NAME

   ping − send ICMP ECHO_REQUEST to network hosts

SYNOPSIS

   ping [**−aAbBdDfhLnOqrRUvV46**] [**−c** *count*] [**−F** *flowlabel*] [**−i** *interval*] [**−I** *interface*] [**−l** *preload*]
       [**−m** *mark*] [**−M** *pmtudisc_option*] [**−N** *nodeinfo_option*] [**−w** *deadline*] [**−W** *timeout*] [**−p** *pattern*]
       [**−Q** *tos*] [**−s** *packetsize*] [**−S** *sndbuf*] [**−t** *ttl*] [**−T** *timestamp option*] [hop...] {destination}

DESCRIPTION

   ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP
   ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams ("pings") have an IP and ICMP
   header, followed by a struct timeval and then an arbitrary number of "pad" bytes used to fill out the packet.

   ping works with both IPv4 and IPv6. Using only one of them explicitly can be enforced by specifying **−4**
   or **−6**.

   ping can also send IPv6 Node Information Queries (RFC4620). Intermediate *hop*s may not be allowed,
   because IPv6 source routing was deprecated (RFC5095).

OPTIONS

   −4

        Use IPv4 only.

   −6

        Use IPv6 only.

   −a

        Audible ping.

   −A

        Adaptive ping. Interpacket interval adapts to round−trip time, so that effectively not more than one (or
        more, if preload is set) unanswered probe is present in the network. Minimal interval is 200msec
        unless super−user. On networks with low RTT this mode is essentially equivalent to flood mode.

   −b

        Allow pinging a broadcast address.

   −B

        Do not allow **ping** to change source address of probes. The address is bound to one selected when
        ping starts.

   −c *count*

        Stop after sending *count* ECHO_REQUEST packets. With *deadline* option, **ping** waits for *count*
        ECHO_REPLY packets, until the timeout expires.

   −d

        Set the SO_DEBUG option on the socket being used. Essentially, this socket option is not used by
        Linux kernel.

   −D

        Print timestamp (unix time + microseconds as in gettimeofday) before each line.

   −f

        Flood ping. For every ECHO_REQUEST sent a period "." is printed, while for every ECHO_REPLY
        received a backspace is printed. This provides a rapid display of how many packets are being dropped.
        If interval is not given, it sets interval to zero and outputs packets as fast as they come back or one
        hundred times per second, whichever is more. Only the super−user may use this option with zero
        interval.

   −F *flow label*

        IPv6 only. Allocate and set 20 bit flow label (in hex) on echo request packets. If value is zero, kernel
        allocates random flow label.

   −h

Show help.

–i *interval*

Wait *interval* seconds between sending each packet. Real number allowed with dot as a decimal separator (regardless locale setup). 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 than 2 ms.

–I *interface*

interface is either an address, an interface name or a VRF name. If *interface* is an address, it sets source address to specified interface address. If *interface* is an interface name, it sets source interface to specified interface. If *interface* is a VRF name, each packet is routed using the corresponding routing table; in this case, the **–I** option can be repeated to specify a source address. NOTE: For IPv6, when doing ping to a link–local scope address, link specification (by the '%'–notation in *destination*, or by this option) can be used but it is no longer required.

–l *preload*

If *preload* is specified, **ping** sends that many packets not waiting for reply. Only the super–user may select preload more than 3.

–L

Suppress loopback of multicast packets. This flag only applies if the ping destination is a multicast address.

–m *mark*

use *mark* to tag the packets going out. This is useful for variety of reasons within the kernel such as using policy routing to select specific outbound processing.

–M *pmtudisc_opt*

Select Path MTU Discovery strategy. *pmtudisc_option* may be either *do* (prohibit fragmentation, even local one), *want* (do PMTU discovery, fragment locally when packet size is large), or *dont* (do not set DF flag).

