Using the ping Utility

The **ping utility** is the most basic TCP/IP utility, and it's included with most TCP/IP stacks for most platforms. In most cases, `ping` is a command-line utility, although there are many GUI implementations available. You use the ping utility for two primary purposes:

- To find out if a host is responding
- To find out if you can reach a host

Here's the syntax (you can use either command):

```
ping hostname
ping IP address
```

If you ping any station that has an IP address, the ICMP that's part of that particular host's TCP/IP stack will respond to the request. The ICMP test and response looks something like this:

```
C:\Users\clarusway>ping 3.225.75.90

Pinging 3.225.75.90 with 32 bytes of data:
Reply from 3.225.75.90: bytes=32 time=137ms TTL=233
Reply from 3.225.75.90: bytes=32 time=136ms TTL=233
Reply from 3.225.75.90: bytes=32 time=134ms TTL=233
Reply from 3.225.75.90: bytes=32 time=134ms TTL=233

Ping statistics for 3.225.75.90:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 134ms, Maximum = 137ms, Average = 135ms
```

Because we've received a reply from the destination station we know that we can reach the host and that it's responding to basic IP requests. Don't forget that you can use name resolution and ping to a name, such as `ping www.clarusway.com`. Most versions of `ping` work the same way, but there are some switches you can use to specify certain information, like the number of packets to send, how big a packet to send, and so on. And if you're running the Windows command-line version of `ping`, just use the `/?` or `-?` switch to display a list of the available options like this:

```
C:\Users\clarusway>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
            [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP
                   Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
                   Per RFC 5095 the use of this routing header has been
                   deprecated. Some systems may drop echo requests if
                   this header is used.
    -S srcaddr     Source address to use.
    -c compartment Routing compartment identifier.
    -p             Ping a Hyper-V Network Virtualization provider address.
    -4             Force using IPv4.
    -6             Force using IPv6.
```

As you can see, there are many options you can use with the `ping` command from a Windows DOS prompt.

The `-a` switch is handy if you have name resolution (such as a DNS server), you can see the name of the destination host even if you only know its IP address. The `-n` switch sets the number of echo requests to send, where four is the default, and the `-w` switch allows you to adjust the time-out in milliseconds. The default ping time-out is 1 second (1,000 ms).

The `-6` is also nice if you want to ping an IPv6 host. And then there's `-t`, which keeps the ping running.

From a MAC, you can use the `ping6` command. Here are the options:

```
$ ping6
usage: ping6 [-DdfHmnNoqrRtvwW] [-a addrtype] [-b bufsiz] [-B boundif]
[-c count][-g gateway] [-h hoplimit] [-I interface] [-i wait] [-l preload][-p
pattern] [-S sourceaddr] [-s packetsize] [-z tclass]
[hops ...] host
```

Using the Address Resolution Protocol

The **Address Resolution Protocol (ARP)** is part of the TCP/IP protocol stack. It's used to translate TCP/IP addresses to MAC addresses using broadcasts. When a machine running TCP/IP wants to know which machine on an Ethernet network is using a certain IP address, it will send an ARP broadcast that says, in effect, "Who is IP address xxx.xxx.xxx.xxx?" The machine that owns the specific address will respond with its own MAC address, supplying the answer. The machine that made the inquiry will respond by adding the newly gained information to its own ARP table.

The ARP table in Windows includes a list of TCP/IP addresses and their associated physical (MAC) addresses. This table is cached in memory so that Windows doesn't have to perform ARP lookups for frequently accessed TCP/IP addresses like those of servers and default gateways. Each entry contains an IP address and a MAC address plus a value for TTL that determines how long each entry will remain in the ARP table.

Remember that the ARP table contains two kinds of entries:

- Dynamic
- Static

Dynamic ARP table entries are created whenever the Windows TCP/IP stack performs an ARP lookup but the MAC address isn't found in the ARP table. When the MAC address of the requested IP address is finally found or resolved, that information is then added into the ARP table as a dynamic entry. Whenever a request to send a packet to the host is sent to the Data Link layer, the ARP cache is checked first before an ARP broadcast is sent out.

Using the arp Utility

ARP is used by IP to determine the MAC address of a device that exists on the same subnet as the requesting device. When a TCP/IP device needs to forward a packet to a device on the local subnet, it first looks in its own table, called an ARP cache or MAC address lookup table, for an association between the known IP address of the destination device on the local subnet and that same device's MAC address. The cache is called that because the contents are periodically weeded out.

If no association that includes the destination IP address can be found, the device will then send out an ARP broadcast that includes its own MAC and IP information as well as the IP address of the target device and a blank MAC address field. Filling in that blank is the object of the whole operation—it's the unknown value that the source device is requesting to be returned to it in the form of an ARP reply. Windows includes a utility called `arp` that allows us to check out the operating system's ARP cache. To view this, from a Windows DOS prompt, use the `arp` command like this:

```
C:\Users\clarusway>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                table are displayed.
  -g            Same as -a.
  -v            Displays current ARP entries in verbose mode.  All invalid
                entries and entries on the loop-back interface will be shown.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr. inet_addr may be
                wildcarded with * to delete all hosts.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a
```

The Windows `arp` utility is primarily useful for resolving duplicate IP addresses. For example, let's say your workstation receives its IP address from a DHCP server but it accidentally receives the same address that some other workstation gets. And so, when you try to ping it, you get no response. Your workstation is basically confused—it's trying to determine the MAC address, and it can't because two machines are reporting that they have the same IP address. To solve this little snag, you can use the `arp` utility to view your local ARP table and see which TCP/IP address is resolved to which MAC address.

