iPerf3 (Linux)	
Command	Description
iperf3 -s	Start iPerf3 server mode to listen for client connections
iperf3 -s -B *ip address*	Bind ingress traffic to an interface when server is multihomed
iperf3 -s -p 443	Assign non-default listening port on server
iperf3 -s -F *filename*	Write a file from the network traffic to server disk
iperf3 -c *ip address or hostname*	Start iPerf3 client mode to server ip address or hostname
iperf3 -c *ip address* -p 443	Assign non-default destination port to match server
iperf3 -c *ip address* --cport 2000	Set client source port to TCP 2000 instead of a dynamic port
iperf3 -c *ip address* -t 20	Set test duration to 20 seconds
iperf3 -c *ip address* -i 2	Set reporting interval to 2 seconds
iperf3 -c *ip address* -O 2	Allow TCP slow start to finish (2 sec) before collecting results
iperf3 -c *ip address* -V	Display verbose mode (detailed) test results
iperf3 -c *ip address* -w 230K	Set TCP window size (socket buffer) to 230 KB
iperf3 -c *ip address* -M 1360	Set TCP MSS size to 1360 bytes for VPN tunnel
iperf3 -c *ip address* -b 10M (or 10000K)	Set TCP maximum bandwidth (bit rate) to 10 Mbps
iperf3 -c *ip address* -l 256K	Set TCP read/write buffer size to 256 KB
iperf3 -c *ip address* -P 10	Specify 10 TCP parallel streams with multi-threading
iperf3 -c *ip address* --sctp	Specify SCTP as transport protocol for test
iperf3 -c *ip address* -u	Specify UDP as transport protocol for test
iperf3 -c *ip address* -u -b 1000M (or 1G)	Set UDP maximum bandwidth (bit rate) to 1000 Mbps
Iperf3 -c *ip address* -u -b 0	Set unlimited bandwidth (bit rate) for UDP protocol
iperf3 -c *ip address* -u -l 1472	Set UDP read/write buffer size (packet size) to 1472 bytes
Iperf3 -c *ip address* -u -w 230K	Set UDP socket buffer size to 230 KB
iperf3 -c *ip address* -R	Specify reverse mode testing from server to client
iperf3 -c *ip address* --bidir	Specify bidirectional mode testing (separate sockets)
iperf3 -c *ip address* -4	Specify IPv4 only addressing for connection
iperf3 -c *ip address* -6	Specify IPv6 only addressing for connection
iperf3 -c *ip address* –N	Disable Nagle algorithm for test
iperf3 -c *ip address* -F *filename*	Read a file from client disk to the network
iperf3 -c -B *ip address*	Bind egress traffic to an interface when client is multihomed
iperf3 -c *ip address* --logfile *filename*	Redirect test results to a log file for review
iperf3 -c *ip address* --get-server-output	Display server-side results on client console

