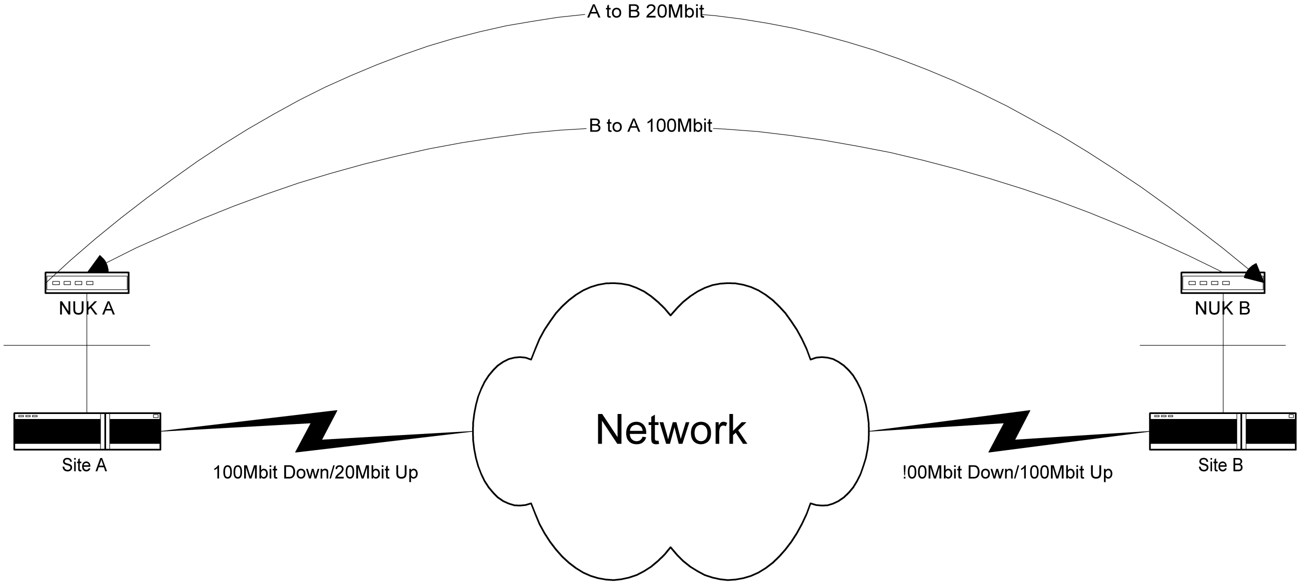
NUK Documentation – Bandwidth testing

Introduction

Bandwidth testing uses an application called iPerf. This requires iPerf running as a service on one NUK and running as a client on another NUK. We need to briefly touch on upload and download speeds to understand what results to expect from iPerf.

If you have a site that has bandwidth as 100Mbit download / 20Mbit upload, then if you run iPerf as a server on the NUK at that site, then you should expect 100Mbit speeds. If you run iPerf as a client to another site with a NUK then you should expect 20Mbit speeds.

This is because sending traffic to the site is a download (from that site's perspective) whereas sending traffic from that site to another is considered an upload (again from that site's perspective). Here's a picture to help you understand:



So to do bandwidth testing we need to load iPerf as a server on one site, then use iperf as a client to test the bandwidth to that site.

Running iPerf as a Server

To load iPerf as a server, you can use Nettools2 from the VNC menu (see NUK Documentation – VNC Menu), load iPerf as a service using Start-Services from the VNC menu, or make sure iPerf as a server is running when the NUK starts using Startup-opts from the VNC menu. If you want to add any options, then you need to run iperf-server from Net-tools2 and enter the options there. The options are set in Net-tools2, but they will take effect no matter how you start iperf. Here are all the available options:

```
-p, --port      #         server port to listen on/connect to
-f, --format    [kmgKMG]  format to report: Kbits, Mbits, KBytes, MBytes
-i, --interval  #         seconds between periodic bandwidth reports
-F, --file name           xmit/recv the specified file
-A, --affinity n/n,m      set CPU affinity
-B, --bind      <host>    bind to a specific interface
-V, --verbose             more detailed output
-J, --json                output in JSON format
-d, --debug               emit debugging output
-v, --version             show version information and quit
-h, --help                show this message and quit
-D, --daemon              run the server as a daemon
-1, --one-off             handle one client connection then exit
```

