Iperf 2

Robert McMahon

March 2023

(as of version 2.1.9)

BROADCOM

Starting off with a joke

“There are three types of test & measurement engineers. Those who can count and those who can’t.”

BROADCOM

The promise

You'll be closer to understanding the difference between the speed of causality (latency) and link capacity (throughput) using end to end tooling, i.e. iperf 2, to measure things, including one way delays & responsiveness, of networking products & the networks themselves (speed, capacity, energy/distance)

BROADCOM

Agenda

- Get to Q&A as quick as possible
- Overview of iperf 2
- Benefits
- How to setup a rig
- Some example outputs
- Example traffic profiles
- How to automate & scale

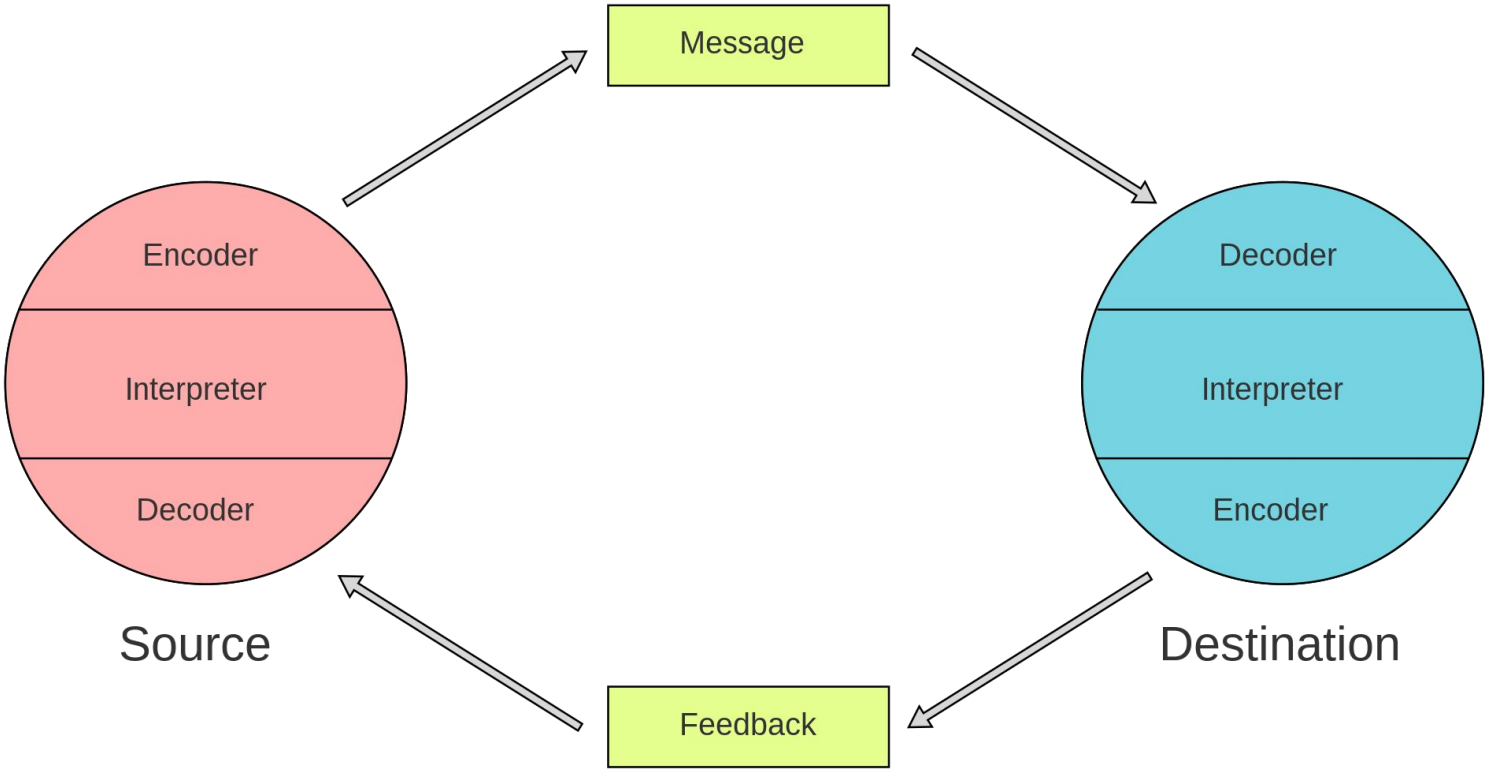
Iperf 2 - What is it?

- Iperf 2 is a free open source network performance tool based on BSD sockets
  - Measures network capacity and latency (or responsiveness)
  - Released as open source via sourceforge
  - End to end (socket calls)
    - or WiFi client to server
    - server to server
  - Generates synthetic traffic per different profiles
  - High performance thread based design
  - Multiple metrics include full histograms and central limit theorem (CLT) averaging
  - iperf2 can measure one-way latencies if a system is configured for clock sync (like Linux ptp)
- Broadcom took over maintenance in 2014
  - Bob McMahon writes most of the code, fixes most bugs & interfaces with the community per tickets & patches
  - Tim Auckland handles builds, some bugs, internal testing & validation and internal deployment
  - Kaushik Battu developed an android release
- A trusted open source network performance tool
  - not a Broadcom proprietary tool nor biased towards Broadcom
  - allowing for fair comparisons & wide adoption
  - At the BSD socket layer
- Supports most all operating systems
- ~2M lifetime downloads (650K per last year)
- The user base is quite large
  - from vwmare, AWS, etc
  - to NICs vendors, to WiFi chips
  - Basically anything that supports network i/o & berkeley sockets
- Closest commercial product from Ixia
  - Costs thousands dollars per license
  - Ixia NRE phase mostly over, feature set immutable

Sorry, about the lack of documentation

- Authoritative information is the source code
- Man page is somewhat maintained
- Would like somebody to write a chapter as part of a networking book

Communications Model (Schramm)



BROADCOM

Domains (for a T&M engineer)

- Metrics
- Mechanisms
- Methodologies

BROADCOM

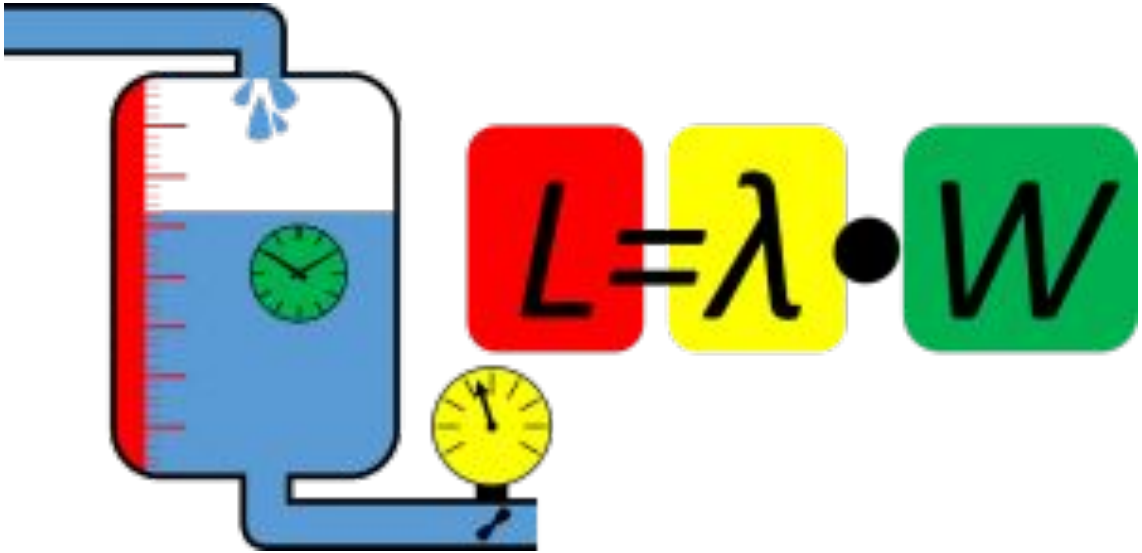
Benefits - New Metrics

- Full histograms - not just CLT averaging
- One way delays
  - UDP packets
  - Isochronous or video frame (both TCP & UDP)
  - Burst period and burst size, a la DASH
- Messages
  - TCP write to read
- Responses per second
- Jitter histograms
- 3WHS times
- TCP stack stats
  - samples tcp rtt
  - cwnd
  - Retries
- E2E queue depth via Little's Law
- E2E network power (per Jaffe)
- Scheduling errors
- Send side bloat mitigations
- Warnings when not i/o limited

Benefits - Traffic Based Mechanisms

- UDP & TCP throughput with -b write or read rate limiting
- UDP & TCP packets & messages
- UDP & TCP streaming isochronous (or streaming video profiles)
- TCP bounceback
- TCP periodic bursts (DASH)
- Working or concurrent loads
- Parallelism via threads
- Multicast both SM & SSM (IGMP/MLD versions controlled by kernel settings)

Little’s law (iperf 2 inP metric)



L items in a *stationary system* is equal to the long-term average effective arrival rate λ multiplied by the average time W that an item spends in the system.

BROADCOM

WiFi Latency Features Iperf2 can help verify

- WiFi scheduling messaging & link layer features
- MuMimo
- 802.11ax triggers
- OFDMA

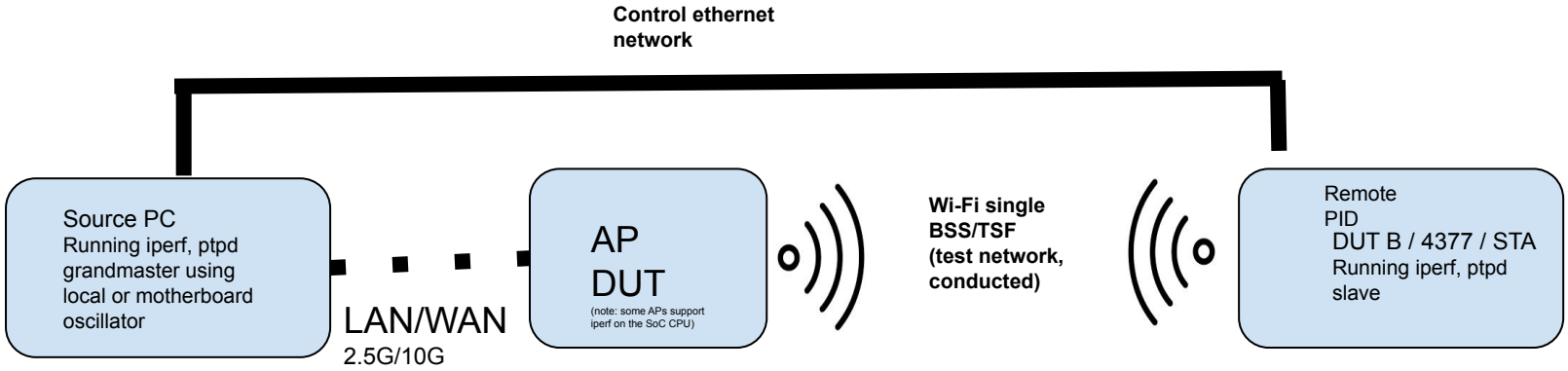
BROADCOM

Methodologies - pyflows facetime example

```
#instantiate DUT host and NIC devices

wifi1 = ssh_node(name='WiFi A', ipaddr=args.host_wifi1, device='eth1', devip='192.168.1.58')
wifi2 = ssh_node(name='WiFi_B', ipaddr=args.host_wifi2, device='eth1', devip='192.168.1.70')

video=iperf_flow(name='VIDEO_FACETIME_UDP', user='root', server=wifi2, client=wifi1, dstip=wifi2.devip, proto='UDP', interval=1, debug=False, srcip=wifi1.devip, srcport='6001', dstport='6001',
offered_load='30:600K',trip_times=True, tos='ac_vi', latency=True, fullduplex=True)

audio=iperf_flow(name='AUDIO_FACETIME_UDP', user='root', server=wifi2, client=wifi1, dstip=wifi2.devip, proto='UDP', interval=1, debug=False, srcip=wifi1.devip, srcport='6002', dstport='6002',
offered_load='50:25K',trip_times=True, tos='ac_vo', latency=True, fullduplex=True)


iperf_flow.run(time=args.time, flows='all', epoch_sync=True)
```