iPerf2 (Linux / Windows)	
Command	Description
iperf -s	Start iPerf2 server mode to listen for client connections
iperf -s -p 443	Assign non-default listening port on server
iperf -s -i 1	Set reporting interval to 1 second on server
iperf -s -V *ipv6 address*	Bind test to an IPv6 address on server
iperf -s -w 230K	Set TCP window size (socket buffer) to 230 KB on server
iperf -s -l 256K	Set TCP read/write buffer size to 256 KB on server
iperf -s -u	Specify UDP as transport protocol on server
iperf -s -u -w 230K	Set UDP socket buffer size to 230 KB on server
iperf -s -u -l 1472	Set UDP read/write buffer size (packet size) to 1472 bytes
iperf -s -B *ip address*	Bind an interface for multicast testing or multihomed server
iperf -s -u -B 239.1.1.1 -i 1	Bind server to join multicast group 239.1.1.1
iperf -c *ip address* or *hostname*	Start iPerf client mode to server ip address or hostname
iperf -c *ip address* -p 443	Assign non-default TCP destination port to match server
iperf -c -V *ipv6 address*	Bind test to an IPv6 address on client
iperf -c *ip address* -t 20	Set test duration to 20 seconds
iperf -c *ip address* -i 1	Set reporting interval to 1 second on client
iperf -c *ip address* -e	Enable enhanced reporting for test
iperf -c *ip address* -w 230K	Set TCP window size (socket buffer) to 230 KB on client
iperf -c *ip address* -M 1360	Set TCP MSS size to 1360 bytes for VPN tunnel
iperf -c *ip address* -m	Display TCP MSS size used for test
iperf –c *ip address* --connect-only -e	Measure TCP handshake connection time only (no data)
iperf -c *ip address* -l 256K	Set TCP read/write buffer size to 256 KB on client
iperf -c *ip address* -u	Specify UDP as transport protocol on client
iperf -c *ip address* -u -l 1472	Set UDP read/write buffer size to 1472 bytes on client
iperf -c *ip address* -u -w 230K	Set UDP socket buffer size to 230 KB on client
iperf -c *ip address* -u -b 100M	Set UDP maximum bandwidth (bit rate) to 100 Mbps
iperf -c *ip address* -P 10	Specify 10 TCP parallel streams with multi-threading
iperf -c *ip address* --full-duplex	Specify full duplex simultaneous mode (same socket)
iperf -c *ip address* -d	Specify simultaneous dualtest mode (separate sockets)
iperf -c *ip address* -r	Specify tradeoff alternating bidirectional mode
iperf -c *ip address* -R	Specify reverse mode testing from server to client

iperf -c *ip address* -N	Disable Nagle algorithm for test
iperf -c -B *ip address*	Bind egress traffic to an interface when client is multihomed
iperf -c 239.1.1.1 -u -T 10 -t 5 -i 1	Generate traffic from client to multicast group 239.1.1.1
iperf-2.2.1-win64 -s (Windows)	Start iPerf2 server mode to listen for client connections
iperf-2.2.1-win64 -c *ip address* or *hostname*	Start iPerf2 client mode to server ip address or hostname
iperf-2.2.1-win64 -s -i 1 -o *filename*	Redirect output of test results to a file
iperf-2.2.1-win64 -help	Help facility for Windows iPerf2

- Throughput and bit rate are reported as upload from client and download to server (default).
- TCP MSS (-M) 1448 bytes packet size = 1460 bytes - 12 byte TCP timestamp option.
- TCP auto-tuning is recommended unless troubleshooting since TCP adapts to network conditions.
- Socket buffer size (-w) = TCP window size and UDP socket buffer size.
- Linux TCP/UDP maximum socket buffer size (-w) is 416 KB.
- Calculate bandwidth delay product (bdp) for optimal TCP window size only.
- Linux TCP default read/write buffer size (-l) is 128 KB and UDP is 1448 bytes (packet size).
- iPerf3 TCP maximum read/write buffer size (-l) is 1 MB and UDP is 65507 bytes. (packet size).
- UDP (-u), socket buffer size (-w), and read/write buffer size (-l) sent to server via control channel (iPerf3).
- UDP (-u), socket buffer size (-w), and read/write buffer size (-l) configured separately on server (iPerf2).
- Set the reporting interval (-i) on client and server for iPerf2 (default 0).
- Bind interface (-B) is the outbound interface on client and inbound interface on server.
- Configure Linux host firewall to allow TCP 5201/UDP 5201 (iPerf3) and TCP 5001 (iPerf2).

