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 detailed 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

- TCP MSS (-M) = 1460 bytes - 12 byte TCP timestamp option = 1448 bytes.
- TCP auto-tuning (default) recommended since TCP adapts to network conditions.
- 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.
- UDP (-u), socket buffer size (-w), and read/write buffer size (-l) sent to server via control channel (iPerf3).
- Bind interface (-B) is the outbound interface on client and inbound interface on server.
- Configure Cisco and/or host firewall to allow TCP 5201/UDP 5201 (iPerf3) and TCP 5001 (iPerf2).
- Custom ports (-p 443) avoid firewall rule requests since they are often permitted by default.

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 (bitrate) and retransmisions from client to server with MSS 1448 bytes. TCP throughput measures average bitrate 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 to prevent 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 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 192.168.216.100 –t 30 -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, and test duration.

Start iPerf3 server mode

iperf3 -s

Start iPerf3 client mode

iperf3 -c 192.168.216.100 -t 30 -R -O 2 -V

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. This is relevant mostly for WAN links where BDP is higher than LAN. TCP parallel streams simulates multiple TCP application sessions with separate TCP windows. This is used to measure average bitrate 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 retransmissions and TCP congestion window (cwnd) that indicate link capacity.You can modify the number of parallel streams to identify when maximum throughput occurs.

Start iPerf3 server mode

iperf3 –s

Start iPerf3 client mode

iperf3 -c 192.168.216.100 -t 20 -P 8 -O 2 -V

Link Quality (voice and video)

Link quality test measures packet loss and jitter for voice and video delay-sensitive real-time 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, 30ms jitter, and 150ms one-way latency. This is the equivalent to 300ms ping RTT latency. You could also test one-way latency in both directions with MTR. This would identify hidden latency in the return path that exceeds 150ms as well.

iperf3 –s -p 161

Start iPerf3 client mode (10 G.711 voice calls)

iperf3 -c 192.168.216.100 -u -b 900K -t 30 -l 160 -V -p 161 --logfile 10-voice-calls.txt

Start iPerf3 server mode

iperf3 –s -p 443

Start iPerf3 client mode (10 TCP video streams)

iperf3 -c 192.168.216.100 -b 5M -R -P 10 -l 1200 -t 60 -i 2 -O 2 -V -p 443

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 161 instead if there are issues with firewall administration.

Start iPerf3 server mode

iperf3 -s -p 443

Start iPerf3 client mode

iperf3 -c 192.168.216.100 -t 20 -O 2 -V -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 ens2 mtu 9000

Configure MTU 9000 bytes on client Ethernet interface

sudo ip link set dev ens2 mtu 9000

Start iPerf3 server mode

iperf3 -s

Start iPerf3 client mode

iperf3 -c 192.168.216.100 -t 20 -M 8960 -O 2 -V

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. It is worthwhile to document your troubleshooting results and root cause for reference. This is valuable since it reveals where problems exist within your network and avoids similar errors.

The root cause of performance issues is often at a lower layer that TCP responds to with reduced TCP congestion window, retransmissions and lower throughput. There are other root causes such as ISP rate-limiting, QoS errors, window scaling, and device bottlenecks. The fundamental cause of lower throughput is network latency that reduces TCP window size. Any packet loss that results from network congestion or interface errors will also increase latency. iPerf will report a maximum of 960 Mbps throughput for a Gigabit link since headers are not included.

<table>
<tr><th>Root Cause</th><th>Symptom</th><th>TCP</th><th>UDP</th></tr>
<tr><td>faulty cablling, transceiver connector, nic</td><td>FCS/CRC interface errors</td><td rowspan="7">packet loss
↓
TCP retransmissions
↓
higher latency
↓
lower throughput</td><td rowspan="7">packet loss
↓
lower throughput</td></tr>
<tr><td>duplex mismatch</td><td>late collisions, CRC, runts interface errors</td></tr>
<tr><td>speed mismatch</td><td>queue drops</td></tr>
<tr><td>traffic bursts</td><td>queue drops</td></tr>
<tr><td>wireless interference</td><td>queue drops</td></tr>
<tr><td>ISP rate-limiting</td><td>queue drops</td></tr>
<tr><td>QoS misconfiguration</td><td>queue drops</td></tr>
<tr><td>MTU mismatch (fragmentation)</td><td colspan="3">lower throughput</td></tr>
</table>