BROADCOM

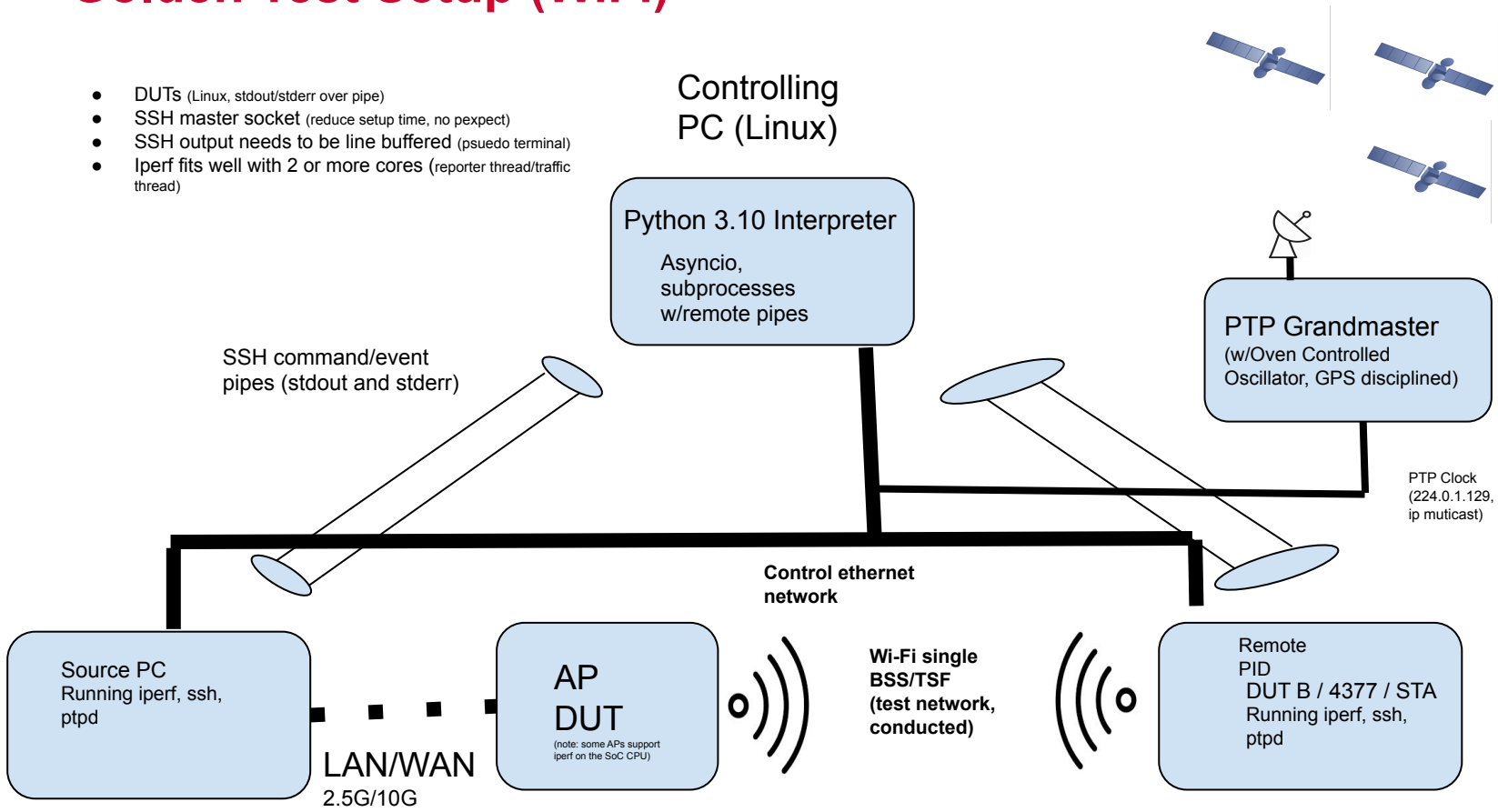
Basic setup (WiFi)



BROADCOM

Golden Test Setup (WiFi)

- DUTs (Linux, stdout/stderr over pipe)
- SSH master socket (reduce setup time, no pexpect)
- SSH output needs to be line buffered (psuedo terminal)
- Iperf fits well with 2 or more cores (reporter thread/traffic thread)



BROADCOM

Time Sync (enable OWD) –trip-times

- Use ptp4l & phc2sys for software or hardware timestamps
- Use ptpd2 for software timestamps
- GPS signal is a good reference
- Facebook's Open Time Appliance project
- Rasberry Pi with GPS hat & pulse per second
- Timebeat (haven't used them but seem interesting)

Unsynchronized detections

Unsynchronized clock detections with --bounceback and --trip-times (as of March 19, 2023): Iperf 2 can detect when the clocks have synchronization errors larger than the bounceback RTT. This is done via the client's send timestamp (clock A), the server's recieve timestamp (clock B) and the client's final receive timestamp (clock A.) The check, done on each bounceback, is write(A) < read(B) < read(A). This is supported in bounceback tests with a slight adjustment: clock write(A) < clock read(B) < clock read(A) - (clock write(B) - clock read(B)). All the timestamps are sampled on the initial write or read (not the completion of.) Error output looks as shown below and **there is no output for a zero value**.

[1] 0.00-10.00 sec Clock sync error count = 100

Server side outputs: UDP Basic OWD

```
[root@ctrl1fc35 iperf-2.1.9-rc2]# iperf -s -i 1 -u -e
------------------------------------------------------------
Server listening on UDP port 5001 with pid 4342
Read buffer size: 1.44 KByte (Dist bin width= 183 Byte)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.15%enp2s0 port 5001 connected with 192.168.1.234 port 39550 (trip-times) (sock=3) (peer 2.1.9-rc2) on
2023-03-08 11:30:52.458 (PST)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total  Latency avg/min/max/stdev PPS  Rx/inP  NetPwr
[  1] 0.00-1.00 sec   131 KBytes  1.07 Mbits/sec   0.122 ms 0/91 (0%) 2.306/0.788/5.321/1.209 ms 92 pps 91/0(0) pkts 58.00
[  1] 1.00-2.00 sec   128 KBytes  1.05 Mbits/sec   0.108 ms 0/89 (0%) 1.084/0.746/1.501/0.101 ms 89 pps 89/0(0) pkts 121
[  1] 2.00-3.00 sec   128 KBytes  1.05 Mbits/sec   0.091 ms 0/89 (0%) 1.088/0.712/1.730/0.110 ms 89 pps 89/0(0) pkts 120
[  1] 3.00-4.00 sec   128 KBytes  1.05 Mbits/sec   0.147 ms 0/89 (0%) 1.109/0.847/1.766/0.128 ms 89 pps 89/0(0) pkts 118
[  1] 4.00-5.00 sec   128 KBytes  1.05 Mbits/sec   0.087 ms 0/89 (0%) 1.086/0.778/1.610/0.108 ms 89 pps 89/0(0) pkts 120
[  1] 5.00-6.00 sec   128 KBytes  1.05 Mbits/sec   0.110 ms 0/89 (0%) 1.091/0.860/1.652/0.123 ms 89 pps 89/0(0) pkts 120
[  1] 6.00-7.00 sec   129 KBytes  1.06 Mbits/sec   0.116 ms 0/90 (0%) 1.124/0.823/2.155/0.175 ms 89 pps 90/0(0) pkts 118
[  1] 7.00-8.00 sec   128 KBytes  1.05 Mbits/sec   0.127 ms 0/89 (0%) 1.120/0.878/1.811/0.149 ms 89 pps 89/0(0) pkts 117
[  1] 8.00-9.00 sec   128 KBytes  1.05 Mbits/sec   0.120 ms 0/89 (0%) 1.088/0.688/1.789/0.126 ms 89 pps 89/0(0) pkts 120
[  1] 9.00-10.00 sec   128 KBytes  1.05 Mbits/sec   0.107 ms 0/89 (0%) 1.103/0.780/1.767/0.101 ms 89 pps 89/0(0) pkts 119
[  1] 0.00-10.02 sec  1.25 MBytes  1.05 Mbits/sec   0.119 ms 0/895 (0%) 1.222/0.688/5.321/0.543 ms 89 pps 895/0(0) pkts 107
```


BROADCOM

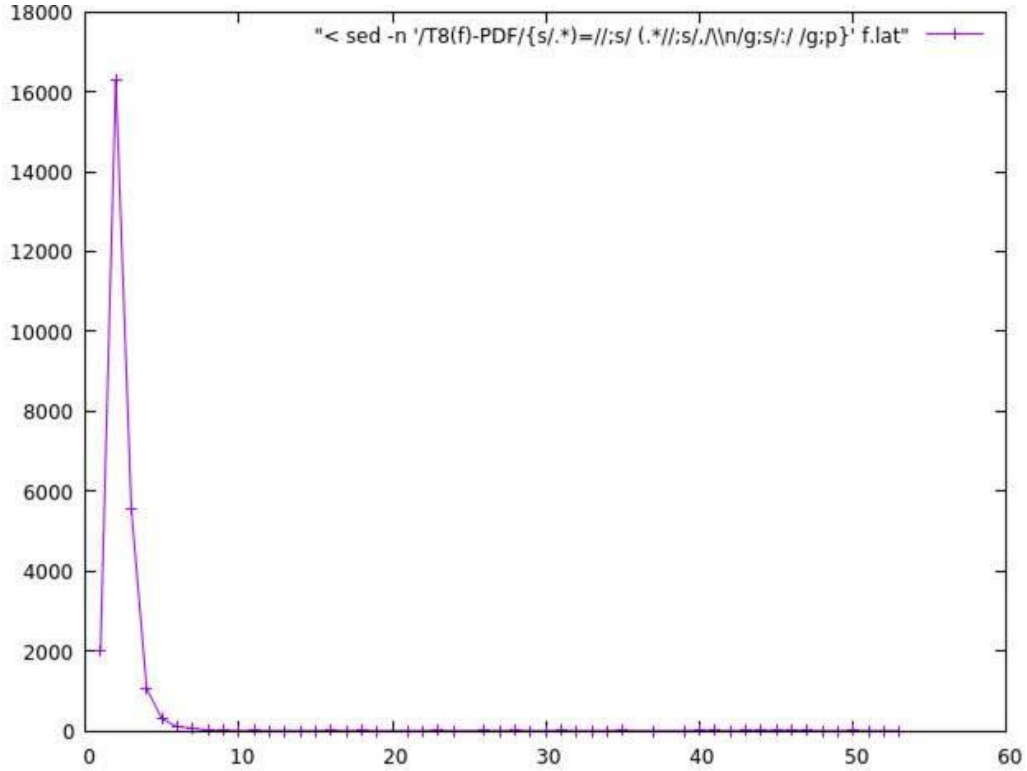
Server side outputs: UDP OWD with histograms

```
[root@ctrl1fc35 iperf-2.1.9-rc2]# iperf -s -i 1 -u -e --histograms
------------------------------------------------------------
Server listening on UDP port 5001 with pid 4355
Read buffer size: 1.44 KByte (Dist bin width= 183 Byte)
Enabled receive histograms bin-width=0.100 ms, bins=10000 (clients should use --trip-times)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.15%enp2s0 port 5001 connected with 192.168.1.234 port 38787 (trip-times) (sock=3) (peer 2.1.9-rc2) on 2023-03-08 11:34:48.808 (PST)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total  Latency avg/min/max/stdev PPS  Rx/inP  NetPwr
[  1] 0.00-1.00 sec   131 KBytes  1.07 Mbits/sec   0.082 ms 0/91 (0%) 1.113/0.906/1.779/0.121 ms 92 pps 91/0(0) pkts 120
[  1] 0.00-1.00 sec T8-PDF: bin(w=100us):cnt(91)=10:8,11:32,12:44,13:3,15:2,16:1,18:1 (5.00/95.00/99.7%=10/13/18,Outliers=2,obl/obu=0/0)
[  1] 1.00-2.00 sec   128 KBytes  1.05 Mbits/sec   0.063 ms 0/89 (0%) 1.101/0.667/1.788/0.107 ms 89 pps 89/0(0) pkts 119
[  1] 1.00-2.00 sec T8-PDF: bin(w=100us):cnt(89)=7:1,9:2,10:6,11:26,12:50,13:3,18:1 (5.00/95.00/99.7%=10/12/18,Outliers=0,obl/obu=0/0)
[  1] 2.00-3.00 sec   128 KBytes  1.05 Mbits/sec   0.082 ms 0/89 (0%) 1.112/0.866/1.631/0.114 ms 89 pps 89/0(0) pkts 118
[  1] 2.00-3.00 sec T8-PDF: bin(w=100us):cnt(89)=9:4,10:3,11:28,12:49,13:1,15:2,16:1,17:1 (5.00/95.00/99.7%=10/13/17,Outliers=2,obl/obu=0/0)
[  1] 3.00-4.00 sec   128 KBytes  1.05 Mbits/sec   0.077 ms 0/89 (0%) 1.090/0.834/1.522/0.097 ms 89 pps 89/0(0) pkts 120
[  1] 3.00-4.00 sec T8-PDF: bin(w=100us):cnt(89)=9:6,10:7,11:24,12:48,13:2,15:1,16:1 (5.00/95.00/99.7%=9/12/16,Outliers=0,obl/obu=0/0)
[  1] 4.00-5.00 sec   128 KBytes  1.05 Mbits/sec   0.085 ms 0/89 (0%) 1.098/0.781/1.689/0.134 ms 89 pps 89/0(0) pkts 119
[  1] 4.00-5.00 sec T8-PDF: bin(w=100us):cnt(89)=8:1,9:5,10:9,11:24,12:46,13:1,17:3 (5.00/95.00/99.7%=9/12/17,Outliers=0,obl/obu=0/0)
[  1] 5.00-6.00 sec   128 KBytes  1.05 Mbits/sec   0.114 ms 0/89 (0%) 1.093/0.794/1.476/0.111 ms 89 pps 89/0(0) pkts 120
[  1] 5.00-6.00 sec T8-PDF: bin(w=100us):cnt(89)=8:1,9:4,10:11,11:24,12:42,13:3,14:2,15:2 (5.00/95.00/99.7%=9/13/15,Outliers=0,obl/obu=0/0)
[  1] 6.00-7.00 sec   129 KBytes  1.06 Mbits/sec   0.080 ms 0/90 (0%) 1.088/0.674/1.507/0.133 ms 89 pps 90/0(0) pkts 122
[  1] 6.00-7.00 sec T8-PDF: bin(w=100us):cnt(90)=7:1,9:3,10:18,11:23,12:37,13:2,14:2,15:2,16:2 (5.00/95.00/99.7%=10/14/16,Outliers=0,obl/obu=0/0)
[  1] 7.00-8.00 sec   128 KBytes  1.05 Mbits/sec   0.134 ms 0/89 (0%) 1.120/0.816/1.836/0.167 ms 89 pps 89/0(0) pkts 117
[  1] 7.00-8.00 sec T8-PDF: bin(w=100us):cnt(89)=9:8,10:5,11:20,12:46,13:2,14:2,15:2,16:2,18:1,19:1 (5.00/95.00/99.7%=9/15/19,Outliers=0,obl/obu=0/0)
[  1] 8.00-9.00 sec   128 KBytes  1.05 Mbits/sec   0.071 ms 0/89 (0%) 1.097/0.880/1.503/0.090 ms 89 pps 89/0(0) pkts 119
[  1] 8.00-9.00 sec T8-PDF: bin(w=100us):cnt(89)=9:4,10:6,11:30,12:46,13:1,15:1,16:1 (5.00/95.00/99.7%=10/12/16,Outliers=0,obl/obu=0/0)
[  1] 9.00-10.00 sec   128 KBytes  1.05 Mbits/sec   0.126 ms 0/89 (0%) 1.108/0.854/1.587/0.114 ms 89 pps 89/0(0) pkts 118
[  1] 9.00-10.00 sec T8-PDF: bin(w=100us):cnt(89)=9:3,10:8,11:29,12:40,13:4,14:2,15:2,16:1 (5.00/95.00/99.7%=10/14/16,Outliers=0,obl/obu=0/0)
[  1] 0.00-10.02 sec  1.25 MBytes  1.05 Mbits/sec   0.102 ms 0/895 (0%) 1.102/0.667/1.836/0.120 ms 89 pps 895/0(0) pkts 119
[  1] 0.00-10.02 sec T8(f)-PDF: bin(w=100us):cnt(895)=7:2,8:2,9:39,10:81,11:262,12:448,13:22,14:8,15:14,16:9,17:4,18:3,19:1 (5.00/95.00/99.7%=10/13/18,Outliers=1,obl/obu=0/0)
```