You really shouldn't need any options. You might want to change the time between bandwidth reports ("-i n"), or change what unit of bandwidth is used ("-f [kmgKMG]").. DON'T start iperf-server as a daemon as you won't be able to stop it if you want to. Once you have iPerf running you should see this. This shows the "Server listening on 5201" message as well as a 3Mbit UDP connection between two sites on the East Coast.

```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from <East Coast2>, port 56633
[  5] local <Upper East Coast> port 5201 connected to <East Coast2> port 31936
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-1.00   sec   216 KBytes  1.76 Mbits/sec  620.465 ms  0/27 (0%)
[  5]   1.00-2.01   sec   352 KBytes  2.86 Mbits/sec  70.082 ms  0/44 (0%)
[  5]   2.01-3.01   sec   344 KBytes  2.81 Mbits/sec  36.200 ms  0/43 (0%)
[  5]   3.01-4.01   sec   360 KBytes  2.95 Mbits/sec  34.670 ms  0/45 (0%)
[  5]   4.01-5.00   sec   352 KBytes  2.91 Mbits/sec  32.327 ms  0/44 (0%)
[  5]   5.00-6.01   sec   352 KBytes  2.87 Mbits/sec  34.447 ms  0/44 (0%)
[  5]   6.01-7.00   sec   360 KBytes  2.97 Mbits/sec  35.554 ms  0/45 (0%)
[  5]   7.00-8.01   sec   352 KBytes  2.86 Mbits/sec  36.560 ms  0/44 (0%)
[  5]   8.01-9.01   sec   368 KBytes  3.01 Mbits/sec  36.058 ms  0/46 (0%)
[  5]   9.01-10.01  sec   352 KBytes  2.89 Mbits/sec  34.677 ms  0/44 (0%)
[  5]  10.01-10.62  sec   216 KBytes  2.92 Mbits/sec  34.125 ms  0/27 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-10.62  sec  0.00 Bytes  0.00 bits/sec  34.125 ms  0/453 (0%)
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

The "Server listening on 5201" part is the important part. Watch out for lines like "Segmentation fault – program aborted" or any other line that suggests that iPerf isn't running. "A segmentation fault is a fatal error and if you see it you'll know iPerf isn't running. The above test is You can also test that iPerf is running by using Putty (or telnet on any Unix machine or Mac) to connect to the NUK on port 5201.

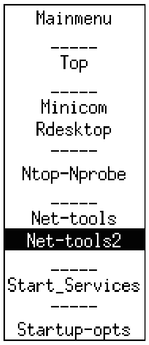
If you don't get an error, iPerf is running. But if you do this test, you'll see the following in the window running iPerf as a server:

```
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
WARNING:  Size of data read does not correspond to offered length
iperf3: error - unable to receive parameters from client:
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Now there is an error message, because you connected to iPerf's port but didn't act as an iPerf client, but the “Server listening on 5201” lets you know that everything is fine. You probably will see messages like this if your server is connected to the public Internet, but as long as you see the “Server listening on 5201” everything is fine.

