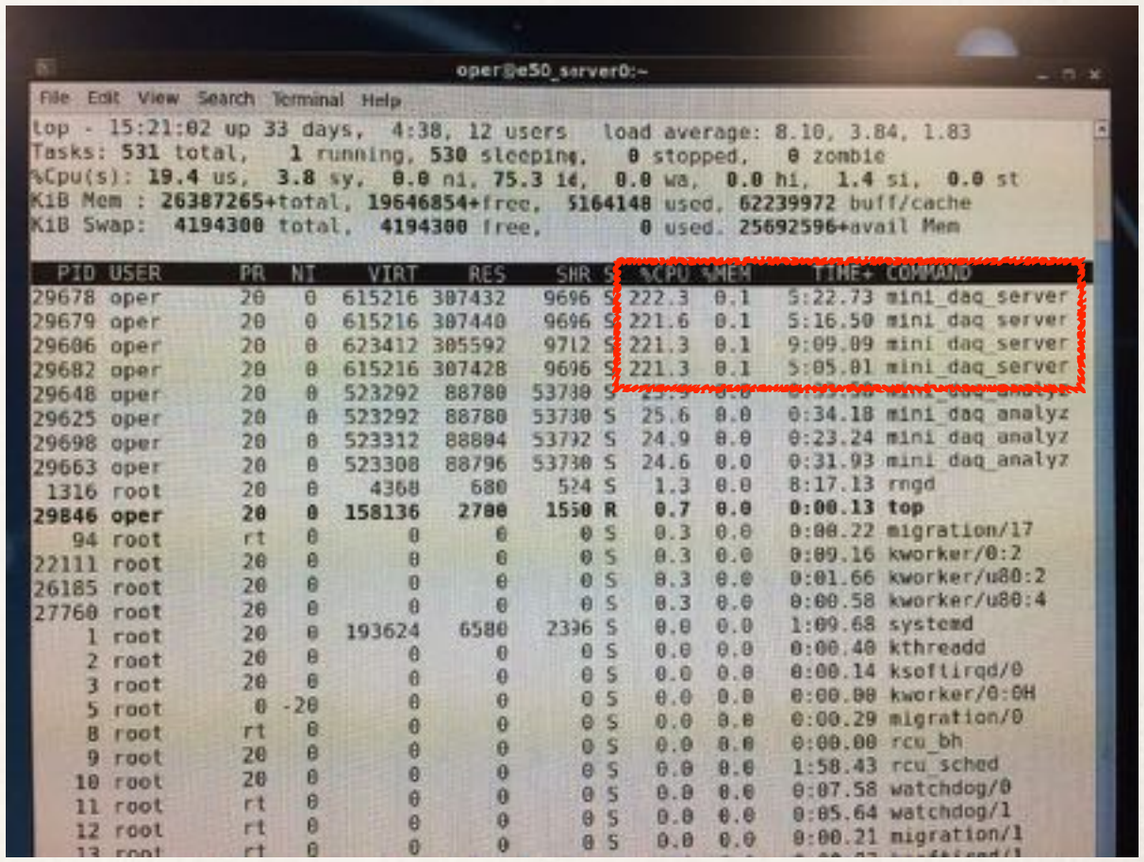


CPU usage

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



configuration 1;
at server side



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!