BROADCOM

Example plot with gnuplot and sed

plot "< sed -n '/T8(f)-PDF/{s/.*)=//;s/ (.*//;s/,/\\n/g;s/:/ /g;p}' lat.txt" with lp

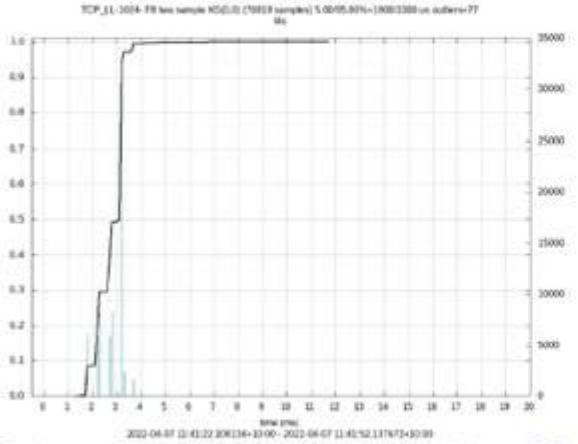


BROADCOM

PDF with CDF overlay

6888 L4S results (1)

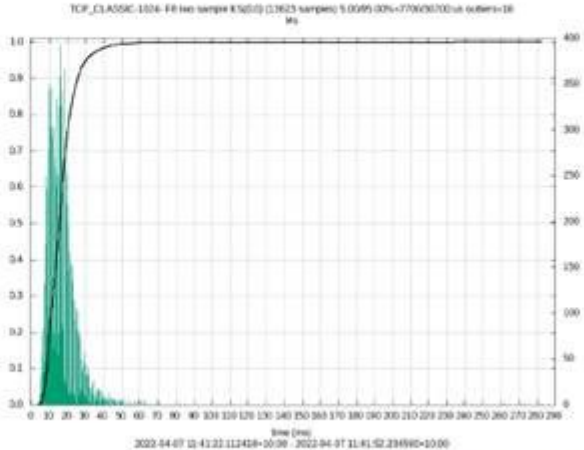
- 3 traffic streams directed into 100 Mbps WAN port
  - DCTCP + Classic TCP + 80 Mbps UDP
  - UDP throughput is 75 Mbps



DCTCP stream – TX rate 19.5-20.5 Mbps, Average Latency 2.5 ms

5% of the packets experiencing up to 1.8 ms latency

95% of the packets experiencing up to 3.3 ms latency



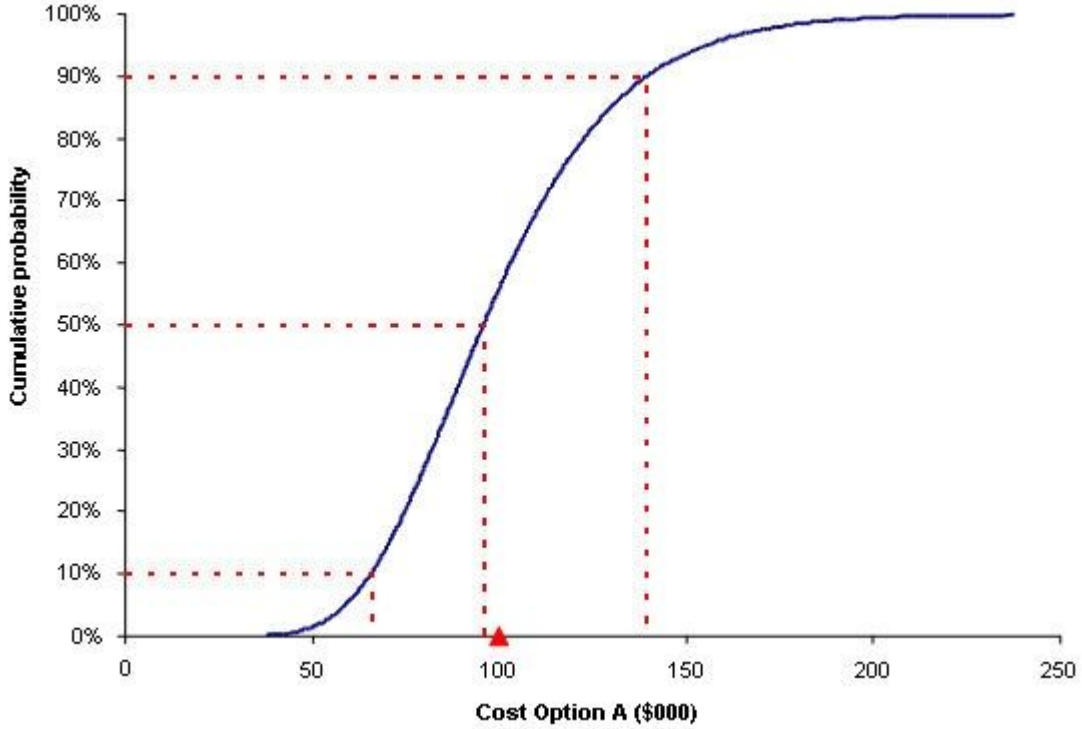
Classic TCP stream – TX rate 3-4 Mbps, Average Latency 18 ms

5% of the packets experiencing up to 7.7 ms latency

95% of the packets experiencing up to 30.7 ms latency

BROADCOM

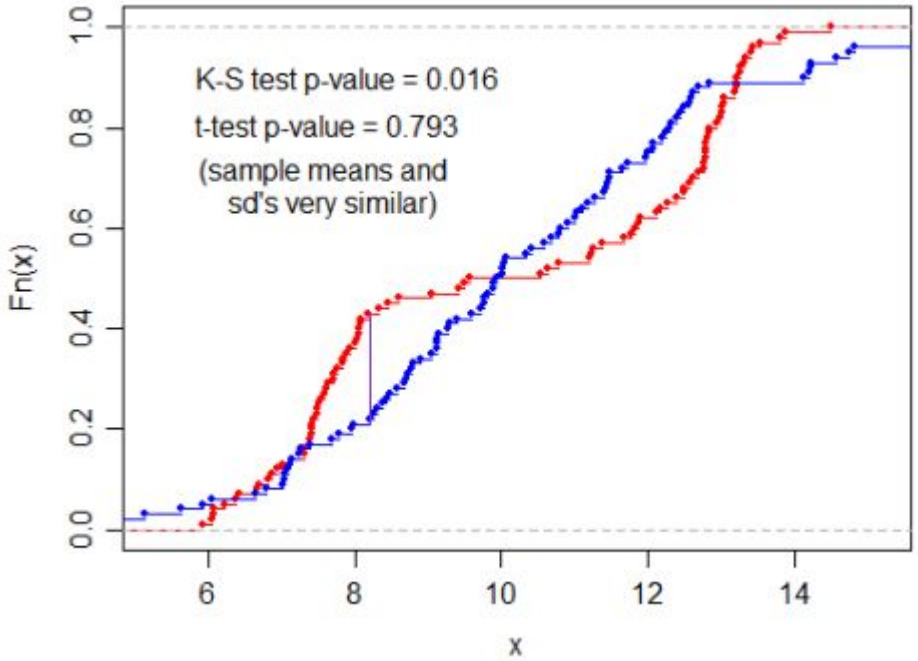
How to read a cumulative distribution function (CDF)



BROADCOM

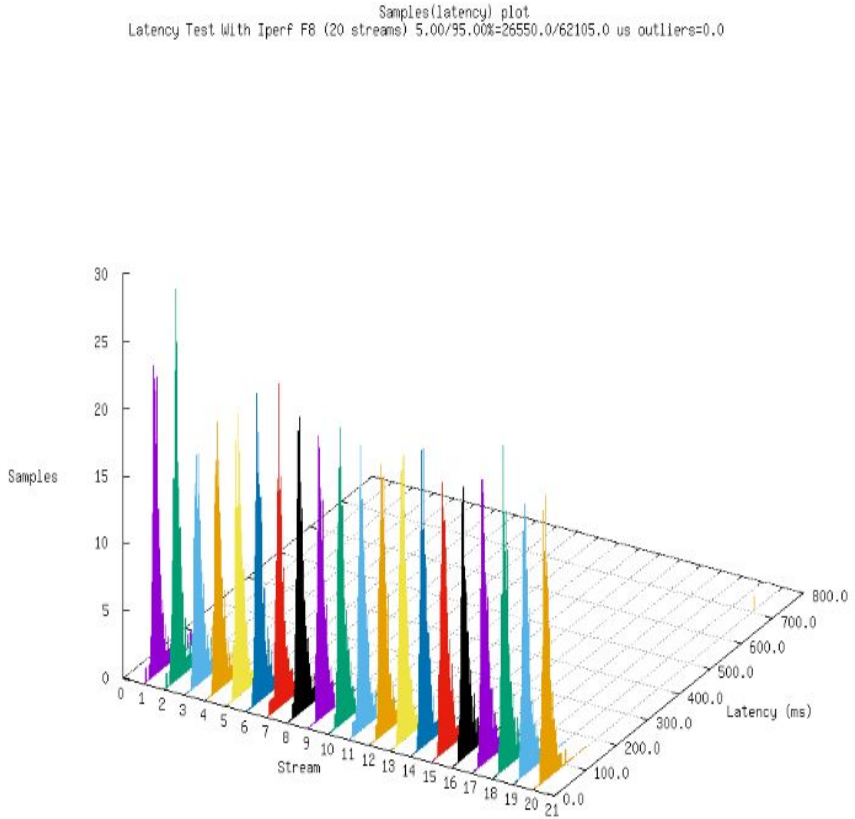
How to compare a CDF with Kolmogorov-Smirnov

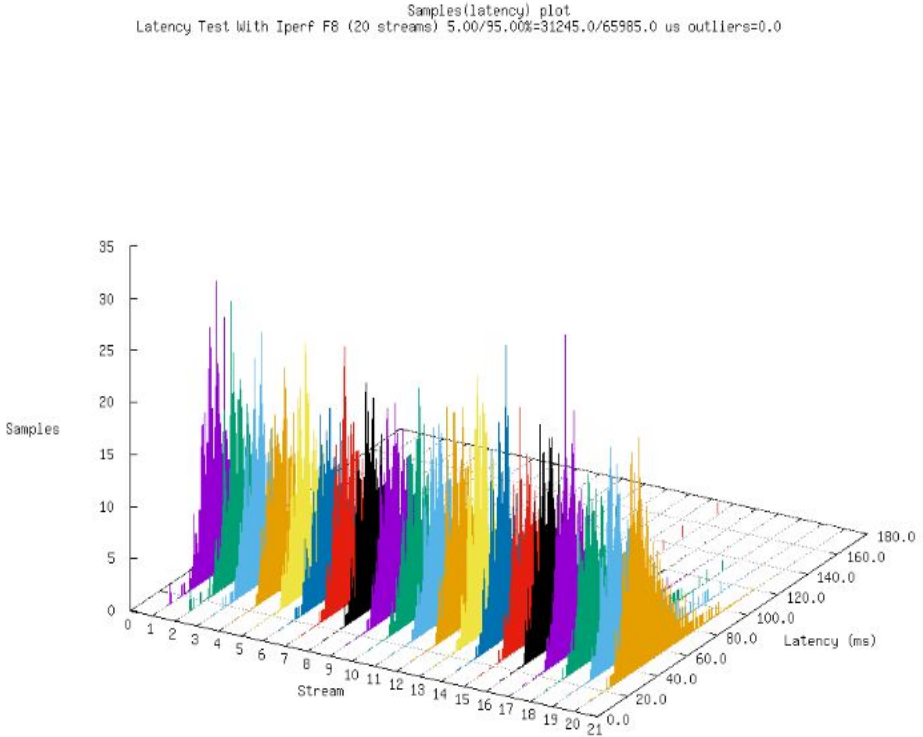
```
d,p = stats.ks_2samp(h1.samples, h2.samples)
```



BROADCOM

Multiple flows (-P 20)