Feature	Command	iPerf3	iPerf2	Notes
TCP	default	●	●	unlimited bit rate from client to server (-b modify)
UDP	-u	●	●	1 Mbps bit rate from client to server (-b modify)
SCTP	--sctp	●		use SCTP instead of TCP (iPerf 3.1+)
bandwidth	-b	●	●	specify TCP or UDP bit rate (iPerf2 UDP only)
server listening port	-p	●	●	specify the same non-default port on client and server allow TCP 5201 / UDP 5201 through firewall (iPerf3) allow TCP 5001 through firewall (iPerf2)
client port	--cport	●		specify static client-side port instead of a dynamic port
report interval	-i	●	●	iPerf2 assigns zero (0) unless value is specified iPerf3 assigns 1 second report intervals by default
test duration	-t	●	●	specify number of seconds to run test (default 10)
socket buffer size	-w	●	●	TCP receive window maximum size UDP socket buffer maximum size configured separately on client and server for iPerf2
parallel streams	-P	●	●	multi-threaded with iPerf2 and iPerf 3.16+
reverse mode	-R	●	●	server to client direction (same socket) NAT and Firewall traversal support
bidirectional mode	--bidir	●		bidirectional simultaneous (separate sockets) NAT and Firewall traversal support
full-duplex mode	--full-duplex		●	bidirectional simultaneous (same socket)
dualtest mode	-d		●	bidirectional simultaneous (separate sockets)
tradeoff mode	-r		●	bidirectional alternating direction (separate sockets)
IPv4 only	-4	●		use only IPv4 interface addresses
IPv6 only	-6	●		use only IPv6 interface addresses
MSS size	-M	●	●	set TCP MSS payload size (default 1448 bytes)
TCP slow start	-O	●		allow TCP slow start to finish before collecting results
Multicast	-u, -B, -T		●	UDP only, -B multicast group address, -T TTL hops
disk read/write	-F *filename*	●		client-side: read file from disk and write to the network server-side: write file from network data to server disk
read/write buffer size	-l	●	●	maximum read/write buffer size vary with iPerf version configured separately on client and server for iPerf2
verbose report	-V	●		mss, buffer size, cpu usage, tcp congestion algorithm
enhanced report	-e		●	buffer size, cwnd, retries, connect time, nagle, netpwr
log test results	--logfile *filename*	●		send test output to a log file instead of console
server-side results	--get-server-output	●		display server-side test results on client console

iPerf3 Test Examples

TCP Throughput

The purpose of this test is to measure TCP throughput and retries (retransmits) from client to server with MSS 1460 bytes. TCP throughput measures average bit rate and maximum data transfer for a single session by default. Throughput testing is limited by the client network interface. You cannot test 10 Gigabit Ethernet links with a client that only has a Gigabit interface.

TCP control channel does PMTUD to verify lowest MTU on forwarding path. TCP channel detects the default MTU 1500 bytes on forwarding path and assigns MSS 1448 bytes. There are 12 bytes deducted from MSS 1460 since TCP timestamps option is enabled. UDP derives the same default packet size value from TCP PMTUD/MTU discovery without fragmentation. The data channel assigns MSS 1460 with -M 1460 parameter setting for this test. You could also omit -M parameter and iPerf would automatically detect path MTU and include with report for troubleshooting purposes.

The other settings are to allow TCP slow start (-O 2) to finish before collecting results and enable verbose mode (-V) for a detailed report. The advantage of TCP slow start option is more accurate results for data throughput by removing initial protocol overhead. TCP slow start command omits the first two seconds of testing. TCP will use all bandwidth available by default with a report interval (-i) of 1 second. Consider a longer test duration (-t) of 5 minutes (300 seconds) instead of 10 seconds (default) when troubleshooting. Start the server first to listen for client connections on default port TCP 5201. Server mode is stopped with Ctrl + C key to prevent unauthorized testing.

Start iPerf3 server mode

```
# iperf3 -s
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -M 1460 -O 2 -V
```

TCP Reverse Mode Throughput (-R)

TCP reverse mode throughput test measures average bit rate and maximum data transfer from server to client for a single session. Consider that network latency is not necessarily symmetrical and reverse path could account for significant latency that causes lower aggregate throughput and retransmits.

Reverse mode is also selected when you have packets traversing NAT or firewall and would like to test in both directions. This is permitted since the client initiates a same socket connection to server instead of the server starting a new session. Configure reverse mode (-R) and display server results on client console. Maintain the same TCP slow start, verbose mode, test duration, and MSS 1460 bytes for best results.