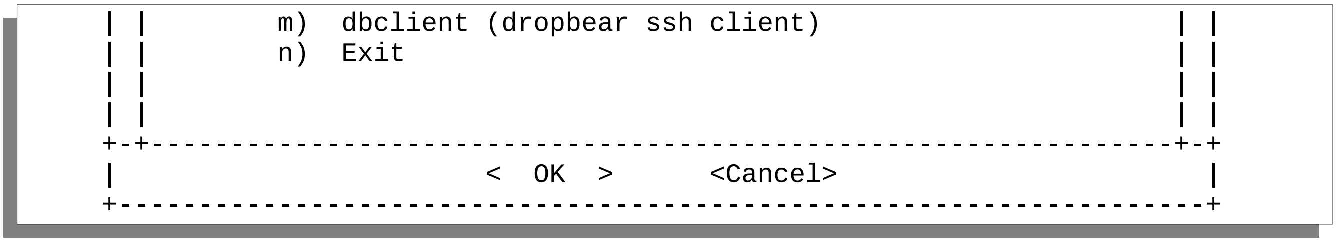
Running iPerf as a client

Now you'll go to the sending NUK, the NUK running iPerf client. You'll start this by using the nettools2 menu:



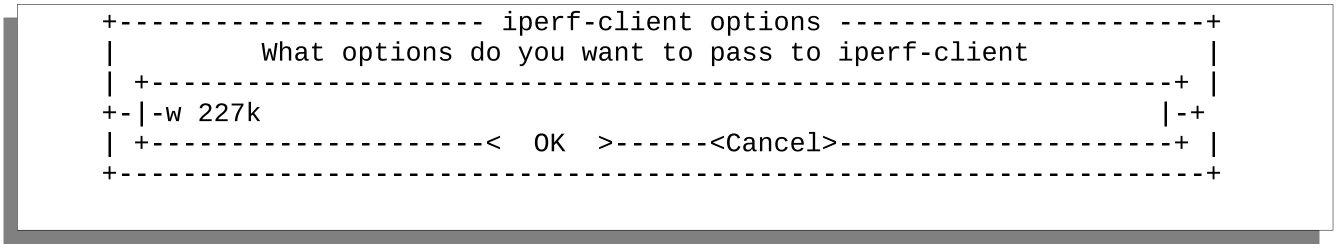
Which brings up this menu:

```
     +------------------------- Network Tools --------------------------+
     |                Please choose from the following:                 |
     | +--------------------------------------------------------------+ |
     | |        a)  mtr (continously updating traceroute)             | |
     | |        b)  nbtscan (Look up Netbios names)                   | |
     | |        c)  iPerf server (throughput testing)                 | |
     | |        d)  iPerf client (throughput testing)                 | |
     | |        e)  nmap (port scanner)                               | |
     | |        f)  rdesktop                                          | |
     | |        g)  rinetd                                            | |
     | |        h)  tinyproxy                                         | |
     | |        i)  iperf server - udp mode (throughput testing)      | |
     | |        j)  flood ping (5 seconds)                            | |
     | |        k)  telnet                                            | |
     | |        l)  nslookup                                          | |
```



```
| |        m)  dbclient (dropbear ssh client)                       | |
| |        n)  Exit                                                 | |
| |                                                                 | |
| |                                                                 | |
+-+-----------------------------------------------------------------+-+
|                       <  OK  >      <Cancel>                        |
+---------------------------------------------------------------------+
```

Then choose choice d). You'll see an options window:



```
+----------------------- iperf-client options ------------------------+
|         What options do you want to pass to iperf-client            |
| +-----------------------------------------------------------------+ |
+-|-w 227k                                                          |-+
| +---------------------<  OK  >------<Cancel>----------------------+ |
+---------------------------------------------------------------------+
```