BROADCOM

Server side outputs: UDP Isochronous

```
[root@ctrl1fc35 iperf-2.1.9-rc2]# iperf -s -i 1 -u -e --histograms
------------------------------------------------------------
Server listening on UDP port 5001 with pid 4386
Read buffer size: 1.44 KByte (Dist bin width= 183 Byte)
Enabled receive histograms bin-width=0.100 ms, bins=10000 (clients should use --trip-times)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.15%enp2s0 port 5001 connected with 192.168.1.234 port 35613 (isoch) (trip-times) (sock=3) (peer 2.1.9-rc2) on 2023-03-08 11:48:53.951 (PST)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Latency avg/min/max/stdev  PPS  Frame:rx/lost  Frame-latency avg/min/max/stdev  NetPwr
[  1] 0.00-1.00 sec  2.39 MBytes  20.0 Mbits/sec   0.029 ms 0/1741 (0%) 1.225/0.808/2.020/0.242 ms 1768 pps   61/0  1.777/1.414/2.234/0.244 ms 2042
[  1] 0.00-1.00 sec T8-PDF: bin(w=100us):cnt(1741)=9:47,10:122,11:463,12:492,13:132,14:125,15:73,16:75,17:88,18:83,19:27,20:12,21:2 (5.00/95.00/99.7%=10/18/20,Outliers=0,obl/obu=0/0)
[  1] 0.00-1.00 sec F8-PDF: bin(w=100us):cnt(60)=15:6,16:15,17:9,18:4,19:8,20:4,21:4,22:6,23:4 (5.00/95.00/99.7%=15/23/23,Outliers=0,obl/obu=0/0) (2.234 ms/1678304934.819126)
[  1] 1.00-2.00 sec  2.38 MBytes  20.0 Mbits/sec   0.035 ms 0/1740 (0%) 1.197/0.796/2.161/0.213 ms 1740 pps   60/0  1.753/1.389/2.586/0.246 ms 2089
[  1] 1.00-2.00 sec T8-PDF: bin(w=100us):cnt(1740)=8:1,9:35,10:114,11:485,12:569,13:118,14:161,15:88,16:55,17:69,18:9,19:12,20:10,22:14 (5.00/95.00/99.7%=10/17/22,Outliers=0,obl/obu=0/0)
[  1] 1.00-2.00 sec F8-PDF: bin(w=100us):cnt(60)=14:1,15:8,16:19,17:4,18:4,19:12,20:3,21:3,22:3,23:1,24:1,26:1 (5.00/95.00/99.7%=15/23/26,Outliers=0,obl/obu=0/0) (2.586
ms/1678304935.619446)
[  1] 2.00-3.00 sec  2.38 MBytes  20.0 Mbits/sec   0.015 ms 0/1740 (0%) 1.194/0.745/1.823/0.199 ms 1741 pps   60/0  1.745/1.286/2.586/0.239 ms 2094
[  1] 2.00-3.00 sec T8-PDF: bin(w=100us):cnt(1740)=8:10,9:28,10:126,11:431,12:595,13:139,14:160,15:54,16:76,17:82,18:29,19:10 (5.00/95.00/99.7%=10/17/19,Outliers=0,obl/obu=0/0)
[  1] 2.00-3.00 sec F8-PDF: bin(w=100us):cnt(60)=13:1,14:1,15:2,16:23,17:7,18:5,19:8,20:4,21:3,22:4,23:1,24:1 (5.00/95.00/99.7%=15/22/24,Outliers=0,obl/obu=0/0) (2.305
ms/1678304936.652457)
[  1] 0.00-3.00 sec  7.16 MBytes  20.0 Mbits/sec   0.021 ms 0/5223 (0%) 1.205/0.745/2.161/0.219 ms 1740 pps  181/0  1.745/1.286/2.586/0.239 ms 2075
[  1] 0.00-3.00 sec T8(f)-PDF: bin(w=100us):cnt(5223)=8:11,9:110,10:362,11:1380,12:1657,13:389,14:446,15:215,16:206,17:239,18:121,19:49,20:22,21:2,22:14
(5.00/95.00/99.7%=10/17/21,Outliers=0,obl/obu=0/0)
[  1] 0.00-3.00 sec F8(f)-PDF: bin(w=100us):cnt(180)=13:1,14:2,15:16,16:57,17:20,18:13,19:28,20:11,21:10,22:13,23:6,24:2,26:1 (5.00/95.00/99.7%=15/23/26,Outliers=0,obl/obu=0/0) (2.586
ms/1678304935.619446)
```


BROADCOM

Client side: —bounceback

```
[rjmcmahon@ryzen3950 ~]$ iperf -c 192.168.1.72 --bounceback --trip-times
------------------------------------------------------------
Client connecting to 192.168.1.72, TCP port 5001 with pid 172305 (1 flows)
Write buffer size:  100 Byte
Bursting:  100 Byte writes 10 times every 1.00 second(s)
Bounce-back test (size= 100 Byte) (server hold req=0 usecs & tcp_quickack)
TOS set to 0x0 and nodelay (Nagle off)
TCP window size: 16.0 KByte (default)
Event based writes (pending queue watermark at 16384 bytes)
------------------------------------------------------------
[  1] local 192.168.1.69%enp4s0 port 58454 connected with 192.168.1.72 port 5001 (prefetch=16384) (bb w/quickack len/hold=100/0) (trip-times) (sock=3)
(icwnd/mss/irtt=14/1448/261) (ct=0.35 ms) on 2023-03-15 09:13:13.912 (PDT)
[ ID] Interval       Transfer    Bandwidth        BB cnt=avg/min/max/stdev        Rtry  Cwnd/RTT    RPS
[  1] 0.00-1.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.489/0.126/3.504/1.060 ms    0   14K/157 us    2045 rps
[  1] 1.00-2.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.191/0.143/0.372/0.079 ms    0   14K/157 us    5247 rps
[  1] 2.00-3.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.203/0.118/0.335/0.056 ms    0   14K/175 us    4921 rps
[  1] 3.00-4.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.217/0.160/0.402/0.070 ms    0   14K/182 us    4598 rps
[  1] 4.00-5.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.220/0.153/0.390/0.068 ms    0   14K/181 us    4554 rps
[  1] 5.00-6.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.229/0.157/0.403/0.071 ms    0   14K/191 us    4357 rps
[  1] 6.00-7.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.235/0.164/0.431/0.075 ms    0   14K/192 us    4252 rps
[  1] 7.00-8.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.234/0.129/0.451/0.092 ms    0   14K/192 us    4270 rps
[  1] 8.00-9.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.229/0.158/0.418/0.075 ms    0   14K/192 us    4376 rps
[  1] 9.00-10.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.229/0.150/0.437/0.082 ms    0   14K/191 us    4374 rps
[  1] 0.00-10.01 sec  19.5 KBytes  16.0 Kbits/sec    100=0.248/0.118/3.504/0.337 ms    0   14K/1413 us    4039 rps
[  1] 0.00-10.01 sec  OWD Delays (ms) Cnt=100 To=0.141/0.054/0.339/0.058 From=0.074/0.035/0.120/0.018 Asymmetry=0.068/0.006/0.228/0.046    4039 rp
[  1] 0.00-10.01 sec BB8(f)-PDF: bin(w=100us):cnt(100)=2:44,3:46,4:3,5:6,36:1 (5.00/95.00/99.7%=2/5/36,Outliers=1,obl/obu=0/0)
```


BROADCOM

Client side: —bounceback –working-load

```
[[rjmcmahon@ryzen3950 ~]$ iperf -c 192.168.1.72 --bounceback --trip-times --working-load=up,4 -t 3
------------------------------------------------------------
Client connecting to 192.168.1.72, TCP port 5001 with pid 172405 (1 flows)
Write buffer size:  100 Byte
Bursting:  100 Byte writes 10 times every 1.00 second(s)
Bounce-back test (size= 100 Byte) (server hold req=0 usecs & tcp_quickack)
TOS set to 0x0 and nodelay (Nagle off)
TCP window size: 16.0 KByte (default)
Event based writes (pending queue watermark at 16384 bytes)
------------------------------------------------------------
[  3] local 192.168.1.69%enp4s0 port 40752 connected with 192.168.1.72 port 5001 (prefetch=16384) (trip-times) (sock=3) (qack) (icwnd/mss/irtt=14/1448/335) (ct=0.37 ms) on 2023-03-15 09:14:24.337 (PDT)
[  2] local 192.168.1.69%enp4s0 port 40754 connected with 192.168.1.72 port 5001 (prefetch=16384) (trip-times) (sock=6) (qack) (icwnd/mss/irtt=14/1448/345) (ct=0.40 ms) on 2023-03-15 09:14:24.337 (PDT)
[  4] local 192.168.1.69%enp4s0 port 40782 connected with 192.168.1.72 port 5001 (prefetch=16384) (trip-times) (sock=7) (qack) (icwnd/mss/irtt=14/1448/349) (ct=0.38 ms) on 2023-03-15 09:14:24.337 (PDT)
[  1] local 192.168.1.69%enp4s0 port 40780 connected with 192.168.1.72 port 5001 (prefetch=16384) (bb w/quickack len/hold=100/0) (trip-times) (sock=5) (icwnd/mss/irtt=14/1448/349) (ct=0.38 ms) on 2023-03-15 09:14:24.337 (PDT)
[  5] local 192.168.1.69%enp4s0 port 40756 connected with 192.168.1.72 port 5001 (prefetch=16384) (trip-times) (sock=4) (qack) (icwnd/mss/irtt=14/1448/351) (ct=0.40 ms) on 2023-03-15 09:14:24.337 (PDT)
[ ID] Interval        Transfer    Bandwidth       Write/Err  Rtry     Cwnd/RTT(var)        NetPwr
[  2] 0.00-1.00 sec  21.9 MBytes   184 Mbits/sec  229797/0        43      383K/7315(87) us  3141
[  5] 0.00-1.00 sec  28.4 MBytes   238 Mbits/sec  297612/0        42      530K/7326(45) us  4062
[ ID] Interval        Transfer    Bandwidth        BB cnt=avg/min/max/stdev        Rtry  Cwnd/RTT    RPS
[  1] 0.00-1.00 sec  1.95 KBytes  16.0 Kbits/sec    10=0.400/0.148/2.171/0.624 ms    0   14K/218 us    2501 rps
[  4] 0.00-1.00 sec  26.6 MBytes   223 Mbits/sec  278786/0        24      486K/7410(100) us  3762
[  3] 0.00-1.00 sec  28.1 MBytes   236 Mbits/sec  295131/0        48      616K/7472(14) us  3950
[SUM] 0.00-1.00 sec  83.1 MBytes   697 Mbits/sec  871529/0       114
[  2] 1.00-2.00 sec  17.1 MBytes   144 Mbits/sec  179521/0         0      466K/8299(73) us  2163
[  5] 1.00-2.00 sec  28.5 MBytes   239 Mbits/sec  298349/0         0      578K/8280(87) us  3603
[  1] 1.00-2.00 sec  1.95 KBytes  16.0 Kbits/sec    10=7.460/7.074/7.716/0.213 ms    0   14K/5557 us    134 rps
[  4] 1.00-2.00 sec  27.4 MBytes   230 Mbits/sec  287291/0         0      533K/8304(86) us  3460
[  3] 1.00-2.00 sec  31.0 MBytes   260 Mbits/sec  325159/0         0      671K/8291(78) us  3922
[SUM] 1.00-2.00 sec  86.9 MBytes   729 Mbits/sec  910799/0         0
[  2] 2.00-3.00 sec  21.6 MBytes   181 Mbits/sec  226274/0         0      514K/10321(31) us  2192
[  5] 2.00-3.00 sec  28.3 MBytes   237 Mbits/sec  296399/0         0      608K/10116(82) us  2930
[  1] 2.00-3.00 sec  1.95 KBytes  16.0 Kbits/sec    10=9.001/8.282/9.436/0.401 ms    0   14K/8110 us    111 rps
[  4] 2.00-3.00 sec  26.2 MBytes   220 Mbits/sec  274454/0         0      564K/10010(30) us  2742
[  3] 2.00-3.00 sec  29.2 MBytes   245 Mbits/sec  306053/0         0      707K/10049(45) us  3046
[SUM] 2.00-3.00 sec  83.6 MBytes   702 Mbits/sec  876906/0         0
[  2] 0.00-3.02 sec  60.6 MBytes   168 Mbits/sec  635592/0        43      514K/9932(61) us  2120
[  1] 0.00-3.02 sec  6.05 KBytes  16.4 Kbits/sec    31=5.757/0.148/9.842/3.844 ms    0   14K/8328 us    174 rps
[  1] 0.00-3.02 sec  OWD Delays (ms) Cnt=31 To=5.683/0.084/9.772/3.872 From=0.052/0.042/0.073/0.010 Asymmetry=5.631/0.012/9.703/3.878    174 rps
[  1] 0.00-3.02 sec BB8(f)-PDF: bin(w=100us):cnt(31)=2:4,3:5,22:1,71:1,72:1,74:2,75:2,76:1,77:1,78:2,83:1,87:1,88:1,89:2,90:1,94:1,95:3,99:1 (5.00/95.00/99.7%=2/95/99,Outliers=0,obl/obu=0/0)
[  3] 0.00-3.02 sec  88.3 MBytes   246 Mbits/sec  926343/0        48      708K/10114(54) us  3035
[  4] 0.00-3.02 sec  80.2 MBytes   223 Mbits/sec  840532/0        24      564K/9941(123) us  2802
[  5] 0.00-3.02 sec  85.1 MBytes   237 Mbits/sec  892360/0        42      608K/9812(71) us  3013
[SUM] 0.00-3.01 sec   254 MBytes   707 Mbits/sec  2659235/0       114
[ CT] final connect times (min/avg/max/stdev) = 0.369/0.385/0.402/14.398 ms (tot/err) = 5/0
```