Start iPerf3 server mode

```
# iperf3 -s
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -t 20 -R -M 1460 -O 2 -V --get-server-output
```

TCP Parallel Streams Throughput (-P)

The purpose of this test is to measure TCP full bandwidth saturation between client and server. It is often difficult to measure the maximum bandwidth available with a single stream since TCP window size limits throughput. TCP parallel streams simulates multiple TCP application sessions with separate TCP windows. This is used to measure average bit rate and maximum throughput with concurrent connections. CPU multi-threading is supported as well so that CPU is not a testing bottleneck.

Other performance attributes include retransmits and TCP congestion window (cwnd) that also indicate link capacity. Configure this test with 10 parallel streams, TCP window size of 230 KB, and test duration of 20 seconds. Notice that aggregate link bandwidth is distributed across all TCP sessions. You can also modify the number of parallel streams to identify when maximum throughput occurs.

Start iPerf3 server mode

```
# iperf3 –s
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -M 1460 -t 20 -P 10 –w 230K -O 2 –V
```

Line Quality (voice and video)

Line quality test measures throughput, packet loss, and jitter for voice and video delay-sensitive applications. They are UDP-based applications with some tolerance for packet loss. UDP protocol is preferred for voice and video since it provides lower latency than TCP. The recommended maximum values are 1% packet loss, 30 ms jitter, and 150 ms one-way latency. This is the equivalent of ping RTT latency of 300 ms. You could also test one-way latency in both directions with MTR. This would identify hidden latency in the return path that exceeds 150 ms as well.

iPerf testing should consider the difference between voice and video bandwidth usage. The maximum bidirectional bandwidth required per voice call is 200 Kbps or 100 Kbps in each direction. You could increase bandwidth setting to 2M for example to test packet loss and jitter for 10 simultaneous voice calls. There is also video streaming that requires approximately 5 Mbps per HD stream or 50 Mbps for 10 concurrent streams. The other distinction between voice and video is UDP packet size (-l). Configure maximum packet size of 200 bytes for voice calls and 1472 bytes for video traffic. Test parameters include UDP (-u) protocol, packet size (-l), report interval (-i), test duration (-t), and maximum bandwidth (-b). There is also a log file used to save test results and display with Linux **cat** command.
Start iPerf3 server mode

```
# iperf3 -s
```

Start iPerf3 client mode (1 voice call)

```
# iperf3 -c 192.168.122.3 -u -i 10 -t 300 -l 200 -b 200K -V
```

Start iPerf3 client mode (10 voice calls)

```
# iperf3 -c 192.168.122.3 -u -i 10 -t 300 -l 200 -b 2M -V --logfile 10-voice-calls
# cat 10-voice-calls
```

Start iPerf3 client mode (1 video stream)

```
# iperf3 -c 192.168.122.3 -u -i 10 -t 300 -l 1472 -b 5M -V
```

Start iPerf3 client mode (10 video streams)

```
# iperf3 -c 192.168.122.3 -u -i 10 -t 300 -l 1472 -b 50M -V --logfile video-10
# cat video-10
```

Link Saturation Throughput

This test will enable unlimited bandwidth to measure LAN/WAN maximum throughput. Testing is based on UDP protocol since TCP window is reduced when network congestion and packet loss occurs. This test is often used to detect rate limiting and firewall bottlenecks for example. For best results, run saturation tests with a client and server that have 10 GE network interfaces to remove bottlenecks.

The bit rate report will approximate near theoretical speed since UDP sends packets as fast as possible and does not retransmit packets. This test will configure UDP to use unlimited bandwidth (-b 0) with no constraints instead of setting a maximum bandwidth ceiling. This is recommended for broadband tests as well where speed fluctuates. There is some packet loss that is evident with throughput (data transfer) that increases with bit rate and fragmentation of larger packet size.

WAN test is assigned UDP maximum packet size (-l) 1472 bytes without fragmentation. LAN test is assigned 65507 bytes maximum packet size for higher data center throughput. UDP is based on a datagram that support up to 65507 bytes of data payload. The zerocopy (-Z) client parameter uses less CPU and recommended for older hardware and high throughput testing.