To display the entire current ARP table, use the `arp` command with the `-a` switch like so to show you the mac address lookup table:

```
C:\Users\clarusway>arp -a

Interface: 192.168.1.22 --- 0xa
  Internet Address      Physical Address      Type
  192.168.1.1           24-00-ba-b8-c7-ec     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Now, from this output, you can tell which MAC address is assigned to which IP address. Then, for static assignments, you can tell which workstation has a specific IP address and if it's indeed supposed to have that address by examining your network documentation.

For DHCP-assigned addresses, you can begin to uncover problems stemming from multiple DHCP scopes or servers doling out identical addresses and other common configuration issues. And remember that under normal circumstances, you shouldn't see IP addresses in the ARP table that isn't a member of the same IP subnet as the interface.

Using the nslookup Utility

Whenever you're configuring a server or a workstation to connect to the Internet, you've got to start by configuring DNS if you want name resolution to happen. When configuring DNS, it's a very good thing to be able to test what IP address DNS is returning to ensure that it's working properly. The nslookup utility allows you to query a name server and quickly find out which name resolves to which IP address.

Tip:

- The Unix **dig** (short for domain information groper) utility does the exact same thing as **nslookup**. It's primarily a command-line utility that allows you to perform a single DNS lookup for a specific entity, but it can also be employed in batch mode for a series of lookups.

You can run `nslookup` from a Windows command prompt. When you're inside this utility, the command prompt will change from something similar to a `C:\>` sign to a shorter `>` sign. It will also display the name and IP address of the default DNS server you will be querying. Then you can start using `nslookup`. The following output gives you a sample of the display after the `nslookup` command has been entered at the `C:\>` prompt.

```
C:\Users\clarusway> nslookup
Default Server: gnt-corpdc1.globalnet.local
Address: 10.100.36.12
>
```

The primary job of `nslookup` is to tell you the many different features of a particular domain name, the names of the servers that serve it, and how they're configured. To get that, just type in a domain name at the `>` prompt, and the `nslookup` utility will then return this information:

```
> clarusway.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    clarusway.com
Addresses:  3.225.75.90
          54.164.151.235
```

What this tells you is that the server that returned the information is not responsible (authoritative) for the zone information of the domain for which you requested an address and that the name server for the domain clarusway.com is located at the IP address `3.225.75.90`.

Using the mtr Utility

`Mtr` or **My traceroute** is a computer program that combines the functions of the traceroute and ping utilities in a single network diagnostic tool. It also adds round-trip time and packet loss to the output. `Mtr` probes routers on the route path by limiting the number of hops individual packets are allowed to traverse and listening to news of their termination. It will regularly repeat this process (usually once per second) and keep track of the response times of the hops along the path.

`Mtr` is available for Linux or Unix. Third-party applications of Mtr are available to install on Windows, but Microsoft did respond with its own version of Mtr —it's called `pathping` and it provides the same functions as `Mtr`. Here's a look at the output and the options:

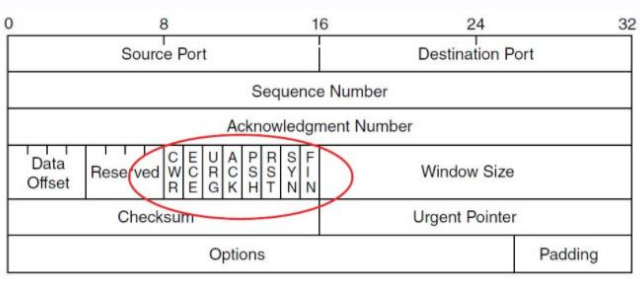
```
C:\Users\clarusway>pathping

Usage: pathping [-g host-list] [-h maximum_hops] [-i address] [-n]
                [-p period] [-q num_queries] [-w timeout]
                [-4] [-6] target_name

Options:
    -g host-list     Loose source route along host-list.
    -h maximum_hops  Maximum number of hops to search for target.
    -i address       Use the specified source address.
    -n               Do not resolve addresses to hostnames.
    -p period        Wait period milliseconds between pings.
    -q num_queries   Number of queries per hop.
    -w timeout       Wait timeout milliseconds for each reply.
    -4               Force using IPv4.
    -6               Force using IPv6.
```

Using the Nmap Utility

Nmap is one of the most popular port scanning tools used today. After performing scans with certain flags set in the scan packets, security analysts (and hackers) can make certain assumptions based on the responses received. These flags are used to control the TCP connection process and so are present only in TCP packets. The below figure shows a TCP header with the important flags circled. Normally flags are "turned on" because of the normal TCP process, but hackers can craft packets to check the flags they want to check.



TCP flags

- URG: Urgent pointer field significant
- ACK: Acknowledgment field significant
- PSH: Push function
- RST: Reset the connection
- SYN: Synchronize sequence numbers
- FIN: No more data from sender

Security analysts and hackers alike can perform scans with these flags set in the scan packets to get responses that allow them to determine the following information:

- If a port is open on a device
- If the port is blocked by a firewall before its gets to the device

Nmap can also be used as follows:

- To determine the live hosts on a network
- To create a logical "map" of the network