NAME

ping - send ICMP or ICMPv6 ECHO_REQUEST packets to network hosts

SYNOPSIS

ping [**-4AaDdfHnoQqRrv**] [**-.***chars*] [**-C** *pcp*] [**-c** *count*] [**-G** *sweepmaxsize*] [**-g** *sweepminsize*] [**-h** *sweepincrsize*] [**-i** *wait*] [**-l** *preload*] [**-M mask** | **time**] [**-m** *ttl*] [**-P** *policy*] [**-p** *pattern*] [**-S** *src_addr*] [**-s** *packetsize*] [**-t** *timeout*] [**-W** *waittime*] [**-z** *tos*] *IPv4-host*

ping [**-4AaDdfHLnoQqRrv**] [**-.***chars*] [**-C** *pcp*] [**-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*] [**-W** *waittime*] [**-z** *tos*] *IPv4-mcast-group*

ping [**-6AaDdEfHNnOoquvYyZ**] [**-.***chars*] [**-b** *bufsiz*] [**-c** *count*] [**-e** *gateway*] [**-I** *interface*] [**-i** *wait*] [**-k** *addrtype*] [**-l** *preload*] [**-m** *hoplimit*] [**-P** *policy*] [**-p** *pattern*] [**-S** *sourceaddr*] [**-s** *packetsize*] [**-t** *timeout*] [**-W** *waittime*] [*IPv6-hops ...*] *IPv6-host*

DESCRIPTION

The **ping** utility invoked with an IPv4 target (*IPv4-host* or *IPv4-mcast-group*) 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.

When invoked with an IPv6 target (*IPv6-host*), it uses the ICMPv6 protocol's mandatory ICMP6_ECHO_REQUEST datagram to elicit an ICMP6_ECHO_REPLY. ICMP6_ECHO_REQUEST datagrams have an IPv6 header and ICMPv6 header formatted as documented in RFC 2463.

When invoked with a hostname, the version to which the target is resolved first is used. In that case, the options and arguments used must be valid for the specific IP version, otherwise **ping** exits with an error. If the target is resolved to both IPv4 and IPv6, the specific IP version can be requested by **-4** or **-6** options, respectively. For backwards-compatibility, ICMPv6 can also be selected by invoking the binary as **ping6**.

Options common to both IPv4 and IPv6 targets

-.*chars*

By default, for every ECHO_REQUEST sent, a period "." is printed, while for every ECHO_REPLY received, a backspace is printed. This option takes an optional string argument listing characters that will be printed one by one in the provided order instead of the default period.

Example usage:

```
ping -.0123456789 freebsd.org
```

-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.

-C *pcp*

Add an 802.1p Ethernet Priority Code Point when sending a packet. 0..7 uses that specific PCP, -1 uses the interface default PCP (or none).

-c *count*

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

For an IPv4 target, if this option is specified in conjunction with ping sweeps, each sweep will consist of *count* packets.

-D Disable fragmentation.

-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. Implies **-.** to print a period for every ECHO_REQUEST sent and a backspace for every ECHO_REPLY received. 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.*

-H Hostname output. Try to do a reverse DNS lookup when displaying addresses. This is the opposite of the **-n** option.

-I *iface*

For an IPv4 target, *iface* is an IP address indentifying an interface from which the packets will be sent. This flag applies only if the ping target is a multicast address.

For an IPv6 target, *iface* is a name of an interface (e.g., 'em0') from which the packets will be sent. This flag applies if the ping target is a multicast address, or link-local/site-local unicast 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 *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 *ttl* For an IPv4 target, 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.

For an IPv6 target, set the IPv6 hoplimit.

-n Numeric output only. No attempt will be made to lookup symbolic names for host addresses. This is the opposite of **-H**, and it is the default behavior.

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

-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.

For IPv4, if the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent.

For IPv6, the source address must be one of the unicast addresses of the sending node, and must be numeric.

-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.

For IPv4, only the super-user may specify values more than default. This option cannot be used with ping sweeps.

For IPv6, you may need to specify **-b** as well to extend socket buffer size.

-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.

-W *waittime*

Time in milliseconds to wait for a reply for each packet sent. If a reply arrives later, the packet is not printed as replied, but considered as replied when calculating statistics.

Options only for IPv4 targets

-4 Use IPv4 regardless of how the target is resolved.

-G *sweepmaxsize*

Specify the maximum size of ICMP payload when sending sweeping pings. This option is required for ping sweeps.

-g *sweepminsize*

Specify the size of ICMP payload to start with when sending sweeping pings. The default value is 0.