Start iPerf3 server mode

```
#iperf3 –s
```

Start iPerf3 client mode WAN test

```
# iperf3 -c 192.168.122.3 -u -i 2 -t 30 -b 0 -l 1472 -Z -V
```

Start iPerf3 client mode LAN test

```
# iperf3 -c 192.168.122.3 -u -i 2 -t 30 -b 0 -l 65507 -Z -V
```

Network Stress Testing

The purpose of stress testing is to verify network resiliency when your traffic exceeds theoretical capacity. Stress testing assigns bandwidth several times greater than theoretical speed. This network congestion will overflow memory buffers, interface queues, and increase device processing delay. This will also test the effectiveness of your QoS configuration. This example is based on a theoretical bandwidth link of 100 Mbps. For best results, run test with a client and server that have 10 GE network interfaces to remove bottlenecks.

The first client mode command is for TCP stress testing with a test duration of 60 seconds. Maximum bandwidth test parameter is 3 times theoretical speed (300 Mbps). It is worthwhile running this test with TCP to verify window scaling and retransmits. The second client mode command is UDP-based stress testing where packet size is 2000 bytes. This triggers packet fragmentation and will add processing delay to devices.

Start iPerf3 server mode

```
# iperf3 -s
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -i 2 -t 60 -M 1460 -O 2 -b 300M -Z -V
# iperf3 -c 192.168.122.3 -u -i 2 -t 60 -b 300M -l 2000 -V
```

Disk Read/Write Throughput

The purpose of this test is to measure disk read/write throughput and compare with network throughput. There is network latency that affects throughput and also server processing latency measured with disk read/write tests. This test will identify whether disk access or the network is a bottleneck. The advantage for network engineers is eliminating the network as the source of performance problems. Select a large file (ISO, tar etc.) in the same client directory as iPerf3 for the disk read test. The server write test creates a file from inbound network traffic and writes it to disk. Compare network throughput with disk read/write tests to identify any bottlenecks.

Start iPerf3 server mode

```
# iperf3 -s
```

This is a standard memory-to-memory network throughput test between client and server.

```
# iperf3 -c 192.168.122.3 -O 2 -V
```

This test will read a file from client disk to the network with a test duration of 20 seconds.

```
# iperf3 -c 192.168.122.3 -F iperf-3.17.1.tar.gz -O 2 -t 20 -V
```

This test will write a file from the network traffic to server disk with a test duration of 40 seconds

```
# Ctrl + C (stop iPerf3 server)
# iperf3 -s -F write-test
# iperf3 -c 192.168.122.3 -O 2 -t 40 -V --get-server-output
```

Non-Default Server Listening Port

The default server listening port for iPerf3 is TCP 5201. This requires an open port request on a network or host-based firewall if it is in the forwarding path. There is also UDP 5201 required for any UDP testing. You can assign a common port such as TCP 443 or UDP 53 instead if there are issues with firewall administration.

Start iPerf3 server mode

```
# iperf3 -s -p 443
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -M 1460 -p 443
```

Jumbo Frames MTU

This test will measure TCP throughput with Ethernet jumbo frames that have MTU 9000 bytes. Jumbo frames enable MSS 8960 bytes after IP header (20 bytes) and TCP header (20 bytes) deducted. That increases payload by a factor of 6 times compared with MSS 1460. All devices must be assigned MTU 9000 to prevent fragmentation..

Configure MTU 9000 bytes on server Ethernet interface

```
# sudo ip link set dev ens4 mtu 9000
```

Configure MTU 9000 bytes on client Ethernet interface

```
# sudo ip link set dev ens4 mtu 9000
```

Start iPerf3 server mode

```
# iperf3 –s
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -M 8960 -O 2 –V
```

What-if Latency Simulation

The purpose of this test is to measure the affect of higher latency on throughput. You could for example have a network link with 10 ms average latency most of the time. How would higher or lower latency affect throughput and retransmits for web-based applications? Linux includes a simulation tool called Netem that allows testing based on different values for packet loss, latency, and jitter. Start with TCP throughput baseline testing for current maximum data throughput and then assign 100 ms additional latency to the client Ethernet interface.