–N *nodeinfo_option*

IPv6 only. Send ICMPv6 Node Information Queries (RFC4620), instead of Echo Request. CAP_NET_RAW capability is required.

help

Show help for NI support.

name

Queries for Node Names.

ipv6

Queries for IPv6 Addresses. There are several IPv6 specific flags.

ipv6–global

Request IPv6 global–scope addresses.

ipv6–sitelocal

Request IPv6 site–local addresses.

ipv6–linklocal

Request IPv6 link–local addresses.

ipv6–all

Request IPv6 addresses on other interfaces.

ipv4

Queries for IPv4 Addresses. There is one IPv4 specific flag.

ipv4–all

Request IPv4 addresses on other interfaces.

subject–ipv6=*ipv6addr*

IPv6 subject address.

subject–ipv4=*ipv4addr*
> IPv4 subject address.

subject–name=*nodename*
> Subject name. If it contains more than one dot, fully–qualified domain name is assumed.

subject–fqdn=*nodename*
> Subject name. Fully–qualified domain name is always assumed.

–n

> Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

–O

> Report outstanding ICMP ECHO reply before sending next packet. This is useful together with the timestamp **–D** to log output to a diagnostic file and search for missing answers.

–p *pattern*
> You may specify up to 16 "pad" bytes to fill out the packet you send. This is useful for diagnosing data–dependent problems in a network. For example, **–p ff** will cause the sent packet to be filled with all ones.

–q

> Quiet output. Nothing is displayed except the summary lines at startup time and when finished.

–Q *tos*
> Set Quality of Service –related bits in ICMP datagrams. *tos* can be decimal (**ping** only) or hex number.

> In RFC2474, these fields are interpreted as 8–bit Differentiated Services (DS), consisting of: bits 0–1 (2 lowest bits) of separate data, and bits 2–7 (highest 6 bits) of Differentiated Services Codepoint (DSCP). In RFC2481 and RFC3168, bits 0–1 are used for ECN.

> Historically (RFC1349, obsoleted by RFC2474), these were interpreted as: bit 0 (lowest bit) for reserved (currently being redefined as congestion control), 1–4 for Type of Service and bits 5–7 (highest bits) for Precedence.

–r

> Bypass the normal routing tables and send directly to a host on an attached interface. If the host is not on a directly–attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it provided the option **–I** is also used.

–R

> **ping** only. Record route. Includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. Note that the IP header is only large enough for nine such routes. Many hosts ignore or discard this option.

–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.

–S *sndbuf*
> Set socket sndbuf. If not specified, it is selected to buffer not more than one packet.

–t *ttl*
> **ping** only. Set the IP Time to Live.

–T *timestamp option*
> Set special IP timestamp options. *timestamp option* may be either *tsonly* (only timestamps), *tsandaddr* (timestamps and addresses) or *tsprespec host1 [host2 [host3 [host4]]]* (timestamp prespecified hops).

–U

> Print full user–to–user latency (the old behaviour). Normally **ping** prints network round trip time, which can be different f.e. due to DNS failures.

−v

> Verbose output. Do not suppress DUP replies when pinging multicast address.

−V

> Show version and exit.

−w *deadline*

> Specify a timeout, in seconds, before **ping** exits regardless of how many packets have been sent or received. In this case **ping** does not stop after *count* packet are sent, it waits either for *deadline* expire or until *count* probes are answered or for some error notification from network.

−W *timeout*

> Time to wait for a response, in seconds. The option affects only timeout in absence of any responses, otherwise **ping** waits for two RTTs. Real number allowed with dot as a decimal separator (regardless locale setup). 0 means infinite timeout.

When using **ping** for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be "pinged". Round−trip times and packet loss statistics are computed. If duplicate packets are received, they are not included in the packet loss calculation, although the round trip time of these packets is used in calculating the minimum/average/maximum/mdev round−trip time numbers.

Population standard deviation (mdev), essentially an average of how far each ping RTT is from the mean RTT. The higher mdev is, the more variable the RTT is (over time). With a high RTT variability, you will have speed issues with bulk transfers (they will take longer than is strictly speaking necessary, as the variability will eventually cause the sender to wait for ACKs) and you will have middling to poor VoIP quality.

When the specified number of packets have been sent (and received) or if the program is terminated with a SIGINT, a brief summary is displayed. Shorter current statistics can be obtained without termination of process with signal SIGQUIT.

If **ping** does not receive any reply packets at all it will exit with code 1. If a packet *count* and *deadline* are both specified, and fewer than *count* packets are received by the time the *deadline* has arrived, it will also exit with code 1. On other error it exits with code 2. Otherwise it exits with code 0. This makes it possible to use the exit code to see if a host is alive or not.

