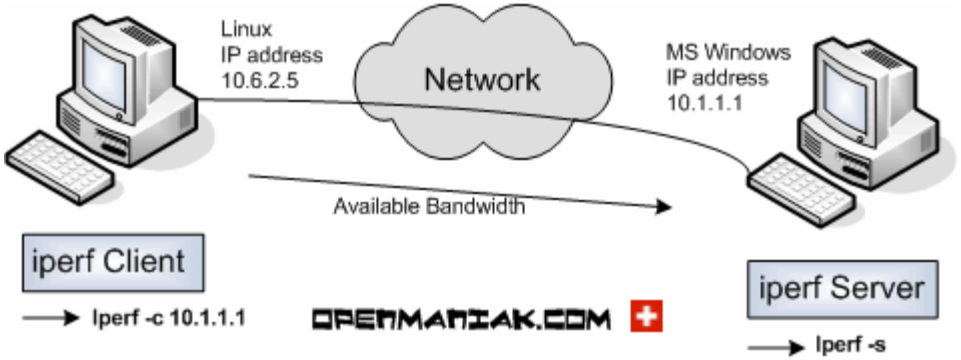
iperf a gnu tool for IP networks

Oscar Mederos



- iperf – a GNU tool for testing network throughput & capacity
- clinet/server model with support for biderectional communication
- Support for tcp and udp data streams
- Tcp – transmission control protocol slower but verifies traffic arrived at sender (email, http, etc.)
- Udp – user datagram protocol, faster than tcp but no verification that packet arrived at destination (VoIP, nfs, tftp)

iperf

- Bandwidth is best measured through tcp (because of syn/ack)
- To start a simple 1 way test have 2 machines on a network(including the internet) running iperf one as client and one as server with: iperf -c **ip of server** and one as: iperf -s
- Iperf client will create a tcp connection over port 50001 to the server

- Client:

```
oscar@vector:~$ iperf -c wolf.cs.fiu.edu
------------------------------------------------------------
Client connecting to wolf.cs.fiu.edu, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 131.94.133.55 port 54391 connected with 131.94.130.46 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec    103 MBytes  86.7 Mbits/sec
```

Server response:

```
omede001@wolf:/homes/esj/work/benchmarks/iperf-2.0.2 16% ./iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 131.94.130.46 port 5001 connected with 131.94.133.55 port
    54391
[  4]  0.0-10.0 sec    103 MBytes  86.4 Mbits/sec
```

Useful switches:

-d for bidirectional, by default only client – server is measured

-w (size) change tcp window size between 2 and 65,535 bytes, On Linux systems, when specifying a TCP buffer size with the -w argument, the kernel allocates double as much as indicated.

Communication port (-p), timing (-t) and interval (-i) the -p must be set on both the client and server connection strings.

UDP

The UDP tests with the -u argument will give invaluable information about jitter and the packet loss.

The jitter value is particularly important on network links supporting voice over IP (VoIP) because a high jitter can break a call.The -b argument allows the allocation if the desired bandwidth.

UDP server

```
omede001@wolf:/homes/esj/work/benchmarks/iperf-2.0.2 17% ./iperf -sui 1
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:   108 KByte (default)
------------------------------------------------------------
[  3] local 131.94.130.46 port 5001 connected with 131.94.133.55 port 39078
[  3]  0.0- 1.0 sec  1.19 MBytes  10.0 Mbits/sec  0.031 ms    0/  850 (0%)
[  3]  1.0- 2.0 sec  1.19 MBytes  10.0 Mbits/sec  0.046 ms    0/  850 (0%)
[  3]  2.0- 3.0 sec  1.19 MBytes  10.0 Mbits/sec  0.036 ms    0/  850 (0%)
[  3]  3.0- 4.0 sec  1.19 MBytes  10.0 Mbits/sec  0.923 ms    0/  850 (0%)
[  3]  4.0- 5.0 sec  1.19 MBytes  10.0 Mbits/sec  0.030 ms    0/  851 (0%)
[  3]  5.0- 6.0 sec  1.19 MBytes  10.0 Mbits/sec  0.037 ms    0/  850 (0%)
[  3]  6.0- 7.0 sec  1.19 MBytes  10.0 Mbits/sec  0.035 ms    0/  850 (0%)
[  3]  7.0- 8.0 sec  1.19 MBytes  10.0 Mbits/sec  0.032 ms    0/  850 (0%)
[  3]  8.0- 9.0 sec  1.19 MBytes  10.0 Mbits/sec  0.041 ms    0/  851 (0%)
[  3]  9.0-10.0 sec  1.19 MBytes  10.0 Mbits/sec  0.310 ms    0/  850 (0%)
[  3]  0.0-10.0 sec  11.9 MBytes  10.0 Mbits/sec  0.261 ms    0/ 8504 (0%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
```

UDP client

```
oscar@vector:~$ iperf -c wolf -ub 10m
------------------------------------------------------------
Client connecting to wolf, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:   112 KByte (default)
------------------------------------------------------------
[  3] local 131.94.133.55 port 39078 connected with 131.94.130.46 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  11.9 MBytes  10.0 Mbits/sec
[  3] Sent 8505 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec  11.9 MBytes  10.0 Mbits/sec  0.261 ms    0/ 8504 (0%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
```