BROADCOM

Client side outputs: TCP with tcp_info

```
[root@fedora iperf-2.1.9-rc2]# iperf -c 192.168.1.15 -i 1 --trip-times  -t 3
------------------------------------------------------------
Client connecting to 192.168.1.15, TCP port 5001 with pid 89674 (1 flows)
Write buffer size: 131072 Byte
TOS set to 0x0 (Nagle on)
TCP window size: 16.0 KByte (default)
Event based writes (pending queue watermark at 16384 bytes)
------------------------------------------------------------
[  1] local 192.168.1.234%eth1 port 34584 connected with 192.168.1.15 port 5001 (prefetch=16384) (trip-times) (sock=3)
(icwnd/mss/irtt=14/1448/8450) (ct=8.52 ms) on 2023-03-08 11:58:34.476 (PST)
[ ID] Interval       Transfer    Bandwidth       Write/Err  Rtry     Cwnd/RTT(var)        NetPwr
[  1] 0.00-1.00 sec  98.9 MBytes   829 Mbits/sec  791/0        0     5609K/18192(233) us  5699
[  1] 1.00-2.00 sec   102 MBytes   860 Mbits/sec  820/0        0     5609K/16251(294) us  6614
[  1] 2.00-3.00 sec   101 MBytes   846 Mbits/sec  807/0        0     5609K/17576(5175) us  6018
[  1] 0.00-3.02 sec   302 MBytes   840 Mbits/sec  2419/0         0     5609K/14192(1405) us  7398
```


Server side outputs: TCP write to read latencies

```
[root@ctrl1fc35 iperf-2.1.9-rc2]# iperf -s -i 1 -e --histograms
------------------------------------------------------------
Server listening on TCP port 5001 with pid 4415
Read buffer size:  128 KByte (Dist bin width=16.0 KByte)
Enabled receive histograms bin-width=0.100 ms, bins=10000 (clients should use --trip-times)
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.15%enp2s0 port 5001 connected with 192.168.1.234 port 34584 (trip-times) (sock=4) (peer 2.1.9-rc2) (icwnd/mss/irtt=14/1448/8286) on 2023-03-08 11:58:34.485 (PST)
[ ID] Interval       Transfer    Bandwidth    Burst Latency avg/min/max/stdev (cnt/size) inP NetPwr  Reads=Dist
[  1] 0.00-1.00 sec  98.8 MBytes   829 Mbits/sec  9.049/2.007/20.555/4.270 ms (790/131152)  932 KByte 11449  8372=6534:1069:368:232:71:28:23:47
[  1] 0.00-1.00 sec F8-PDF:
bin(w=100us):cnt(790)=21:1,22:3,23:2,24:6,25:3,26:6,27:4,28:3,29:9,30:7,31:4,32:4,33:5,34:3,35:6,36:7,37:4,38:10,39:7,40:7,41:4,42:10,43:11,44:6,45:5,46:5,47:9,48:10,49:6,50:4,51:4,52:12,53:5,54:4,55:13,56:4,57:10,58:5,59:6,60:9,61:6,62:3,63:5,64:6,65:5,66:3,67:7,68:4,69:5,70:4,71:6,72:5,73:8,74:5,75:5,76:7,77:2,78:3,79:6,80:8,81:5,82:4,83:5,84:6,85:5,86:15,87:11,88:5,89:5,90:4,91:9,92:8,93:8,94:6,95:5,96:8,97:9,98:7,99:2,100:5,101:9,102:8,103:3,104:2,105:9,106:7,107:3,108:6,109:6,110:4,111:7,112:6,113:4,114:4,115:2,116:5,117:5,118:6,119:4,120:3,121:3,122:7,123:9,124:2,125:5,126:1,127:4,128:1,129:2,130:7,131:3,132:2,133:3,134:6,135:9,136:6,137:4,138:5,139:4,140:4,141:5,142:7,143:6,144:1,145:5,146:1,147:7,148:2,149:1,150:3,151:3,152:2,153:5,154:4,155:3,157:1,158:2,159:1,160:2,161:4,162:3,163:4,164:1,165:3,166:5,167:3,168:1,169:3,170:2,171:4,172:2,173:3,174:2,175:2,176:3,177:3,180:1,182:1,183:4,185:1,187:1,192:1,196:1,198:1,202:1,206:1 (5.00/95.00/99.7%=30/167/198,Outliers=0,obl/obu=0/0) (20.555 ms/1678305514.678640)
[  1] 1.00-2.00 sec   102 MBytes   853 Mbits/sec  8.992/1.922/22.936/4.549 ms (813/131096)  928 KByte 11853  8091=6245:986:418:245:71:37:21:68
[  1] 1.00-2.00 sec F8-PDF:
bin(w=100us):cnt(813)=20:2,22:4,23:9,24:6,25:3,26:4,27:7,28:7,29:6,30:11,31:4,32:9,33:7,34:9,35:6,36:4,37:4,38:7,39:11,40:6,41:2,42:7,43:7,44:7,45:16,46:9,47:2,48:8,49:9,50:8,51:6,52:7,53:7,54:5,55:7,56:7,57:6,58:2,59:8,60:4,61:4,62:8,63:10,64:4,65:4,66:4,67:4,68:7,69:5,70:6,71:2,72:11,73:8,74:4,75:4,76:11,77:8,78:7,79:4,80:12,81:6,82:5,83:3,84:4,85:5,86:8,87:7,88:3,89:4,90:8,91:8,92:3,93:4,94:4,95:5,96:7,97:7,98:8,99:6,100:5,101:7,102:5,103:3,104:9,105:4,106:4,107:4,108:7,109:3,110:9,111:4,112:6,113:1,114:3,115:3,116:4,117:5,118:3,119:3,120:2,121:6,122:2,123:9,124:5,125:7,126:1,127:6,128:3,129:4,130:4,131:5,132:5,133:8,134:1,135:1,136:4,137:5,138:4,139:3,140:6,141:7,142:4,143:7,144:4,145:5,146:4,147:4,148:2,149:4,150:6,151:1,152:2,153:3,154:5,155:3,157:3,158:2,160:2,161:2,162:4,163:3,164:3,165:2,166:2,167:3,169:4,170:3,171:5,172:2,173:2,174:2,175:1,176:2,178:2,179:3,180:2,181:3,182:1,183:2,185:1,187:1,189:1,190:3,191:2,197:2,199:1,203:1,206:1,207:1,218:1,221:1,230:1 (5.00/95.00/99.7%=28/171/218,Outliers=0,obl/obu=0/0) (22.936 ms/1678305515.751730)
[  1] 2.00-3.00 sec   100 MBytes   841 Mbits/sec  8.439/1.809/20.049/3.966 ms (802/131040)  865 KByte 12453  8710=6792:1117:445:206:67:27:21:35
[  1] 2.00-3.00 sec F8-PDF:
bin(w=100us):cnt(802)=19:1,20:2,21:2,22:3,23:4,24:9,25:2,26:2,27:6,28:5,29:3,30:5,31:7,32:4,33:8,34:14,35:4,36:10,37:6,38:4,39:8,40:7,41:10,42:9,43:5,44:9,45:4,46:11,47:9,48:11,49:5,50:5,51:5,52:6,53:14,54:8,55:4,56:5,57:6,58:8,59:6,60:5,61:7,62:8,63:6,64:3,65:7,66:6,67:4,68:10,69:4,70:11,71:9,72:4,73:6,74:7,75:7,76:8,77:3,78:6,79:3,80:9,81:15,82:10,83:2,84:7,85:8,86:10,87:9,88:4,89:2,90:12,91:9,92:7,93:5,94:5,95:7,96:5,97:9,98:8,99:8,100:6,101:5,102:6,103:8,104:8,105:3,106:4,107:4,108:5,109:5,110:6,111:7,112:2,113:5,114:3,115:7,116:6,117:5,118:4,119:5,120:3,121:3,122:2,123:4,124:5,125:7,126:4,127:1,128:3,129:6,130:1,131:5,132:4,133:2,134:4,135:2,136:3,137:6,138:4,139:4,140:8,141:4,143:3,144:7,145:6,146:4,147:4,148:4,149:2,150:2,151:2,152:1,153:2,154:2,157:3,158:2,159:3,160:4,161:3,162:1,163:6,165:1,166:3,168:2,171:3,172:1,173:1,176:1,177:1,180:1,183:1,184:2,187:1,190:1,193:1,201:1 (5.00/95.00/99.7%=30/157/190,Outliers=0,obl/obu=0/0) (20.049 ms/1678305517.191966)
[  1] 0.00-3.02 sec   302 MBytes   841 Mbits/sec  8.835/1.809/22.936/4.272 ms (2419/131072)  838 KByte 11898  25284=19649:3186:1244:684:211:92:66:152
[  1] 0.00-3.02 sec F8(f)-PDF:
bin(w=100us):cnt(2419)=19:1,20:4,21:3,22:10,23:15,24:21,25:8,26:12,27:17,28:15,29:18,30:23,31:15,32:17,33:20,34:26,35:16,36:21,37:14,38:21,39:26,40:20,41:16,42:26,43:23,44:23,45:25,46:25,47:20,48:29,49:20,50:17,51:15,52:25,53:26,54:17,55:24,56:16,57:22,58:15,59:20,60:18,61:17,62:19,63:22,64:13,65:16,66:13,67:15,68:21,69:14,70:21,71:17,72:20,73:22,74:17,75:16,76:26,77:13,78:16,79:14,80:29,81:26,82:19,83:10,84:17,85:18,86:33,87:27,88:12,89:11,90:24,91:26,92:18,93:17,94:16,95:18,96:20,97:25,98:23,99:17,100:16,101:22,102:19,103:14,104:19,105:17,106:15,107:11,108:18,109:14,110:19,111:18,112:14,113:10,114:10,115:12,116:15,117:15,118:13,119:12,120:8,121:12,122:11,123:22,124:12,125:19,126:7,127:12,128:7,129:12,130:12,131:13,132:11,133:13,134:12,135:12,136:13,137:16,138:13,139:11,140:18,141:16,142:11,143:16,144:12,145:16,146:9,147:15,148:8,149:7,150:11,151:6,152:5,153:10,154:11,155:6,157:7,158:6,159:4,160:8,161:10,162:8,163:13,164:4,165:6,166:10,167:6,168:3,169:7,170:5,171:12,172:5,173:6,174:4,175:3,176:6,177:4,178:2,179:3,180:4,181:3,182:2,183:7,184:2,185:2,187:3,189:1,190:4,191:2,192:1,193:1,196:1,197:2,198:1,199:1,201:1,202:1,203:1,206:2,207:1,218:1,221:1,230:1 (5.00/95.00/99.7%=29/166/202,Outliers=0,obl/obu=0/0) (22.936 ms/1678305515.751730)
```


BROADCOM

Server side outputs: TCP bursts (--burst-size 10M --burst-period 1)

```
[root@ctrl1fc35 iperf-2.1.9-rc2]# iperf -s -i 1 -e --histograms
------------------------------------------------------------
Server listening on TCP port 5001 with pid 11729
Read buffer size:  128 KByte (Dist bin width=16.0 KByte)
Enabled receive histograms bin-width=0.100 ms, bins=10000 (clients should use --trip-times)
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.15%enp2s0 port 5001 connected with 192.168.1.234 port 34636 (burst-period=1.00s) (trip-times) (sock=4) (peer
2.1.9-rc2) (icwnd/mss/irtt=14/1448/5009) on 2023-03-08 21:10:04.609 (PST)
[ ID] Burst (start-end)  Transfer     Bandwidth       XferTime  (DC%)     Reads=Dist         NetPwr
[  1] 0.00-0.12 sec  10.0 MBytes   674 Mbits/sec  124.533 ms (12%)    195=33:37:18:58:14:3:10:22  676
[  1] 1.00-1.11 sec  10.0 MBytes   751 Mbits/sec  111.671 ms (11%)    206=43:37:24:53:16:1:6:26  841
[  1] 2.00-2.11 sec  10.0 MBytes   752 Mbits/sec  111.530 ms (11%)    201=34:40:23:54:16:3:9:22  843
[  1] 0.00-3.00 sec  30.0 MBytes  83.8 Mbits/sec  115.911/111.530/124.533/7.467 ms  602=110:114:65:165:46:7:25:70
[  1] 0.00-3.00 sec F8(f)-PDF: bin(w=100us):cnt(3)=1116:1,1117:1,1246:1 (5.00/95.00/99.7%=1116/1246/1246,Outliers=0,obl/obu=0/0)
(124.533 ms/1678338604.609173)
```


Client side outputs: TCP Congestion Control Algorithm (CCA)

```
[root@fedora iperf-2.1.9-rc2]# iperf -c 192.168.1.15 -i 1 --trip-times -Z cubic -t 3
------------------------------------------------------------
Client connecting to 192.168.1.15, TCP port 5001 with pid 89744 (1 flows)
Write buffer size: 131072 Byte
TCP congestion control set to cubic
TOS set to 0x0 (Nagle on)
TCP window size: 16.0 KByte (default)
Event based writes (pending queue watermark at 16384 bytes)
------------------------------------------------------------
[  1] local 192.168.1.234%eth1 port 34596 connected with 192.168.1.15 port 5001 (prefetch=16384) (trip-times) (sock=3) (icwnd/mss/irtt=14/1448/8878) (ct=8.95 ms) on 2023-03-08 12:29:05.428
(PST)
[ ID] Interval       Transfer     Bandwidth       Write/Err  Rtry     Cwnd/RTT(var)        NetPwr
[  1] 0.00-1.00 sec  93.0 MBytes   780 Mbits/sec  744/0          0     2874K/23863(5038) us  4087
[  1] 1.00-2.00 sec   100 MBytes   843 Mbits/sec  804/0          0     2874K/17060(387) us  6177
[  1] 2.00-3.00 sec   102 MBytes   854 Mbits/sec  814/0          0     2874K/16052(776) us  6647
[  1] 0.00-3.03 sec   295 MBytes   818 Mbits/sec  2363/0         0     2874K/18610(1635) us  5493

[root@fedora iperf-2.1.9-rc2]# iperf -c 192.168.1.15 -i 1 --trip-times -Z bbr -t 3
------------------------------------------------------------
Client connecting to 192.168.1.15, TCP port 5001 with pid 89747 (1 flows)
Write buffer size: 131072 Byte
TCP congestion control set to bbr
TOS set to 0x0 (Nagle on)
TCP window size: 16.0 KByte (default)
Event based writes (pending queue watermark at 16384 bytes)
------------------------------------------------------------
[  1] local 192.168.1.234%eth1 port 34598 connected with 192.168.1.15 port 5001 (prefetch=16384) (trip-times) (sock=3) (icwnd/mss/irtt=14/1448/4392) (ct=4.46 ms) on 2023-03-08 12:29:13.419
(PST)
[ ID] Interval       Transfer     Bandwidth       Write/Err  Rtry     Cwnd/RTT(var)        NetPwr
[  1] 0.00-1.00 sec  36.5 MBytes   306 Mbits/sec  292/0          0      562K/5276(793) us  7254
[  1] 1.00-2.00 sec  37.2 MBytes   312 Mbits/sec  298/0          0      559K/5709(1050) us  6842
[  1] 2.00-3.00 sec  37.1 MBytes   311 Mbits/sec  297/0          0      619K/3812(488) us  10212
[  1] 0.00-3.02 sec   111 MBytes   308 Mbits/sec  888/0          0     619K/3935(515) us  9797
```