Here's a list of all of the possible iperf-client options:

```
  -p, --port      #         server port to listen on/connect to
  -f, --format    [kmgKMG]  format to report: Kbits, Mbits, KBytes, MBytes
  -i, --interval  #         seconds between periodic bandwidth reports
  -F, --file name           xmit/recv the specified file
  -A, --affinity n/n,m      set CPU affinity
  -B, --bind      <host>    bind to a specific interface
  -V, --verbose             more detailed output
  -J, --json                output in JSON format
  -d, --debug               emit debugging output
  -v, --version             show version information and quit
  -h, --help                show this message and quit
  -u, --udp                 use UDP rather than TCP
  -b, --bandwidth #[KMG][/#] target bandwidth in bits/sec (0 for unlimited)
                            (default 1 Mbit/sec for UDP, unlimited for TCP)
                            (optional slash and packet count for burst mode)
  -t, --time      #         time in seconds to transmit for (default 10 secs)
  -n, --bytes     #[KMG]    number of bytes to transmit (instead of -t)
  -k, --blockcount #[KMG]   number of blocks (packets) to transmit (instead of -t
or -n)
  -l, --len       #[KMG]    length of buffer to read or write
                            (default 128 KB for TCP, 8 KB for UDP)
  -P, --parallel  #         number of parallel client streams to run
  -R, --reverse             run in reverse mode (server sends, client receives)
  -w, --window    #[KMG]    set window size / socket buffer size
  -M, --set-mss   #         set TCP maximum segment size (MTU - 40 bytes)
  -N, --nodelay             set TCP no delay, disabling Nagle's Algorithm
  -4, --version4            only use IPv4
  -6, --version6            only use IPv6
  -S, --tos N               set the IP 'type of service'
  -Z, --zerocopy            use a 'zero copy' method of sending data
  -O, --omit N              omit the first n seconds
  -T, --title str           prefix every output line with this string
  --get-server-output       get results from server
```

“-w 227k” sets the send window for iperf-client. 227K is the highest window the NUK can use. Other platforms can possibly go higher. Other options are:

“-t n”   run for n seconds

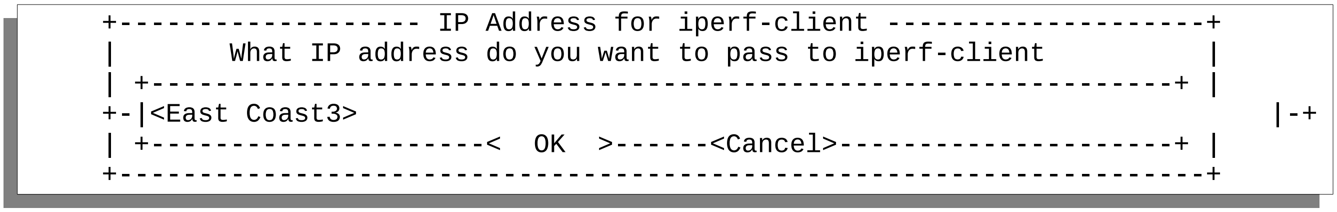
“-i n”   report the bandwidth achieved every n seconds

“-R”   Reverse mode (server sends to client)

“-O n” Ignore the first n seconds. This option is very useful because the first 2 seconds usually contain the most packet drops (especially in TCP mode)

“--get-server-output” This retrieves the output from iperf-server on the far end NUK, which saves you the trouble of connecting to said NUK

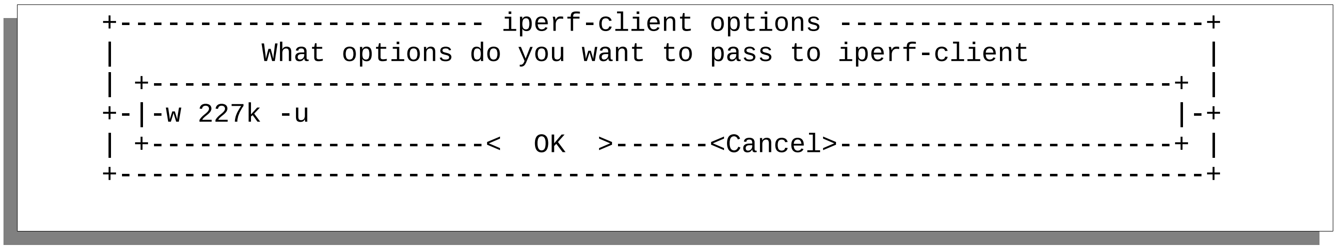
Then you're asked which IP address to connect to:



```
     +------------------- IP Address for iperf-client ---------------------+
     |       What IP address do you want to pass to iperf-client           |
     | +-----------------------------------------------------------------+ |
     +-|<East Coast3>                                                         |-+
     | +---------------------<  OK  >------<Cancel>----------------------+ |
     +---------------------------------------------------------------------+
```

