NAME

ping — send ICMP ECHO_REQUEST packets to network hosts

SYNOPSIS

ping [**-AaDdfnoQqRrv**][**-c** *count*][**-i** *wait*][**-l** *preload*][**-M mask** | **time**]
[**-m** *ttl*][**-P** *policy*][**-p** *pattern*][**-S** *src_addr*][**-s** *packetsize*]
[**-t** *timeout*][**-z** *tos*]*host*
ping [**-AaDdfLnoQqRrv**][**-c** *count*][**-I** *iface*][**-i** *wait*][**-l** *preload*]
[**-M mask** | **time**][**-m** *ttl*][**-P** *policy*][**-p** *pattern*][**-S** *src_addr*]
[**-s** *packetsize*][**-T** *ttl*][**-t** *timeout*][**-z** *tos*]*mcast-group*

DESCRIPTION

The **ping** utility 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. The options are as follows:

-A Audible. Output a bell (ASCII 0x07) character when no packet is received before the next packet is transmitted. To cater for round-trip times that are longer than the interval between transmissions, further missing packets cause a bell only if the maximum number of unreceived packets has increased.

-a Audible. Include a bell (ASCII 0x07) character in the output when any packet is received. This option is ignored if other format options are present.

-c *count*
Stop after sending (and receiving) *count* ECHO_RESPONSE packets. If this option is not specified, **ping** will operate until interrupted.

-D Set the Don't Fragment bit.

-d Set the SO_DEBUG option on the socket being used.

-f Flood ping. Outputs packets as fast as they come back or one hundred times per second, whichever is more. 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. Only the super-user may use this option. *This can be very hard on a network and should be used with caution.*

-I *iface*
Source multicast packets with the given interface address. This flag only applies if the ping destination is a multicast address.

-i *wait*
Wait *wait* seconds *between sending each packet*. The default is to wait for one second between each packet. The wait time may be fractional, but only the super-user may specify values less than 1 second. This option is incompatible with the **-f** option.

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

-l *preload*
If *preload* is specified, **ping** sends that many packets as fast as possible before falling into its normal mode of behavior. Only the super-user may use this option.

-M mask | **time**
Use ICMP_MASKREQ or ICMP_TSTAMP instead of ICMP_ECHO. For **mask**, print the netmask of the remote machine. Set the *net.inet.icmp.maskrepl* MIB variable to enable ICMP_MASKREPLY. For **time**, print the origination, reception and transmission timestamps.

-m *ttl*
Set the IP Time To Live for outgoing packets. If not specified, the kernel uses the value of the *net.inet.ip.ttl* MIB variable.

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

-o Exit successfully after receiving one reply packet.

-P *policy*
policy specifies IPsec policy for the ping session. For details please refer to ipsec(4) and ipsec_set_policy(3).

-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 Somewhat quiet output. Don't display ICMP error messages that are in response to our query messages. Originally, the **-v** flag was required to display such errors, but **-v** displays all ICMP error messages. On a busy machine, this output can be overbearing. Without the **-Q** flag, **ping** prints out any ICMP error messages caused by its own ECHO_REQUEST messages.

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

-R 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; the traceroute(8) command is usually better at determining the route packets take to a particular destination. If more routes come back than should, such as due to an illegal spoofed packet, ping will print the route list and then truncate it at the correct spot. Many hosts ignore or discard the RECORD_ROUTE option.

-r Bypass the normal routing tables and send directly to a host on an attached network. 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 (e.g., after the interface was dropped by routed(8)).

-S *src_addr*
Use the following IP address as the source address in outgoing packets. On hosts with more than one IP address, this option can be used to force the source address to be something other than the IP address of the interface the probe packet is sent on. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent.

-s *packetsize*
Specify 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. Only the super-user may specify values more than default.

-T *ttl*
Set the IP Time To Live for multicasted packets. This flag only applies if the ping destination is a multicast address.

-t *timeout*
Specify a timeout, in seconds, before ping exits regardless of how many packets have been received.

-v Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

-z *tos*
Use the specified type of service.

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 round-trip time statistics. 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, showing the number of packets sent and received, and the minimum, mean, maximum, and standard deviation of the round-trip times.

If **ping** receives a SIGINFO (see the **status** argument for stty(1)) signal, the current number of packets sent and received, and the minimum, mean, and maximum of the round-trip times will be written to the standard error output.

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 indicated 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 eight bytes large, **ping** uses the first eight bytes of this space to include a timestamp which it uses in the computation of round trip times. If less than eight bytes of pad are specified, no round trip times are given.

DUPLICATE AND DAMAGED PACKETS

The **ping** utility will report duplicate and damaged packets. Duplicate packets should never occur when pinging a unicast address, 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. Duplicates are expected when pinging a broadcast or multicast address, since they are not really duplicates but replies from different hosts to the same request.

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 does not have sufficient "transitions", such as all ones or all zeros, or a pattern right at the edge, such as almost all zeros. It is not 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 cannot 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 recommends setting the TTL field for IP packets to 64, but many systems use smaller values (4.3BSD uses 30, 4.2BSD 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 BSD 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 BSD 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.

RETURN VALUES

The **ping** utility returns an exit status of zero if at least one response was heard from the specified *host*; a status of two if the transmission was successful but no responses were received; or another value (from <`sysexits.h`>) if an error occurred.

SEE ALSO

`netstat`(1), `ifconfig`(8), `routed`(8), `traceroute`(8)

HISTORY

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

AUTHORS

The original **ping** utility was written by Mike Muuss while at the US Army Ballistics Research Laboratory.

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 fbod pinging the broadcast address should only be done under very controlled conditions.

The **-v** option is not worth much on busy hosts.