Start iPerf3 server mode

```
# iperf3 –s
```

Start iPerf3 client mode for baseline testing

```
# iperf3 -c 192.168.122.3 -t 20 -O 2 -M 1460 –V
# ping 192.168.122.3
```

Add 100 ms latency to client Ethernet interface

```
# sudo tc qdisc add dev ens4 root netem delay 100ms
```

Start iPerf3 client mode

```
# iperf3 -c 192.168.122.3 -t 20 -O 2 -M 1460 –V
# ping 192.168.122.3
```

Remove 100 ms latency from client Ethernet interface

```
# sudo tc qdisc delete dev ens4 root netem
```

iPerf2 Test Examples

TCP Throughput

TCP throughput test measures average bit rate and maximum data transfer for a single session. Configure MSS size 1460 and enable enhanced reporting (-e). iPerf2 has a reporting interval (-i) of zero (0) by default. This test has a reporting interval of 1 second that must be configured on server as well since some test parameters are not sent to server.

Start iPerf2 server mode

```
# iperf –s -i 1
```

Start iPerf2 client mode

```
# iperf -c 192.168.122.5 -i 1 -M 1460 -e
```

Line Quality (voice and video)

This test is similar to the line quality test with iPerf3 for voice and video traffic. Test parameters such as UDP (-u), report interval (-i), and packet size (-) are configured on the server as well. Test line quality at different times of the day to compare packet loss and jitter results. Network latency is tested with MTR or ping between client and server.

Start iPerf2 server mode (voice)

```
# iperf -s -u -i 10 -l 200
```

Start iPerf2 client mode (1 voice call)

```
# iperf -c 192.168.122.5 -u -i 10 -t 300 -l 200 -b 200K -V
```

Start iPerf2 client mode (10 voice calls)

```
# iperf -c 192.168.122.5 -u -i 10 -t 300 -l 200 -b 2M -V --logfile 10-voice-calls
# cat 10-voice-calls
```

Start iPerf2 server mode (video)

```
# iperf -s -u -i 10 -l 1472
```

Start iPerf2 client mode (single video)

```
# iperf -c 192.168.122.5 -u -i 10 -t 300 -l 1472 -b 5M -V
```

Start iPerf2 client mode (10 video streams)

```
# iperf -c 192.168.122.5 -u -i 10 -t 300 -l 1472 -b 50M -V --logfile video-10
# cat video-10
```

Network Stress Testing

This test is similar to iPerf3 network stress testing where test bandwidth exceeds theoretical capacity for TCP and UDP. Testing is based on a theoretical link of 100 Mbps and stress testing at 300 Mbps. Test parameters UDP (-u), report interval (-i), and packet size (-l) are configured on the server as well since there is no control channel. The packet size (-l) of 2000 bytes will trigger fragmentation to increase processing delay on network devices.

Start iPerf2 server mode

```
# iperf -s -u -i 2 -l 2000
```

Start iPerf2 client mode

```
# iperf -c 192.168.122.3 -i 2 -t 60 -M 1460 -O 2 -b 300M –V
# iperf -c 192.168.122.5 -u -i 2 -t 60 -b 300M -l 2000 -e
```

Multicast Throughput

Multicast throughput testing is currently only available with iPerf2. This test example sends UDP traffic to multicast group 239.1.1.1 address from client. The bind (-B) option explicitly assigns multicast group 239.1.1.1 to the server. That is the IP address that client will send UDP traffic for testing. TTL (-T) option is used to set maximum number of network hops between client and server. The test duration is 5 seconds and reporting interval is 1 second.

Start iPerf2 server mode

```
# iperf -s -u -B 239.1.1.1 -i 1
```

Start iPerf2 client mode

```
# iperf -c 239.1.1.1 -u -T 10 -t 5 -i 1
```

TCP Connect Time

This test measures TCP handshake connection latency at the start of a session and sends no data. TCP connect time testing is only available with iPerf2. Enhanced reporting is also included with –e option.