BROADCOM

Client side: –txstart-times to coordinate senders

```
[rjmcmahon@ryzen3950 iperf2-code]$ iperf -c 192.168.1.72 --txstart-time $(expr $(date +%s) + 1).$(date +%N) -i1 -t3 -e
------------------------------------------------------------
Client connecting to 192.168.1.72, TCP port 5001 with pid 89611 (1 flows)
Write buffer size: 131072 Byte
TOS set to 0x0 (Nagle on)
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.69%enp4s0 port 53122 connected with 192.168.1.72 port 5001 (epoch-start) (sock=3)
(icwnd/mss/irtt=14/1448/258) (ct=0.38 ms) on 2023-03-14 17:50:36.177 (PDT)
[  1] Client traffic to start in 1 seconds at 2023-03-14 17:50:37.177 current time is 2023-03-14 17:50:36.177 (PDT)
[ ID] Interval       Transfer     Bandwidth       Write/Err  Rtry     Cwnd/RTT(var)        NetPwr
[  1] 0.00-1.00 sec  1.10 GBytes  9.43 Gbits/sec  8992/0        0     1552K/1089(22) us  1082277
[  1] 1.00-2.00 sec  1.10 GBytes  9.42 Gbits/sec  8980/0        0     1552K/1089(13) us  1080832
[  1] 2.00-3.00 sec  1.10 GBytes  9.42 Gbits/sec  8979/0        0     1552K/1088(29) us  1081705
[  1] 0.00-3.01 sec  3.29 GBytes  9.40 Gbits/sec  26952/0       0     1552K/1081(26) us  1087278
```


Fast sampling (./configure --enable-fastsampling)

```
[rjmcmahon@ryzen3950 iperf2-code]$ src/iperf -c 192.168.1.72 -i 0.001 -e -t 1
------------------------------------------------------------
Client connecting to 192.168.1.72, TCP port 5001 with pid 2826116 (1 flows)
Write buffer size: 131072 Byte
TOS set to 0x0 (Nagle on)
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.69%enp4s0 port 42462 connected with 192.168.1.72 port 5001 (sock=3) (icwnd/mss/irtt=14/1448/695) (ct=0.77 ms) on 2023-03-09 21:03:59.453 (PST)
[ ID] Interval          Transfer    Bandwidth       Write/Err  Rtry     Cwnd/RTT(var)        NetPwr
[  1] 0.0000-0.0010 sec   256 KBytes  2.10 Gbits/sec  2/0        0      63K/376(198) us  696654
[  1] 0.0010-0.0020 sec   196 KBytes  1.60 Gbits/sec  2/0        0      69K/480(116) us  417333
[  1] 0.0020-0.0030 sec   174 KBytes  1.42 Gbits/sec  2/0        0      79K/550(117) us  323502
[  1] 0.0030-0.0040 sec  85.5 KBytes   700 Mbits/sec  1/0        0      79K/550(117) us  159070
[  1] 0.0040-0.0050 sec   128 KBytes  1.05 Gbits/sec  1/0        0      87K/588(93) us  222689
[  1] 0.0050-0.0060 sec   191 KBytes  1.56 Gbits/sec  2/0        0      94K/634(100) us  308020
[  1] 0.0060-0.0070 sec  62.9 KBytes   515 Mbits/sec  1/0        0      94K/634(100) us  101488
[  1] 0.0070-0.0080 sec   128 KBytes  1.05 Gbits/sec  1/0        0     104K/754(139) us  173662
[  1] 0.0080-0.0090 sec   255 KBytes  2.08 Gbits/sec  2/0        0     115K/757(123) us  343963
[  1] 0.0090-0.0100 sec   127 KBytes  1.04 Gbits/sec  1/0        0     115K/757(123) us  170989
[  1] 0.0100-0.0110 sec   128 KBytes  1.05 Gbits/sec  1/0        0     127K/877(83) us  149306
[  1] 0.0110-0.0120 sec   127 KBytes  1.04 Gbits/sec  1/0        0     127K/877(83) us  147592
[  1] 0.0120-0.0130 sec   256 KBytes  2.10 Gbits/sec  2/0        0     139K/972(109) us  269426
[  1] 0.0130-0.0140 sec  62.2 KBytes   509 Mbits/sec  1/0        0     139K/972(109) us  65424
[  1] 0.0140-0.0150 sec   256 KBytes  2.10 Gbits/sec  2/0        0     151K/1103(120) us  237427
[  1] 0.0150-0.0160 sec  62.2 KBytes   509 Mbits/sec  1/0        0     151K/1103(120) us  57654
[  1] 0.0160-0.0170 sec   128 KBytes  1.05 Gbits/sec  1/0        0     162K/1198(196) us  109300
[  1] 0.0170-0.0180 sec   127 KBytes  1.04 Gbits/sec  1/0        0     162K/1198(196) us  108046
[  1] 0.0180-0.0190 sec   256 KBytes  2.10 Gbits/sec  2/0        0     173K/1286(154) us  203641
[  1] 0.0190-0.0200 sec  62.2 KBytes   509 Mbits/sec  1/0        0     173K/1286(154) us  49450
[  1] 0.0200-0.0210 sec   256 KBytes  2.10 Gbits/sec  2/0        0     190K/1334(125) us  196313
```


Congestion Techniques & Options

- Single TCP stream
- Parallel TCP streams (same host)
- Multiple simultaneous streams (multiple STAs, same AP)
- Multiple simultaneous streams with microsecond resolution start timestamps
- OBSS traffic via 5 branch tree & distance matrix (not shown)

- -P or –parallel greater than 1
- -P with –incr-dstip
- -p port range, -p 5001-5016 both client & receiver (multicast too)
- -P with –incr-srcport or –incr-dstport
- Python flows instances

Bufferbloat

- Value provided per iperf 2's inP
- Displacement units are bytes for TCP and packets for UDP
- This is a sample based upon synthetic traffic, your mileage will vary depending upon actual traffic loads

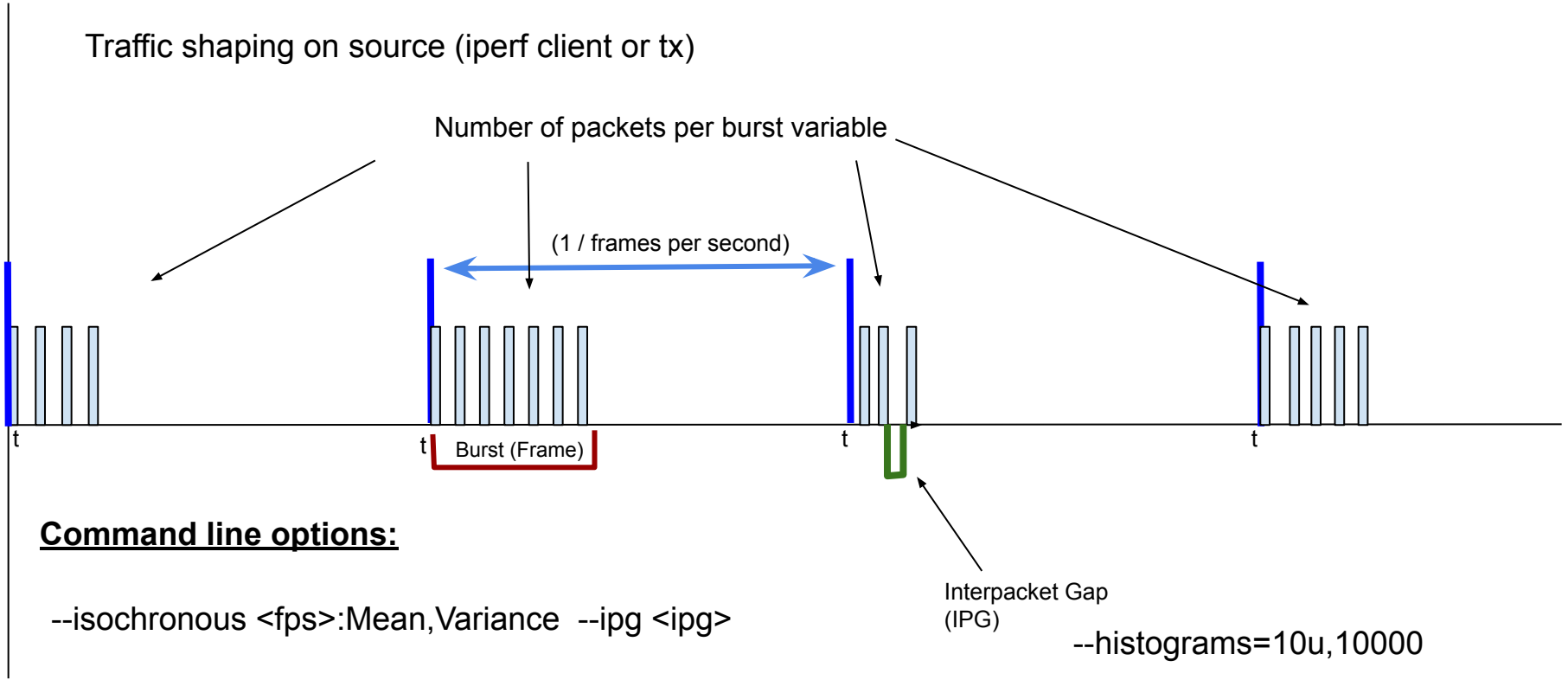
BROADCOM

Responsiveness

- TCP app level bounceback, write to read to write to read in bursts
- Part of the open community responsiveness intiatives

BROADCOM

Isochronous w/cbr or vbr (both TCP & UDP)



Command line options:

--isochronous <fps>:Mean,Variance --ipg <ipg>

--histograms=10u,10000

BROADCOM

How to scale

- Use of -P, incr-dstip, etc
- Use of passworldless openssh and control master
- Use python 3.10 with asyncio
- Number of cores on a DUT should match number of active traffic streams
- High speed python controller - CPU cores will help with lots of ssh settings
- Increase file descriptor minimums

Code repository on sourceforge

- Iperf 2 repository
- Man page
- Iperf 2 vs iperf3
- git clone https://git.code.sf.net/p/iperf2/code iperf2-code

Note: IPerf 2 (this program) is different from the iperf3 found at https://github.com/esnet/iperf Each can be used to measure network performance, however, they DO NOT interoperate. They are completely independent implementations with different strengths, different options, and different capabilities. Both are under active development (as of mid-2021)

Back up slides

BROADCOM

Iperf3 statement on iperf2

What is the history of iperf3, and what is the difference between iperf2 and iperf3?
iperf2 was orphaned in the late 2000s at version 2.0.5, despite some known bugs and issues. After spending some time trying to fix iperf2's problems, ESnet decided by 2010 that a new, simpler tool was needed, and began development of iperf3. The goal was make the tool as simple as possible, so others could contribute to the code base. For this reason, it was decided to make the tool single threaded, and not worry about backwards compatibility with iperf2. Many of the feature requests for iperf3 came from the perfSONAR project (http://www.perfsonar.net).

Then in 2014, Bob (Robert) McMahon from Broadcom restarted development of iperf2 (See https://sourceforge.net/projects/iperf2/). He fixed many of the problems with iperf2, and added a number of new features similar to iperf3. iperf2.0.8, released in 2015, made iperf2 a useful tool. iperf2's current development is focused is on using UDP for latency testing, as well as broad platform support.

As of this writing (2017), both iperf2 and iperf3 are being actively (although independently) developed. We recommend being familiar with both tools, and use whichever tool's features best match your needs.

