



## iPerf3 User Documentation

Support Team - 2021-10-21 - Comments (0) - General

### iPerf 3 user documentation

#### GENERAL OPTIONS

Command line option	Description
<b>-p, --port <i>n</i></b>	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.
<b>--cport<i>n</i></b>	Option to specify the client-side port. (new in iPerf 3.1)
<b>-f, --format[<i>kmKM</i>]</b>	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.
<b>-i, --interval <i>n</i></b>	Sets the interval time in seconds between periodic bandwidth, jitter, and loss reports. If non-zero, a report is made every <b>interval</b> seconds of the bandwidth since the last report. If zero, no periodic reports are printed. Default is zero.
<b>-F, --file name</b>	<b>client-side:</b> read from the file and write to the network, instead of using random data; <b>server-side:</b> read from the network and write to the file, instead of throwing the data away.
<b>-A, --affinity<i>n/n,m-F</i></b>	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 <i>n</i> form of this argument (where <i>n</i> is a CPU number). In addition, on the client side you can override the server's affinity for just that one test, using the <i>n,m</i> 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 potentially multiple CPUs).
<b>-B, --bind <i>host</i></b>	Bind to <b>host</b> , 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.
<b>-V, --verbose</b>	give more detailed output

<b>-J, --json</b>	output in JSON format
<b>--logfile file</b>	send output to a log file. (new in iPerf 3.1)
<b>--d, --debug</b>	emit debugging output. Primarily (perhaps exclusively) of use to developers.
<b>-v, --version</b>	Show version information and quit.
<b>-h, --help</b>	Show a help synopsis and quit.

## SERVER SPECIFIC OPTIONS

Command line option	Description
<b>-s, --server</b>	Run iPerf in server mode. (This will only allow one iperf connection at a time)
<b>-D, --daemon</b>	Run the server in background as a daemon.
<b>-I, --pidfile <i>file</i></b>	write a file with the process ID, most useful when running as a daemon. (new in iPerf 3.1)

## CLIENT SPECIFIC OPTIONS

Command line option	Description
<b>-c, --client <i>host</i></b>	Run iPerf in client mode, connecting to an iPerf server running on <i>host</i> .
<b>--sctp</b>	Use SCTP rather than TCP (Linux, FreeBSD and Solaris). (new in iPerf 3.1)
<b>-u, --udp</b>	Use UDP rather than TCP. See also the <a href="#">-b</a> option.
<b>-b, --bandwidth <i>n</i>[<i>KM</i>]</b>	Set target bandwidth to <i>n</i> 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.
<b>-t, --time <i>n</i></b>	The time in seconds to transmit for. iPerf normally works by repeatedly sending an array of <i>len</i> bytes for <i>time</i> seconds. Default is 10 seconds. See also the <a href="#">-l</a> , <a href="#">-k</a> and <a href="#">-n</a> options.

<b>-n, --num <i>n</i>[KM]</b>	The number of buffers to transmit. Normally, iPerf sends for 10 seconds. The -n option overrides this and sends an array of <b>len</b> bytes <b>num</b> times, no matter how long that takes. See also the <a href="#">-l</a> , <a href="#">-k</a> and <a href="#">-t</a> options.
<b>-k, --blockcount <i>n</i>[KM]</b>	The number of blocks (packets) to transmit. (instead of -t or -n) See also the <a href="#">-t</a> , <a href="#">-l</a> and <a href="#">-n</a> options.
<b>-l, --length <i>n</i>[KM]</b>	The length of buffers to read or write. iPerf works by writing an array of <b>len</b> bytes a number of times. Default is 128 KB for TCP, 8 KB for UDP. See also the <a href="#">-n</a> , <a href="#">-k</a> and <a href="#">-t</a> options.
<b>-P, --parallel <i>n</i></b>	The number of simultaneous connections to make to the server. Default is 1.
<b>-R, --reverse</b>	Run in reverse mode (server sends, client receives).
<b>-w, --window <i>n</i>[KM]</b>	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)
<b>-M, --set-mss <i>n</i></b>	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).
<b>-N, --no-delay</b>	Set the TCP no delay option, disabling Nagle's algorithm. Normally this is only disabled for interactive applications like telnet.
<b>-4, --version4</b>	only use IPv4.
<b>-6, --version6</b>	only use IPv6.
<b>-S, --tos <i>n</i></b>	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
<b>-L, --flowlabel <i>n</i></b>	Set the IPv6 flow label (currently only supported on Linux).
<b>-Z, --zerocopy</b>	Use a "zero copy" method of sending data, such as sendfile(2), instead of the usual write(2). This uses much less CPU.
<b>-O, --omit <i>n</i></b>	Omit the first <i>n</i> seconds of the test, to skip past the TCP <a href="#">TCP slowstart</a> period.
<b>-T, --title <i>str</i></b>	Prefix every output line with this string.

**-C, --linux-congestion**      Set the [congestion control algorithm](#) (Linux only for iPerf  
**algo**                              3.0, Linux and FreeBSD for iPerf 3.1).

See also <https://github.com/esnet/iperf>