-h *sweepincrsize*

Specify the number of bytes to increment the size of ICMP payload after each sweep when sending sweeping pings. The default value is 1.

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

-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 and *net.inet.icmp.maskfake* if you want to override the netmask in the response. For **time**, print the origination, reception and transmission timestamps. Set the *net.inet.icmp.tstamprepl* MIB variable to enable or disable ICMP_TSTAMPREPLY.

-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.

-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)).

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

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

IPv4-host
hostname or IPv4 address of the final destination node.

IPv4-mcast-group
IPv4 multicast address of the final destination nodes.

Options only for IPv6 targets

-6 Use IPv6 regardless of how the target is resolved.

-b *bufsiz*
Set socket buffer size.

-e *gateway*
Specifies to use *gateway* as the next hop to the destination. The gateway must be a neighbor of

the sending node.

-k *addrtype*
Generate ICMPv6 Node Information Node Addresses query, rather than echo-request. *addrtype* must be a string constructed of the following characters.

a requests unicast addresses from all of the responder's interfaces. If the character is omitted, only those addresses which belong to the interface which has the responder's address are requests.

c requests responder's IPv4-compatible and IPv4-mapped addresses.

g requests responder's global-scope addresses.

s requests responder's site-local addresses.

l requests responder's link-local addresses.

A requests responder's anycast addresses. Without this character, the responder will return unicast addresses only. With this character, the responder will return anycast addresses only. Note that the specification does not specify how to get responder's anycast addresses. This is an experimental option.

-N Probe node information multicast group address (ff02::2:ffxx:xxxx). *host* must be string hostname of the target (must not be a numeric IPv6 address). Node information multicast group will be computed based on given *host*, and will be used as the final destination. Since node information multicast group is a link-local multicast group, outgoing interface needs to be specified by **-I** option.

When specified twice, the address (ff02::2:xxxx:xxxx) is used instead. The former is in RFC 4620, the latter is in an old Internet Draft draft-ietf-ipngwg-icmp-name-lookup. Note that KAME-derived implementations including FreeBSD use the latter.

-O Generate ICMPv6 Node Information supported query types query, rather than echo-request. **-s** has no effect if **-O** is specified.

-u By default, **ping** asks the kernel to fragment packets to fit into the minimum IPv6 MTU. The **-u** option will suppress the behavior in the following two levels: when the option is specified once, the behavior will be disabled for unicast packets. When the option is more than once, it will be disabled for both unicast and multicast packets.

-Y Same as **-y**, but with old packet format based on 03 draft. This option is present for backward compatibility. **-s** has no effect if **-y** is specified.

-y Generate ICMPv6 Node Information DNS Name query, rather than echo-request. **-s** has no effect if **-y** is specified.

IPv6-hops
IPv6 addresses for intermediate nodes, which will be put into type 0 routing header.

IPv6-host
IPv6 address of the final destination node.

Experimental options only for IPv6 target

-E Enables transport-mode IPsec encapsulated security payload.

-Z Enables transport-mode IPsec authentication header.

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, maximum, and standard deviation of the round-trip times will be written to the standard 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**.

IPv4 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.

The maximum possible value of this field is 255, and some 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 64; this is what current FreeBSD systems do. In this case the TTL value in the received packet will be 64 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.

EXIT STATUS

The **ping** utility exits with one of the following values:

0 At least one response was heard from the specified *host*.

2 The transmission was successful but no responses were received.

any other value
An error occurred.

EXAMPLES

The following will send ICMPv6 echo request to dst.example.com.

```
ping -6 -n dst.example.com
```

The following will probe hostnames for all nodes on the network link attached to wi0 interface. The address ff02::1 is named the link-local all-node multicast address, and the packet would reach every node on the network link.

```
ping -6 -y ff02::1%wi0
```

The following will probe addresses assigned to the destination node, dst.example.com.

```
ping -6 -k agl dst.example.com
```

SEE ALSO

netstat(1), icmp(4), icmp6(4), inet6(4), ip6(4), ifconfig(8), routed(8), traceroute(8), traceroute6(8)

A. Conta and S. Deering, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 2463, December 1998.

Matt Crawford, *IPv6 Node Information Queries*, draft-ietf-ipngwg-icmp-name-lookups-09.txt, May 2002, work in progress material.

HISTORY

The **ping** utility appeared in 4.3BSD. The **ping6** utility with IPv6 support first appeared in the WIDE Hydrangea IPv6 protocol stack kit.

IPv6 and IPsec support based on the KAME Project (*https://www.kame.net/*) stack was initially integrated into FreeBSD 4.0.

The **ping6** utility was merged to **ping** in Google Summer of Code 2019.

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 IPv4 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.

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