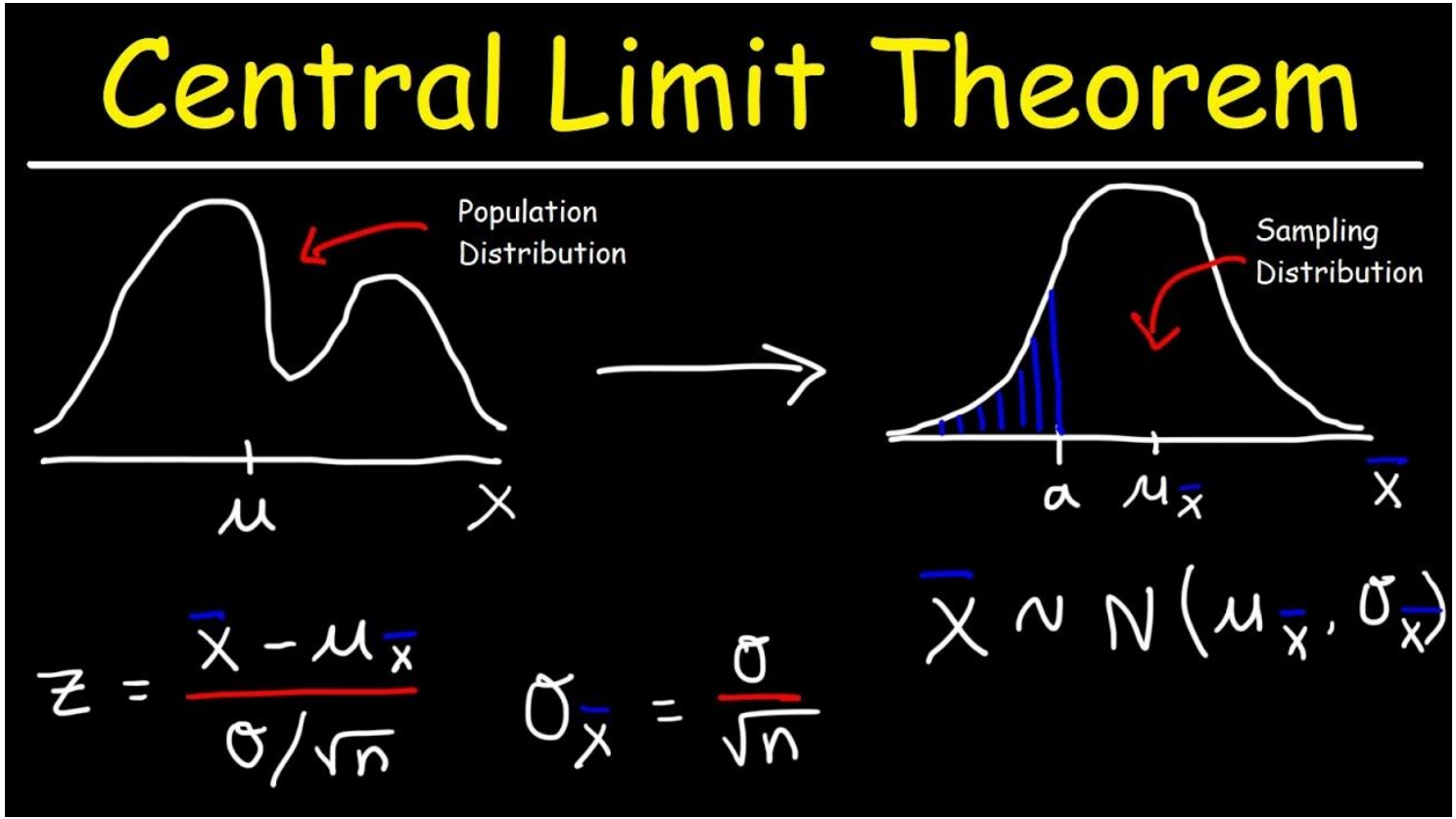
iperf3 parallel stream performance is much less than iperf2. Why?
iperf3 is single threaded, and iperf2 is multi-threaded. We recommend using iperf2 for parallel streams.

BROADCOM

Other Benefits

- All science & engineering requires test & measurement to validate
- No payments for IXIA licenses
- Wide adoption
- Been used for more than a decade
- Accuracy of measurements thoroughly vetted
  - through hundreds of Broadcom test rigs
  - statistical process controls (SPC)
- Python flows code with scikit-learn support
- Works as a bench test or integrated into anyone's automation framework

Central limit theorem (CLT)



BROADCOM

What does clock sync even mean?

Let me start by saying that I have not been intimately involved with the IEEE 1588 effort (PTP), however I was involved in the 802.11 efforts along a similar vein, just adding the wireless first hop component and it's effects on PTP.

What was apparent from the outset was that there was a lack of understanding what the terms "to synchronize" or "to be synchronized" actually mean. It's not trivial … because we live in a (approximately, that's another story!) 4-D space-time continuum where the Lorentz metric plays a critical role. ***Therein, simultaneity (aka "things happening at the same time") means the "distance" between two such events is zero*** *and that distance is given by sqrt(x^2 + y^2 + z^2 – (ct)^2) and the "thing happening" can be the tick of a clock somewhere. Now since everything is relative (time with respect to what? / location with respect to where?) it's pretty easy to see that "if you don't know where you are, you can't know what time it is!" (English sailors of the 18th century knew this well!) Add to this the fact that if everything were stationary, nothing would happen (as Einstein said "Nothing happens until something moves!"), special relativity also pays a role. Clocks on GPS satellites run approx. 7usecs/day slower than those on earth due to their "speed" (8700 mph roughly)! Then add the consequence that without mass we wouldn't exist (in these forms at leastJ), and gravitational effects (aka General Relativity) come into play. Those turn out to make clocks on GPS satellites run 45usec/day faster than those on earth! The net effect is that GPS clocks run about 38usec/day faster than clocks on earth. So what does it mean to "synchronize to GPS"?* ***Point is: it's a non-trivial question with a very complicated answer. The reason it is important to get all this right is that the "what that ties time and space together" is the speed of light and that turns out to be a "foot-per-nanosecond" in a vacuum (roughly 300m/usec). This means if I am uncertain about my location to say 300 meters, then I also am not sure what time it is to a usec AND vice-versa!***

All that said, the simplest explanation of synchronization is probably: Two clocks are synchronized if, when they are brought (slowly) into physical proximity ("sat next to each other") in the same (quasi-)inertial frame and the same gravitational potential (not so obvious BTW … see the FYI below!), an observer of both would say "they are keeping time identically". Since this experiment is rarely possible, one can never be "sure" that his clock is synchronized to any other clock elsewhere. And what does it mean to say they "were synchronized" when brought together, but now they are not because they are now in different gravitational potentials! (FYI, there are land mine detectors being developed on this very principle! I know someone who actually worked on such a project!)

This all gets even more complicated when dealing with large networks of networks in which the "speed of information transmission" can vary depending on the medium (cf. coaxial cables versus fiber versus microwave links!) In fact, the atmosphere is one of those media and variations therein result in the need for "GPS corrections" (cf. RTCM GPS correction messages, RTK, etc.) in order to get to sub-nsec/cm accuracy. Point is if you have a set of nodes distributed across the country all with GPS and all "synchronized to GPS time", and a second identical set of nodes (with no GPS) instead connected with a network of cables and fiber links, all of different lengths and composition using different carrier frequencies (dielectric constants vary with frequency!) "synchronized" to some clock somewhere using NTP or PTP), the synchronization of the two sets will be different unless a common reference clock is used AND all the above effects are taken into account, and good luck with that! J

In conclusion, if anyone tells you that clock synchronization in communication networks is simple ("Just use GPS!"), you should feel free to chuckle (under your breath if necessary)

Cheers,
RR

BROADCOM

TCP optimizations

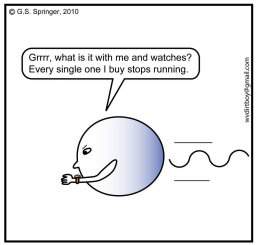
This article will expose some of the interactions between the web server, Operating System and network and how to tune a server to optimize performance for end users.

BROADCOM

Reminder: Packets & Photons aren't things and aren't conserved

Feynman's dad used to tease young Richard by telling him that if he kept saying "no" that he would run out of no's and would have to answer "yes" to everything.

In physics jargon, we say that photons are not "conserved". Energy is conserved, and if you want to create a photon you need energy. But the photon is not hiding inside something ready to be produced; it doesn't exist until it is made.

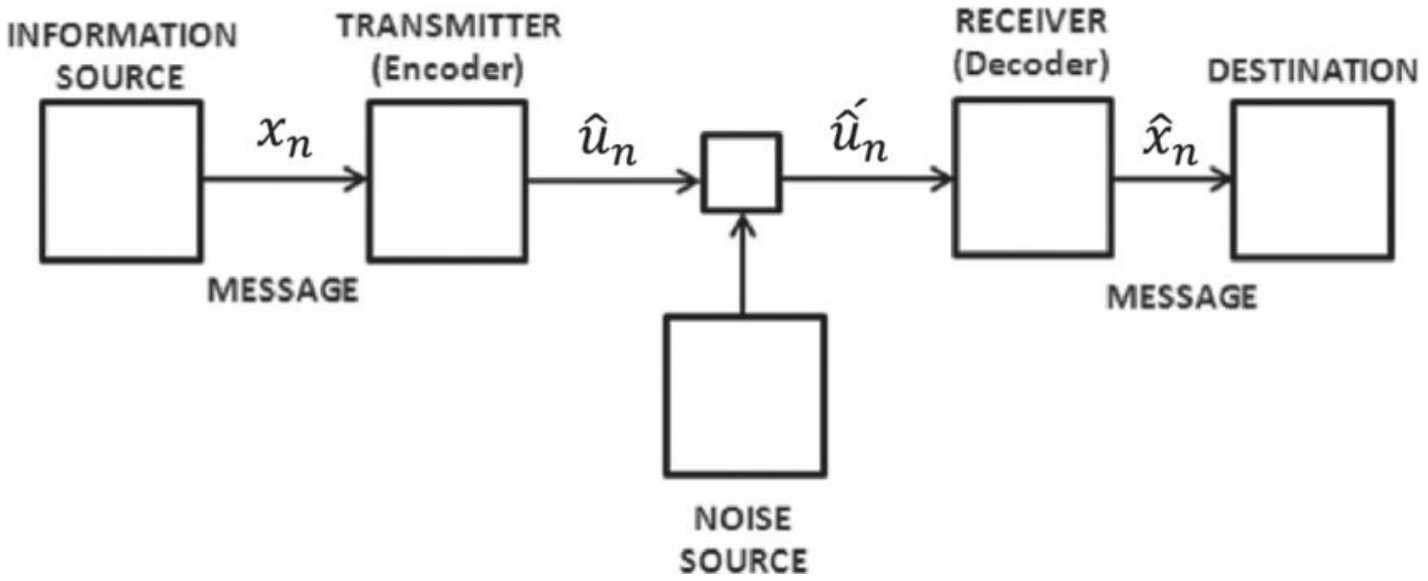


Packet's aren't things either. They're not dropped like an apple. They're not lost. The question is, like a word or message, do they affect or not and within the relevant space & time frames.

BROADCOM

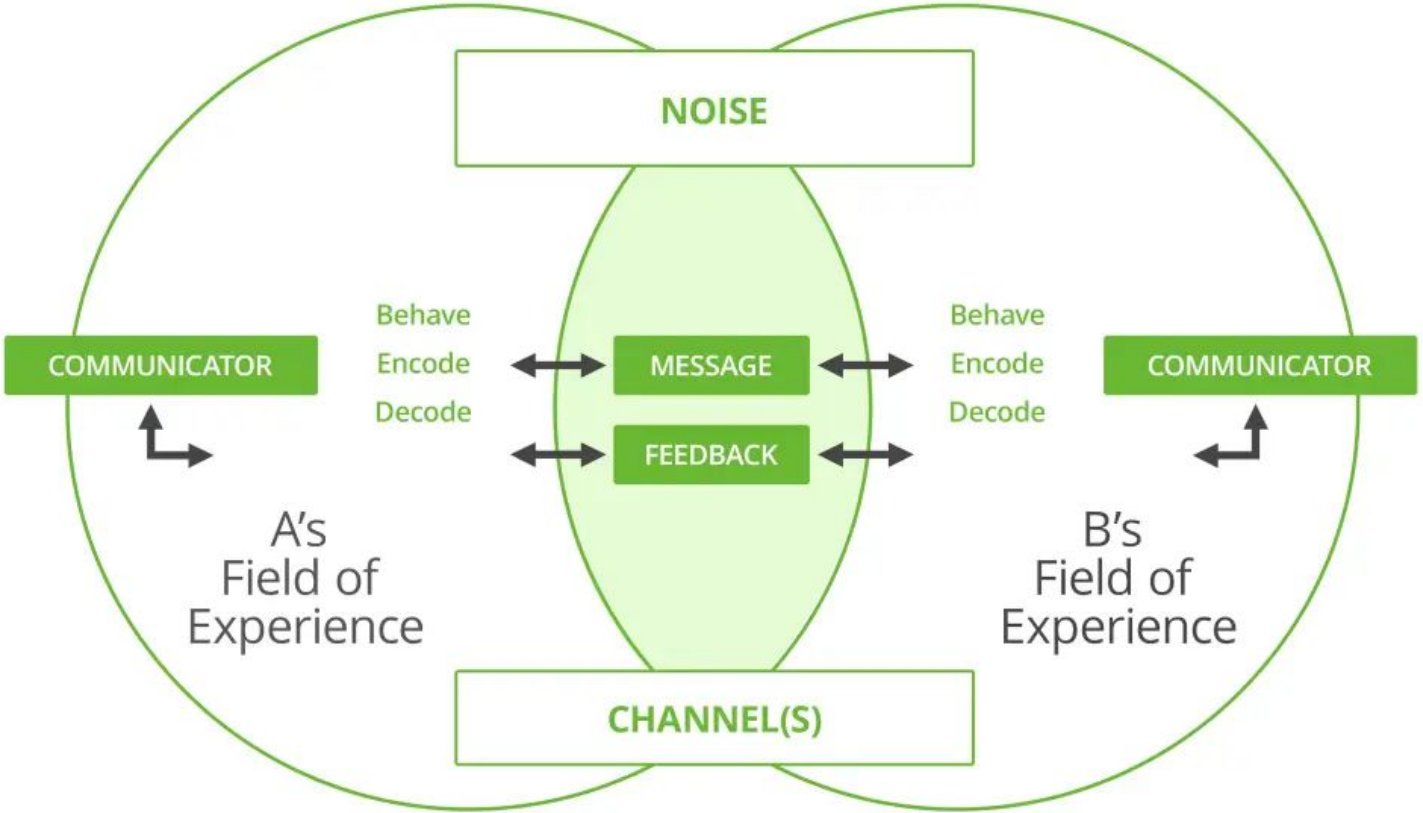
Shannon (no mention of layers, packets nor sockets)

"The fundamental problem of communication is that of reproducing at one point, either exactly or approximately, a message selected at another point."