This would run iperf in TCP mode connecting to the server <East Coast3>

If you want to use UDP mode add “-u” in the options window:



```
     +----------------------- iperf-client options ------------------------+
     |         What options do you want to pass to iperf-client            |
     | +-----------------------------------------------------------------+ |
     +-|-w 227k -u                                                      |-+
     | +---------------------<  OK  >------<Cancel>----------------------+ |
     +---------------------------------------------------------------------+
```

and add “-b <bandwidth>” after the IP address. Examples of -b:

-b 75M means 75Mbit/sec

-b 200k means 200kbit/sec

So the window would look like this:

```
     +------------------- IP Address for iperf-client --------------------+
     |       What IP address do you want to pass to iperf-client          |
     | +----------------------------------------------------------------+ |
     +-|<East Coast3> -b 3M                                              |-+
     | +---------------------<  OK  >------<Cancel>---------------------+ |
     +--------------------------------------------------------------------+
```

Which means that it would connect to the IP address <East Coast3> in UDP mode and send data at 3Mbit/second.

```
Fri Jul  7 20:32:14 GMT 2017
Connecting to host <Upper East Coast2>, port 5201
[  4] local <East Coast3> port 64204 connected to <Upper East Coast2> port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.01   sec   336 KBytes  2.72 Mbits/sec  42
[  4]   1.01-2.02   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   2.02-3.02   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   3.02-4.02   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   4.02-5.01   sec   360 KBytes  2.97 Mbits/sec  45
[  4]   5.01-6.01   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   6.01-7.01   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   7.01-8.02   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   8.02-9.02   sec   368 KBytes  3.01 Mbits/sec  46
[  4]   9.02-10.02  sec   368 KBytes  3.01 Mbits/sec  46
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  4]   0.00-10.02  sec  3.55 MBytes  2.98 Mbits/sec  8.529 ms  0/455 (0%)
[  4] Sent 455 datagrams

iperf Done.
Fri Jul  7 20:32:25 GMT 2017
```

The most important part of the output is bolded and in a larger font. As long as that number is 0, then the Bandwidth number is valid. If that number and percentage get high, you have to multiply 100 minus that percentage by the bandwidth number to get the actual bandwidth achieved. For example if we up the bandwidth considerably, we'll see slightly different results:

```
Fri Jul 14 22:57:31 GMT 2017
Connecting to host 74.103.91.203, port 5201
[  4] local <East Coast> port 65533 connected to 74.103.91.203 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec  8.23 MBytes  68.9 Mbits/sec  5910  (omitted)
[  4]   1.00-2.01   sec  9.09 MBytes  75.3 Mbits/sec  6525  (omitted)
[  4]   0.00-1.00   sec  9.19 MBytes  77.1 Mbits/sec  6603
[  4]   1.00-2.00   sec  9.01 MBytes  75.6 Mbits/sec  6469
[  4]   2.00-3.01   sec  8.56 MBytes  71.4 Mbits/sec  6149
[  4]   3.01-4.00   sec  9.09 MBytes  76.7 Mbits/sec  6529
[  4]   4.00-5.00   sec  8.85 MBytes  74.3 Mbits/sec  6357
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  4]   0.00-5.00   sec  44.7 MBytes  75.0 Mbits/sec  0.167 ms  268/32104 (0.83%)
[  4] Sent 32104 datagrams
```

```
iperf Done.
Fri Jul 14 22:57:39 GMT 2017
```

In this instance, the actual bandwidth achieved is (100-0.83)% or 99.17% of 75 Mbits/sec or 74.38Mbit. In cases where the NUK's are close together, geographically, or are on the same network provider, then there can be situations where there are no dropped (lost) packets. The example above involves two sites that are within 15 miles of each other.

