iPerf3 User Documentation

General Options

Command Line Option	Description
-p, --port *n*	The server port for the server to listen on and the client to connect to. This should be the same in both client and server. Default is 5201.
--cport *n*	Option to specify the client-side port. (new in iPerf 3.1)

-f, --format *[kmKM]*	A letter specifying the format to print bandwidth numbers in. Supported formats are
	`k' = Kbits/sec` `'K' = KBytes/sec`
	`m' = Mbits/sec` `'M' = MBytes/sec`
	The adaptive formats choose between kilo- and mega- as appropriate.

-i, --interval *n*	Sets the interval time in seconds between periodic bandwidth, jitter, and loss reports. If non-zero, a report is made every ***interval*** seconds of the bandwidth since the last report. If zero, no periodic reports are printed. Default is zero.
-F, --file name	**client-side:** read from the file and write to the network, instead of using random data; **server-side:** read from the network and write to the file, instead of throwing the data away.
-A, --affinity *n/n,m-F*	Set the CPU affinity, if possible (Linux and FreeBSD only). On both the client and server you can set the local affinity by using the n form of this argument (where n is a CPU number). In addition, on the client side you can override the server's affinity for just that one test, using the n,m form of argument. Note that when using this feature, a process will only be bound to a single CPU (as opposed to a set containing potentialy multiple CPUs).
-B, --bind *host*	Bind to ***host*** , one of this machine's addresses. For the client this sets the outbound interface. For a server this sets the incoming interface. This is only useful on multihomed hosts, which have multiple network interfaces.
-V, --verbose	give more detailed output
-J, --json	output in JSON format
--logfile file	send output to a log file. (new in iPerf 3.1)
--d, --debug	emit debugging output. Primarily (perhaps exclusively) of use to developers.
-v, --version	Show version information and quit.
-h, --help	Show a help synopsis and quit.

Server Specific Options

Command Line Option	Description
-s, --server	Run iPerf in server mode. (This will only allow one iperf connection at a time)
-D, --daemon	Run the server in background as a daemon.
-I, --pidfile*file*	write a file with the process ID, most useful when running as a daemon.

Client Specific Options

Command Line Option	Description
-c, --client ***host***	Run iPerf in client mode, connecting to an iPerf server running on ***host***.
--sctp	Use SCTP rather than TCP (Linux, FreeBSD and Solaris). (new in iPerf 3.1)
-u, --udp	Use UDP rather than TCP. See also the -b option.
-b, --bandwidth ***n[KM]***	Set target bandwidth to n bits/sec (default 1 Mbit/sec for UDP, unlimited for TCP). If there are multiple streams (-P flag), the bandwidth limit is applied separately to each stream. You can also add a '/' and a number to the bandwidth specifier. This is called "burst mode". It will send the given number of packets without pausing, even if that temporarily exceeds the specified bandwidth limit.
-t, --time ***n***	The time in seconds to transmit for. iPerf normally works by repeatedly sending an array of ***len*** bytes for ***time*** seconds. Default is 10 seconds. See also the -l, -k and -n options.
-n, --num ***n[KM]***	The number of buffers to transmit. Normally, iPerf sends for 10 seconds. The -n option overrides this and sends an array of ***len*** bytes ***num*** times, no matter how long that takes. See also the -l, -k and -t options.
-k, --blockcount ***n[KM]***	The number of blocks (packets) to transmit. (instead of -t or -n) See also the -t, -l and -n options.
-l, --length ***n[KM]***	The length of buffers to read or write. iPerf works by writing an array of ***len*** bytes a number of times. Default is 128 KB for TCP, 8 KB for UDP. See also the -n, -k and -t options.
-P, --parallel ***n***	The number of simultaneous connections to make to the server. Default is 1.
-R, --reverse	Run in reverse mode (server sends, client receives).
-w, --window ***n[KM]***	Sets the socket buffer sizes to the specified value. For TCP, this sets the TCP window size. (this gets sent to the server and used on that side too)
-M, --set-mss ***n***	Attempt to set the TCP maximum segment size (MSS). The MSS is usually the MTU - 40 bytes for the TCP/IP header. For ethernet, the MSS is 1460 bytes (1500 byte MTU).
-N, --no-delay	Set the TCP no delay option, disabling Nagle's algorithm. Normally this is only disabled for interactive applications like telnet.
-4, --version4	only use IPv4.
-6, --version4	only use IPv6.

-S, --tos ***n***	The type-of-service for outgoing packets. (Many routers ignore the TOS field.) You may specify the value in hex with a '0x' prefix, in octal with a '0' prefix, or in decimal. For example, '0x10' hex = '020' octal = '16' decimal. The TOS numbers specified in RFC 1349 are:
	`IPTOS_LOWDELAY     minimize delay        0x10`
	`IPTOS_THROUGHPUT   maximize throughput   0x08`
	`IPTOS_RELIABILITY  maximize reliability  0x04`
	`IPTOS_LOWCOST      minimize cost         0x02`

-O, --omit ***n***	Omit the first n seconds of the test, to skip past the TCP TCP slowstart period.