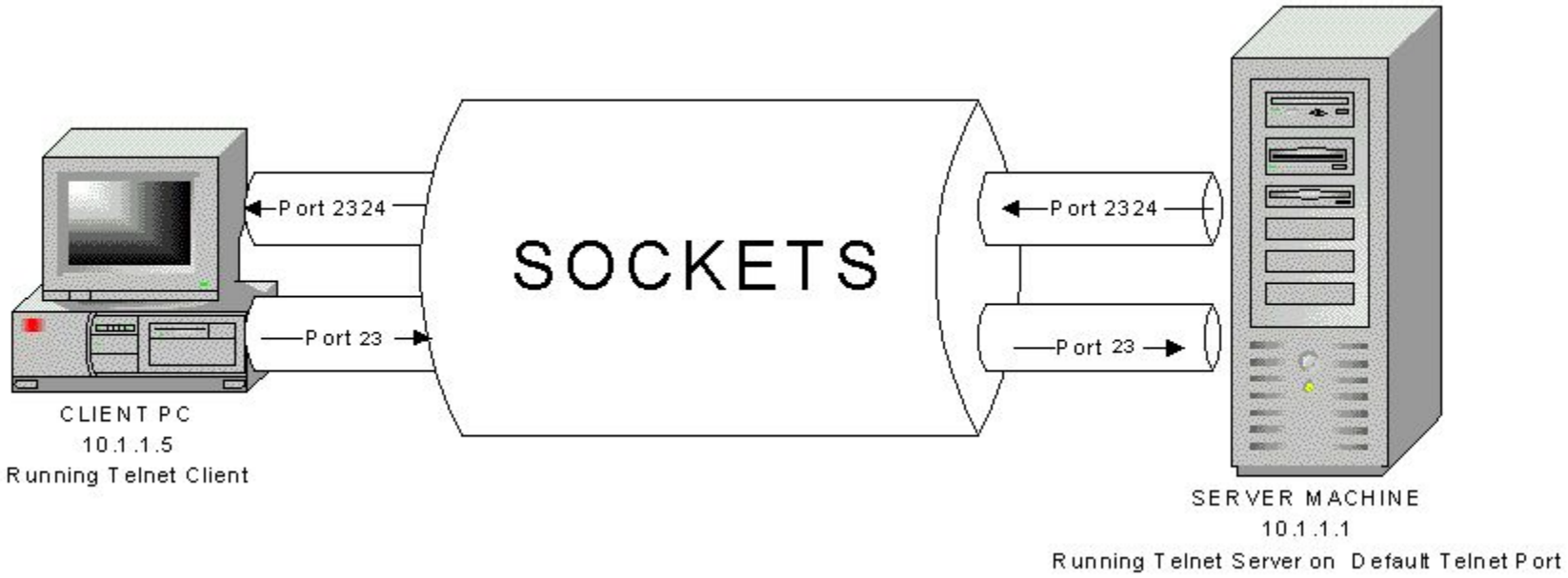
BROADCOM

Model with Noise & Two Way



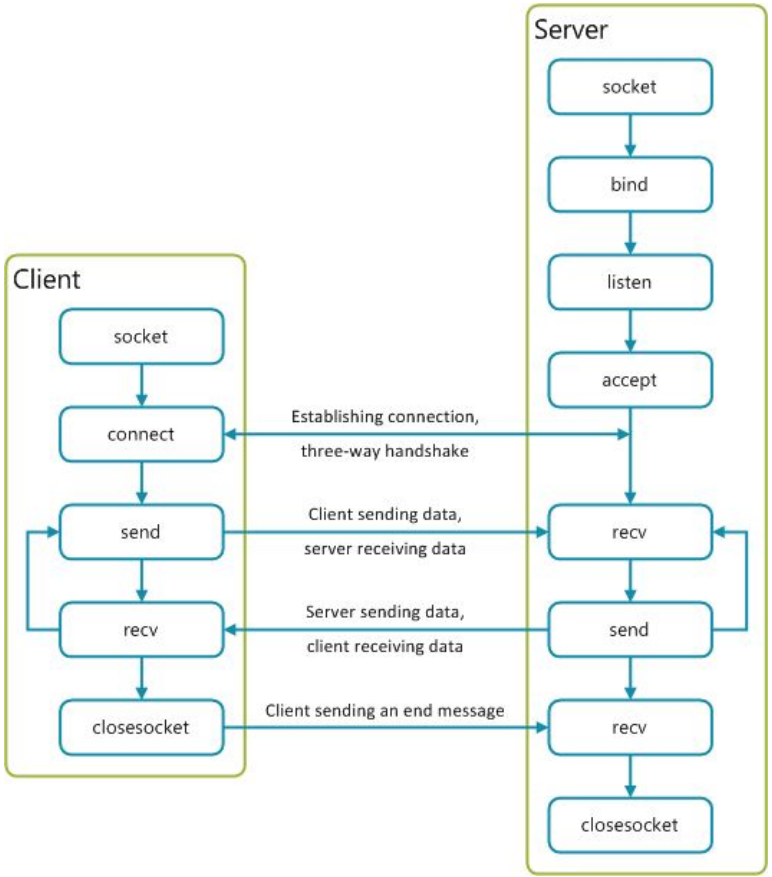
BROADCOM

Sockets



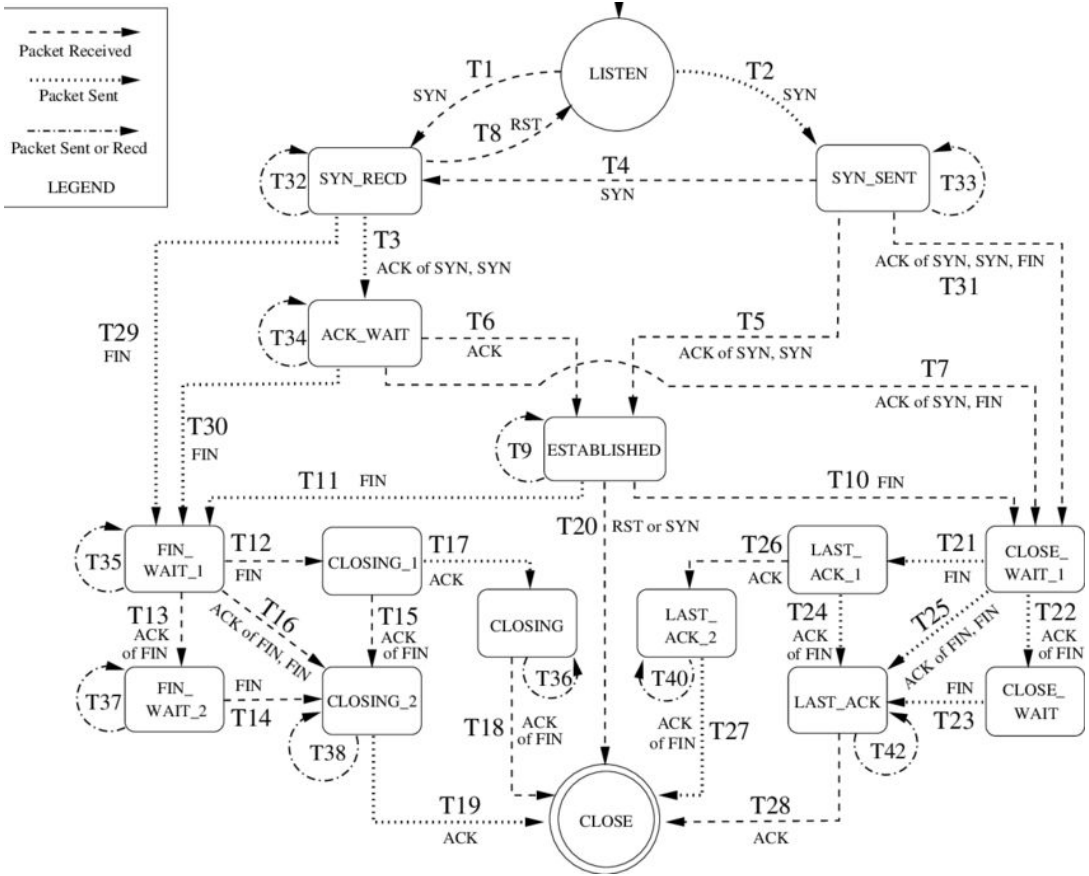
BROADCOM

Sockets - let the state machines begin



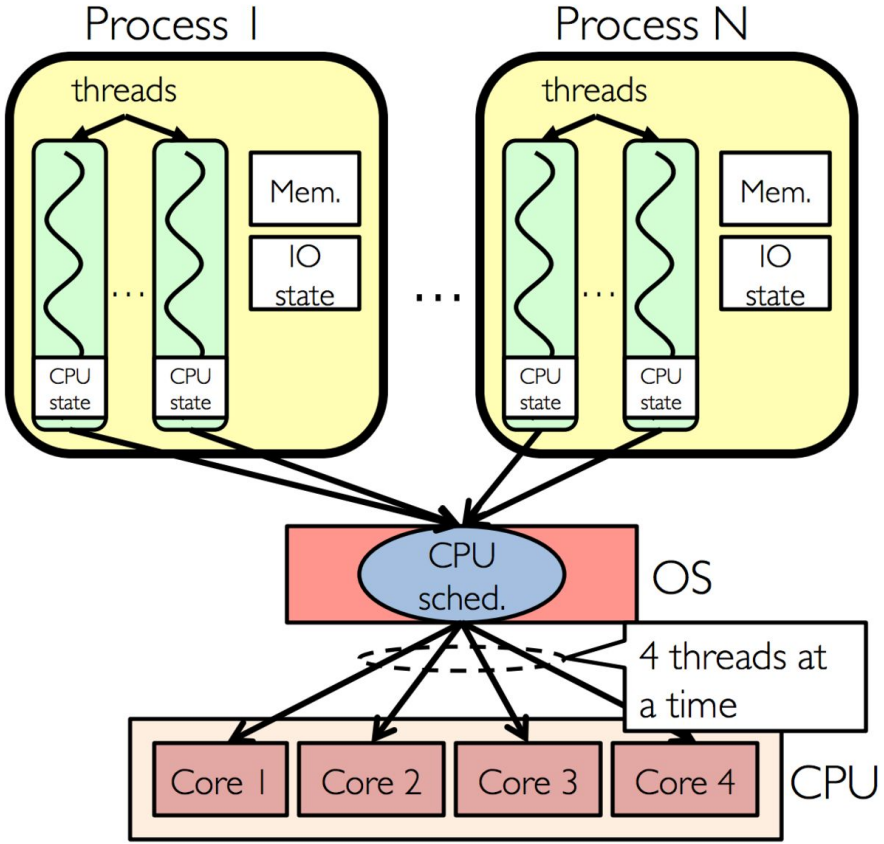
BROADCOM

Let’s kick things off like TCP does



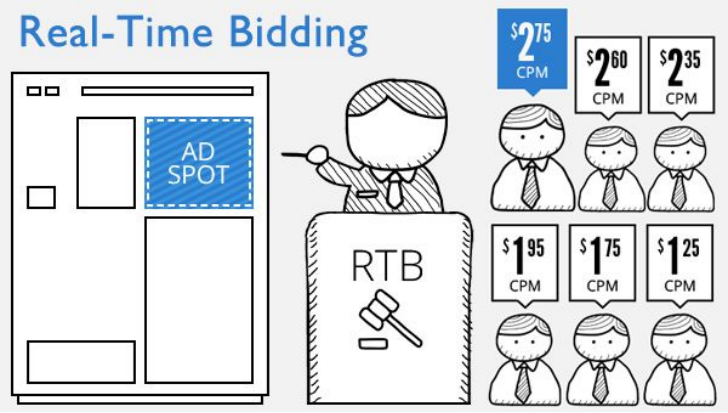
BROADCOM

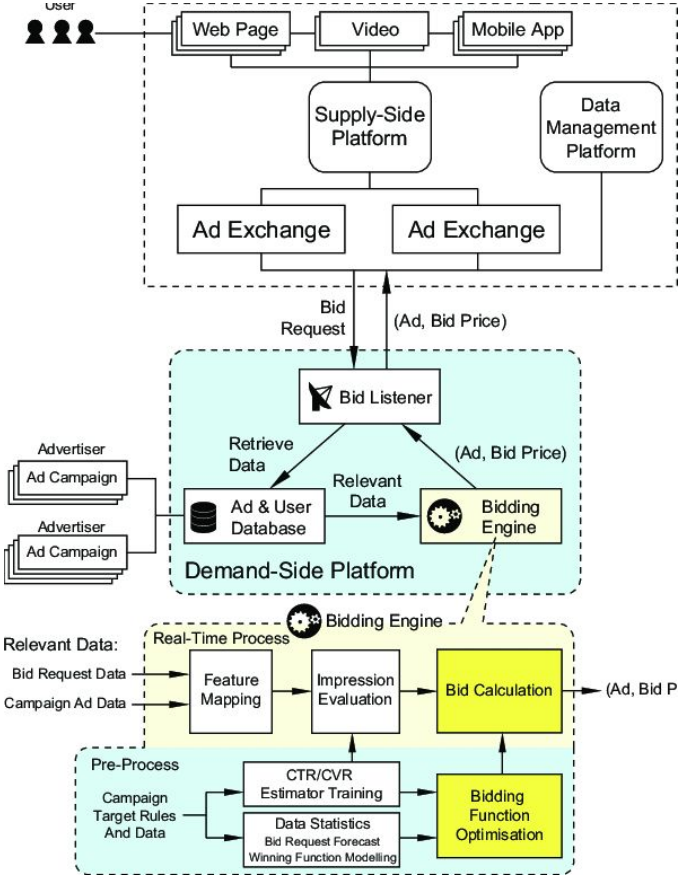
Yeah, threads!!!



BROADCOM

Realtime bidding for Ad spots





BROADCOM

Telemetry

BROADCOM