This program is intended for use in network testing, measurement and management. Because of the load it can impose on the network, it is unwise to use **ping** during normal operations or from automated scripts.

ICMP PACKET DETAILS

An IP header without options is 20 bytes. An ICMP ECHO_REQUEST packet contains an additional 8 bytes worth of ICMP header followed by an arbitrary amount of data. When a *packetsize* is given, this indicates the size of this extra piece of data (the default is 56). Thus the amount of data received inside of an IP packet of type ICMP ECHO_REPLY will always be 8 bytes more than the requested data space (the ICMP header).

If the data space is at least of size of struct timeval **ping** uses the beginning bytes of this space to include a timestamp which it uses in the computation of round trip times. If the data space is shorter, no round trip times are given.

DUPLICATE AND DAMAGED PACKETS

ping will report duplicate and damaged packets. Duplicate packets should never occur, and seem to be caused by inappropriate link−level retransmissions. Duplicates may occur in many situations and are rarely (if ever) a good sign, although the presence of low levels of duplicates may not always be cause for alarm.

Damaged packets are obviously serious cause for alarm and often indicate broken hardware somewhere in the **ping** packet's path (in the network or in the hosts).

TRYING DIFFERENT DATA PATTERNS

The (inter)network layer should never treat packets differently depending on the data contained in the data portion. Unfortunately, data−dependent problems have been known to sneak into networks and remain

undetected for long periods of time. In many cases the particular pattern that will have problems is something that doesn't have sufficient "transitions", such as all ones or all zeros, or a pattern right at the edge, such as almost all zeros. It isn't necessarily enough to specify a data pattern of all zeros (for example) on the command line because the pattern that is of interest is at the data link level, and the relationship between what you type and what the controllers transmit can be complicated.

This means that if you have a data–dependent problem you will probably have to do a lot of testing to find it. If you are lucky, you may manage to find a file that either can't be sent across your network or that takes much longer to transfer than other similar length files. You can then examine this file for repeated patterns that you can test using the **−p** option of **ping**.

TTL DETAILS

The TTL value of an IP packet represents the maximum number of IP routers that the packet can go through before being thrown away. In current practice you can expect each router in the Internet to decrement the TTL field by exactly one.

The TCP/IP specification states that the TTL field for TCP packets should be set to 60, but many systems use smaller values (4.3 BSD uses 30, 4.2 used 15).

The maximum possible value of this field is 255, and most Unix systems set the TTL field of ICMP ECHO_REQUEST packets to 255. This is why you will find you can "ping" some hosts, but not reach them with **telnet**(1) or **ftp**(1).

In normal operation ping prints the TTL value from the packet it receives. When a remote system receives a ping packet, it can do one of three things with the TTL field in its response:

• Not change it; this is what Berkeley Unix systems did before the 4.3BSD Tahoe release. In this case the TTL value in the received packet will be 255 minus the number of routers in the round–trip path.

• Set it to 255; this is what current Berkeley Unix systems do. In this case the TTL value in the received packet will be 255 minus the number of routers in the path **from** the remote system **to** the **ping**ing host.

• Set it to some other value. Some machines use the same value for ICMP packets that they use for TCP packets, for example either 30 or 60. Others may use completely wild values.

BUGS

• Many Hosts and Gateways ignore the RECORD_ROUTE option.

• The maximum IP header length is too small for options like RECORD_ROUTE to be completely useful. There's not much that can be done about this, however.

• Flood pinging is not recommended in general, and flood pinging the broadcast address should only be done under very controlled conditions.

SEE ALSO

ip(8), **ss**(8).

HISTORY

The **ping** command appeared in 4.3BSD.

The version described here is its descendant specific to Linux.

As of version s20150815, the **ping6** binary doesn't exist anymore. It has been merged into **ping**. Creating a symlink named **ping6** pointing to **ping** will result in the same functionality as before.

SECURITY

ping requires CAP_NET_RAW capability to be executed 1) if the program is used for non–echo queries (See **−N** option), or 2) if kernel does not support non–raw ICMP sockets, or 3) if the user is not allowed to create an ICMP echo socket. The program may be used as set–uid root.

AVAILABILITY

ping is part of *iputils* package.