NAME

traceroute6 - print the route IPv6 packets will take to a network node

SYNOPSIS

traceroute6 [**-adIlnNrSTUv**] [**-f** *firsthop*] [**-g** *gateway*] [**-m** *hoplimit*] [**-p** *port*] [**-q** *probes*] [**-s** *src*] [**-t** *tclass*] [**-w** *waittime*] [**-A** *as_server*] *target* [*datalen*]

DESCRIPTION

The **traceroute6** utility uses the IPv6 protocol hop limit field to elicit an ICMPv6 TIME_EXCEEDED response from each gateway along the path to some host.

The only mandatory parameter is the destination host name or IPv6 address. The default probe datagram carries 20 bytes of payload, in addition to the IPv6 header. The size of the payload can be specified by giving a length (in bytes) after the destination host name.

Other options are:

-a Turn on AS# lookups for each hop encountered.

-A *as_server*
Turn on AS# lookups and use the given server instead of the default.

-d Debug mode.

-f *firsthop*
Specify how many hops to skip in trace.

-g *gateway*
Specify intermediate gateway. Please note that **traceroute6** tries to use routing headers.

-I Use ICMP6 ECHO instead of UDP datagrams.

-l Print both host hostnames and numeric addresses. Normally **traceroute6** prints only hostnames if **-n** is not specified, and only numeric addresses if **-n** is specified.

-m *hoplimit*
Specify maximum hoplimit, up to 255. The default is the value of the *net.inet6.ip6.hlim* sysctl(8) (the same default used for TCP connections).

-n Do not resolve numeric address to hostname.

-N Use a packet with no upper layer header for the probes, instead of UDP datagrams.

-p *port* Set SCTP/TCP/UDP port number to *port*.

-q *probes*
Set the number of probe per hop count to *probes*.

-r Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-connected network, an error is returned. This option corresponds to the SO_DONTROUTE socket option; it can be used to ping a local host through an interface that has no route through it (e.g., after the interface was dropped by a routing daemon).

-s *src* *Src* specifies the source IPv6 address to be used.

-S Use SCTP packets for the probes. The size of probe packets must be a multiple of 4. If *datalen* is up to 28, probe packets consist of a SHUTDOWN-ACK chunk possibly bundled with a PAD chunk. For larger probe packets, an INIT chunk is used.

-t *tclass*
tclass specifies the *traffic class* used when sending probe packets. The value must be a decimal integer in the range 0 to 255. The default is 0.

-T Use TCP segments for the probes.

-U Use UDP datagrams for the probes. This is the default.

-v Be verbose.

-w *waittime*
Specify the delay time between probes.

This program prints the route to the given destination and the round-trip time to each gateway, in the same manner as traceroute.

Here is a list of possible annotations after the round-trip time for each gateway:

!N Destination Unreachable - No Route to Host.

!P Destination Unreachable - Administratively Prohibited.

!S Destination Unreachable - Not a Neighbour.

!A Destination Unreachable - Address Unreachable.

!H Parameter Problem - Unrecognized Next Header Type.

! This is printed if the hop limit is <= 1 on a port unreachable message. This means that the packet got to the destination, but that the reply had a hop limit that was just large enough to allow it to get back to the source of the traceroute6. This was more interesting in the IPv4 case, where some IP stack bugs could be identified by this behaviour.

EXIT STATUS

The **traceroute6** utility will exit with 0 on success, and non-zero on errors.

SEE ALSO

ping(8), traceroute(8)

HISTORY

The **traceroute6** utility first appeared in WIDE hydrangea IPv6 protocol stack kit.