Start iPerf2 server mode

```
# iperf -s -i 1
```

Start iPerf2 client mode

```
# iperf -c 192.168.122.5 -i 1 --connect-only –e
```

Troubleshooting Tips and Tricks

iPerf is a network testing and troubleshooting tool that generates reports based on test parameters. The results are symptoms that point to possible root cause of slow performance or internet connectivity for example. There are also other troubleshooting tools such as MTR and tcpdump that help connect the dots and identify root cause issues. It is worthwhile to document your troubleshooting results and root cause for reference considering you could spend hours or days on a single issue. This is valuable since it reveals where problems exist within your network and avoids similar errors. iPerf reports vary based on whether you are testing with TCP or UDP protocol.

- Throughput (bytes)
- Bandwidth (bit rate)
- TCP retransmits
- TCP congestion window
- TCP MSS size
- UDP packet loss
- UDP jitter
- CPU usage

TCP-based applications account for most network traffic whether across the internet or within a data center. iPerf reports on a variety of performance metrics that include data throughput, speed, retransmits, congestion window, and other TCP attributes. The root cause is often at a lower layer that TCP responds to with reduced TCP congestion window, retransmits and lower throughput. There are also other root causes such as ISP rate-limiting, QoS errors, window scaling, and device bottlenecks. The calculation for actual throughput is some percentage of theoretical value. The fundamental cause of lower throughput is network latency that reduces TCP window size (RWND). Any packet loss that results from network congestion or interface errors will increase latency as well.

Throughput calculation provides some context to what is reported with iPerf when comparing theoretical and real performance. The results compare theoretical and real throughput for a Gigabit interface with different TCP window size. Real throughput with the default maximum TCP window size of 64 KB is only 50% of theoretical value. This would also require line conditions with negligible packet loss so that TCP congestion control is not triggered.

TCP window size of 125 KB with window scaling enabled would enable 100% maximum throughput. This however does NOT represent actual data throughput since headers will reduce goodput. iPerf will report a maximum of 960 Mbps throughput for a Gigabit link with window scaling. This is an important distinction between iPerf that reports data throughput and Speedtest. Throughput will also increase as latency is reduced for the same TCP window size.

theoretical throughput (1000 Mbps) = 1,000,000,000 bits / 8 bits = 125 Mbytes (LAN)

RWND (bytes) / RTT (sec) = 65535 bytes /.001 sec = 65,535,000 bytes (65.5 Mbytes)

65,535,000 bytes x 8 bits = 524,280,000 bps = 525 Mbps (50%)

RWND (bytes) / RTT (sec) = 125000 bytes /.001 sec = 125,000,000 bytes (125 Mbytes)

125,000,000 bytes x 8 bits = 1,000,000 bps = 1000 Mbps (100%)

theoretical throughput (100 Mbps) = 100,000,000 / 8 bits = 12.5 Mbytes (WAN)

RWND (bytes) / RTT (sec) = 65535 bytes /.020 sec = 3,276,750 bytes (3.27 Mbytes)

3,276,750 bytes x 8 bits = 26,214,000 bps = 26 Mbps (25%)

RWND (bytes) / RTT (sec) = 125000 bytes /.020 sec = 6,250,000 bytes (6.25 Mbytes)

6,250,000 bytes x 8 bits = 50,000,000 bps = 50 Mbps (50%)

iPerf reports on bandwidth (bit rate) and convert that to data throughput for the interval (-i) selected. There are also number of retries (retransmits) for throughput per time interval such as one second. The recommended number of retries for a normal network connection is maximum of 1%. This refers to packet drops that TCP has retransmitted since it is connection-oriented with guaranteed delivery. iPerf reports provide different results based on whether you are testing with TCP or UDP.

Root Cause	Symptom	TCP	UDP
faulty cablling, transceiver connector, nic	FCS/CRC interface errors	packet loss ↓ TCP retransmits ↓ higher latency ↓ lower throughput	packet loss ↓ lower throughput
duplex mismatch	late collisions, CRC, runts interface errors		
speed mismatch	queue drops		
traffic bursts	queue drops		
wireless interference	queue drops		
ISP rate-limiting	queue drops		
QoS misconfiguration	queue drops		
MTU mismatch (fragmentation)	lower throughput		