Here are the options passed to iperf-client:

```
    +----------------------- iperf-client options -----------------------+
    |         What options do you want to pass to iperf-client            |
    | +-----------------------------------------------------------------+ |
    +-|-O 2 -t 5 -u --get-server-output                                 |-+
    | +---------------------<  OK  >------<Cancel>---------------------+ |
    +---------------------------------------------------------------------+
```

Interpreting the options from left to right, we get: “Ignore the first two seconds of testing, test for 5 seconds, use UDP mode, and get the output from the server”

```
Fri Jul 14 23:16:27 GMT 2017
Connecting to host 74.103.91.203, port 5201
[  4] local <East Coast> port 65529 connected to 74.103.91.203 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.01   sec  8.90 MBytes  73.8 Mbits/sec  6390  (omitted)
[  4]   1.01-2.01   sec  9.59 MBytes  80.3 Mbits/sec  6886  (omitted)
[  4]   0.00-1.00   sec  9.43 MBytes  79.0 Mbits/sec  6773
[  4]   1.00-2.00   sec  9.55 MBytes  80.0 Mbits/sec  6862
[  4]   2.00-3.01   sec  9.54 MBytes  79.9 Mbits/sec  6850
[  4]   3.01-4.00   sec  9.69 MBytes  81.7 Mbits/sec  6957
[  4]   4.00-5.00   sec  9.42 MBytes  79.1 Mbits/sec  6768
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  4]   0.00-5.00   sec  47.6 MBytes  79.9 Mbits/sec  0.165 ms  666/34210 (1.9%)
[  4] Sent 34210 datagrams

Server output:
Accepted connection from <East Coast>, port 65530
[  5] local 74.103.91.203 port 5201 connected to <East Coast> port 65529
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-1.00   sec  6.59 MBytes  55.2 Mbits/sec  0.255 ms  212/4942 (4.3%)
(omitted)
[  5]   1.00-2.00   sec  9.52 MBytes  79.9 Mbits/sec  0.157 ms  1/6837 (0.015%)
(omitted)
[  5]   0.00-1.00   sec  9.42 MBytes  79.0 Mbits/sec  0.120 ms  83/6848 (1.2%)
[  5]   1.00-2.00   sec  9.44 MBytes  79.2 Mbits/sec  0.150 ms  62/6844 (0.91%)
[  5]   2.00-3.01   sec  9.36 MBytes  77.8 Mbits/sec  0.129 ms  171/6896 (2.5%)
[  5]   3.01-4.00   sec  9.29 MBytes  78.7 Mbits/sec  0.154 ms  137/6806 (2%)
[  5]   4.00-5.00   sec  9.52 MBytes  79.8 Mbits/sec  0.155 ms  0/6837 (0%)
[  5]   5.00-5.23   sec  2.06 MBytes  76.2 Mbits/sec  0.165 ms  0/1476 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-5.23   sec  0.00 Bytes  0.00 bits/sec  0.165 ms  453/35707 (1.3%)


iperf Done.
Fri Jul 14 23:16:35 GMT 2017
Press Ctrl-C to close this window.
```

The server output is in italics, in the above text and all output below. You can see that there are some intervals where there are no dropped (lost) packets, and another interval where there was 2.5% dropped packets. This is due to the ebb and flow of traffic on the Internet.

