IPERF GENERAL OPTIONS	
Command Line Option	**Description**
-p, --port ***n***	The server port for the server to listen on and the client to connect to.
-f, --format ***[kmKM]***	A letter specifying the format to print bandwidth numbers in. Supported formats: 'k' = Kbits/sec 'K' = KBytes/sec 'm' = Mbits/sec 'M' = MBytes/sec
-i, --interval ***n***	Sets the interval time in seconds between periodic bandwidth, jitter, and loss reports. If zero, no periodic reports are printed. Default is zero.
-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.
-v, --version	Show version information and quit.
IPERF SERVER SPECIFIC OPTIONS	
-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.
IPERF CLIENT SPECIFIC OPTIONS	
-c, --client ***host***	Run iPerf in client mode, connecting to an iPerf server running on ***host***.
-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.
-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.
-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.
DIG MOST USED OPTIONS	
-p *n*	*n* is the port number that dig will send its queries instead of the standard DNS port number 53. This option would be used to test a name server that has been configured to listen for queries on a non-standard port number.
-x	Perform a reverse lookup.
-t *type*	Sets the query type to *type*. The default query type is "A", unless the **-x** option is supplied to indicate a reverse lookup.
@IP/name	Specify the name or IP address of the name server to query. This can be an IPv4 address in dotted-decimal notation or an IPv6 address in colon-delimited notation.

PING MOST USED OPTIONS	
-b	Allow pinging a broadcast address.
-c *count*	Stop after sending count ECHO_REQUEST packets. With deadline option, ping waits for count ECHO_REPLY packets, until the timeout expires.
-f	Flood ping. For every ECHO_REQUEST sent a period "." is printed, while for ever ECHO_REPLY received a backspace is printed. Only the super-user may use this option with zero interval.
-i *interval*	Wait interval seconds between sending each packet. The default is to wait for one second between each packet normally, or not to wait in flood mode. Only super-user may set interval to values less 0.2 seconds.
-Q *tos*	Set Quality of Service -related bits in ICMP datagrams. tos can be either decimal or hex number.
-s *packetsize*	Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.
-t *ttl*	Set the IP Time to Live.
-W *timeout*	Time to wait for a response, in seconds. The option affects only timeout in absense of any responses, otherwise ping waits for two RTTs.
TRACEROUTE MOST USED OPTIONS	
-T	Use TCP SYN for probes
-I	Use ICMP ECHO for probes
-F	Do not fragment probe packets.
-n	Do not try to map IP addresses to host names when displaying them.
-p *n*	For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.
-t *tos*	For IPv4, set the Type of Service (TOS) and Precedence value. Useful values are 16 (low delay) and 8 (high throughput). Note that in order to use some TOS precendence values, you have to be super user. For IPv6, set the Traffic Control value.
-w *waittime*	Set the time (in seconds) to wait for a response to a probe (default 5.0 sec).
-q *nqueries*	Sets the number of probe packets per hop. The default is 3.
-A	Perform AS path lookups in routing registries and print results directly after the corresponding addresses.
-M *method*	Use specified method for traceroute operations. Default traditional udp method has name default, icmp (-I) and tcp (-T) have names icmp and tcp respectively. Method-specific options can be passed by -O . Most methods have their simple shortcuts, (-I means -M icmp, etc).
--mtu	Discover MTU along the path being traced. Implies -F -N 1. New mtu is printed once in a form of F=NUM at the first probe of a hop which requires such mtu to be reached.
CURL MOST USED OPTIONS	
-k, --insecure	By default, every SSL connection curl makes is verified to be secure. This option allows curl to proceed and operate even for server connections otherwise considered insecure.
-L, --location	(HTTP) If the server reports that the requested page has moved to a different location (indicated with a Location: header and a 3XX response code), this option will make curl redo the request on the new place.