Now below is an example where the locations are 2600 miles away. Strangely enough we see similar results.

```


Fri Jul 14 23:39:55 GMT 2017
Connecting to host <East Coast>, port 5201
[  4] local <West Coast> port 65528 connected to <East Coast> port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.01   sec  8.32 MBytes  69.3 Mbits/sec  7075  (omitted)
[  4]   1.01-2.01   sec  8.95 MBytes  75.0 Mbits/sec  6431  (omitted)
[  4]   0.00-1.00   sec  8.96 MBytes  75.0 Mbits/sec  6432
[  4]   1.01-2.00   sec  9.03 MBytes  76.4 Mbits/sec  6488
[  4]   2.00-3.02   sec  8.88 MBytes  73.6 Mbits/sec  6376
[  4]   3.02-4.01   sec  9.06 MBytes  76.6 Mbits/sec  6507
[  4]   4.01-5.01   sec  8.85 MBytes  74.1 Mbits/sec  6354
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  4]   0.00-5.01   sec  44.7 MBytes  74.9 Mbits/sec  0.095 ms  502/32103 (1.6%)
[  4] Sent 32103 datagrams





Server output:
Accepted connection from <East Coast>, port 65529
[  5] local <West Coast> port 5201 connected to <East Coast> port 65528
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-1.00   sec  5.64 MBytes  47.3 Mbits/sec  0.251 ms  186/4238 (4.4%)
(omitted)
[  5]   1.00-2.00   sec  8.86 MBytes  74.4 Mbits/sec  0.156 ms  56/6418 (0.87%)
(omitted)
[  5]   0.00-1.00   sec  8.82 MBytes  74.0 Mbits/sec  0.117 ms  77/6414 (1.2%)
[  5]   1.00-2.00   sec  8.83 MBytes  74.0 Mbits/sec  0.100 ms  47/6386 (0.74%)
[  5]   2.00-3.00   sec  9.01 MBytes  75.5 Mbits/sec  0.125 ms  44/6513 (0.68%)
[  5]   3.00-4.00   sec  8.99 MBytes  75.4 Mbits/sec  0.093 ms  31/6486 (0.48%)
[  5]   4.00-5.00   sec  8.68 MBytes  72.8 Mbits/sec  0.178 ms  61/6297 (0.97%)
[  5]   5.00-5.26   sec  2.44 MBytes  77.4 Mbits/sec  0.095 ms  0/1754 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-5.26   sec  0.00 Bytes  0.00 bits/sec  0.095 ms  260/33850 (0.77%)



iperf Done.
Fri Jul 14 23:41:25 GMT 2017
```

But if we reverse the connection (from West to East, rather than East to West), we see a different picture:

```
Fri Jul 14 23:39:55 GMT 2017
Connecting to host <East Coast>, port 5201
[  4] local <West Coast> port 65528 connected to <East Coast> port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.01   sec  8.32 MBytes  69.3 Mbits/sec  7075  (omitted)
[  4]   1.01-2.01   sec  8.95 MBytes  75.0 Mbits/sec  6431  (omitted)
[  4]   0.00-1.00   sec  8.96 MBytes  75.0 Mbits/sec  6432
[  4]   1.00-2.00   sec  8.96 MBytes  75.0 Mbits/sec  6433
[  4]   2.00-3.00   sec  9.16 MBytes  77.1 Mbits/sec  6577
[  4]   3.00-4.01   sec  8.73 MBytes  72.7 Mbits/sec  6269
[  4]   4.01-5.01   sec  8.96 MBytes  75.0 Mbits/sec  6434
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  4]   0.00-5.01   sec  44.8 MBytes  75.0 Mbits/sec  0.135 ms  5429/32144 (17%)
[  4] Sent 32144 datagrams

Server output:
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from <West Coast>, port 65522
[  5] local <East Coast> port 5201 connected to <West Coast> port 65528
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-1.00   sec  5.08 MBytes  42.6 Mbits/sec  0.163 ms  1644/5293 (31%)
(omitted)
[  5]   1.00-2.00   sec  7.95 MBytes  66.7 Mbits/sec  0.124 ms  716/6427 (11%)
(omitted)
[  5]   0.00-1.00   sec  8.22 MBytes  68.9 Mbits/sec  0.128 ms  575/6477 (8.9%)
[  5]   1.00-2.00   sec  8.27 MBytes  69.4 Mbits/sec  0.142 ms  461/6399 (7.2%)
[  5]   2.00-3.00   sec  7.92 MBytes  66.4 Mbits/sec  0.149 ms  644/6331 (10%)
[  5]   3.00-4.00   sec  8.23 MBytes  69.1 Mbits/sec  0.106 ms  605/6519 (9.3%)
[  5]   4.00-5.00   sec  8.07 MBytes  67.7 Mbits/sec  0.122 ms  620/6414 (9.7%)
[  5]   5.00-5.28   sec  2.26 MBytes  68.6 Mbits/sec  0.135 ms  164/1790 (9.2%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total
Datagrams
[  5]   0.00-5.28   sec  0.00 Bytes  0.00 bits/sec  0.135 ms  3069/33930 (9%)
```

```


iperf Done.
Fri Jul 14 23:40:03 GMT 2017
```

First of all, the West Coast side sees 17% dropped (lost) packets, whereas the East Coast side sees 9% (note the 31% in the first interval, which was ignored (omitted). I dropped the bandwidth down to 3M, but still saw similar packet loss. The lesson to learn is that the Internet is complicated and has bottlenecks, especially when you have to cross the country. To show the reason for using UDP mode and looking at all of the lost packets, we'll try it in TCP mode. The results are laughable, so we'll leave off the –get-server-output, option.

```
Sat Jul 15 00:04:17 GMT 2017
Connecting to host <East Coast>, port 5201
[  4] local <West Coast> port 65514 connected to <East Coast> port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec   115 KBytes   944 Kbits/sec                  (omitted)
[  4]   1.00-2.01   sec  0.00 Bytes  0.00 bits/sec                  (omitted)
[  4]   0.00-1.01   sec  0.00 Bytes  0.00 bits/sec
[  4]   1.01-2.01   sec  0.00 Bytes  0.00 bits/sec
[  4]   2.01-3.01   sec  9.90 KBytes  81.8 Kbits/sec
[  4]   3.01-4.01   sec  4.24 KBytes  34.7 Kbits/sec
[  4]   4.01-5.01   sec  5.66 KBytes  46.3 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-5.01   sec  19.8 KBytes  32.4 Kbits/sec                  sender
[  4]   0.00-5.01   sec  19.8 KBytes  32.4 Kbits/sec                  receiver
```

We get a grand total of 32.4 KILObit per second :) Certainly not the best the circuits can do. Now reversing the direction helps, but the max we get is 22.3Mbit/second in the last interval whereas with UDP mode we were getting over 99% of 75Mbit/second.

```
Sat Jul 15 00:06:06 GMT 2017
Connecting to host <West Coast>, port 5201
[  4] local <East Coast> port 65527 connected to <West Coast> port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.01   sec   612 KBytes  4.98 Mbits/sec                  (omitted)
[  4]   1.01-2.01   sec  1.13 MBytes  9.50 Mbits/sec                  (omitted)
[  4]   0.00-1.00   sec  1.45 MBytes  12.1 Mbits/sec
[  4]   1.00-2.01   sec  1.85 MBytes  15.5 Mbits/sec
[  4]   2.01-3.02   sec  2.04 MBytes  17.0 Mbits/sec
[  4]   3.02-4.01   sec  2.44 MBytes  20.6 Mbits/sec
[  4]   4.01-5.00   sec  2.65 MBytes  22.3 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-5.00   sec  10.4 MBytes  17.5 Mbits/sec                  sender
[  4]   0.00-5.00   sec  10.7 MBytes  18.0 Mbits/sec                  receiver

iperf Done.
Sat Jul 15 00:06:16 GMT 2017
```

Here is a table of IP addresses assigned to NUK's that you can refer to as you run iPerf.

NUK Location	NUK Mgmt IP	NUK Mgmt range	NUK Monitor1 IP	NUK Monitor 2 IP