Using The Ping Command – Network Testing

The ping command is one of the most often used networking utilities for detecting devices on a network and for troubleshooting network problems.

When you ping a device you send that device a short message, which it then sends back **(the echo**).

The general format is **ping hostname** or **ping IPaddress**.

Example

ping www.google.com or **ping 216.58.208.68**

The **ping command** is one of the most often used networking utilities for troubleshooting network problems.

You can use the **ping command** to test the availability of a networking device (usually a computer) on a network.

When you ping a device you send that device a short message, which it then sends back (the echo).

If you receive a reply then the device is working OK , if you don't then:

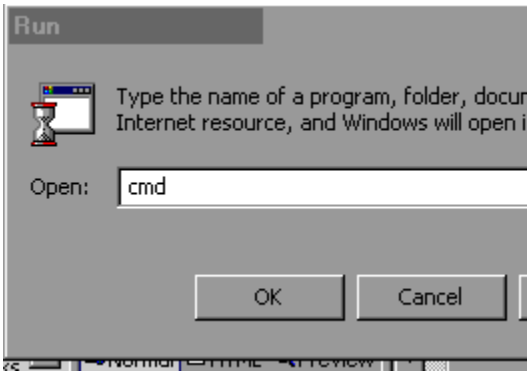
- The device is faulty, disconnected, switched off, incorrectly configured
- Your network or the device you are working on is not working properly.

Note: In this tutorial we will use the **ping command** on Windows but it works the same on Linux

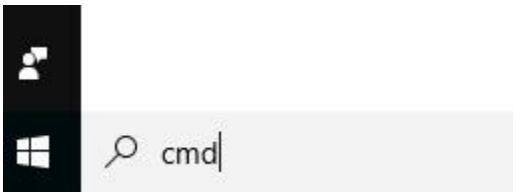
Ping Command Prompt

To use the ping command you go to the command line.

On Windows (XP,7) – **Start Menu>Run** and enter **cmd** to open a command prompt.



On Windows 10 type**cmd** into the search box and select the **cmd prompt** from the displayed programs.



You can use the **ping cmd** with an **IP address** or the **computer/host name**.

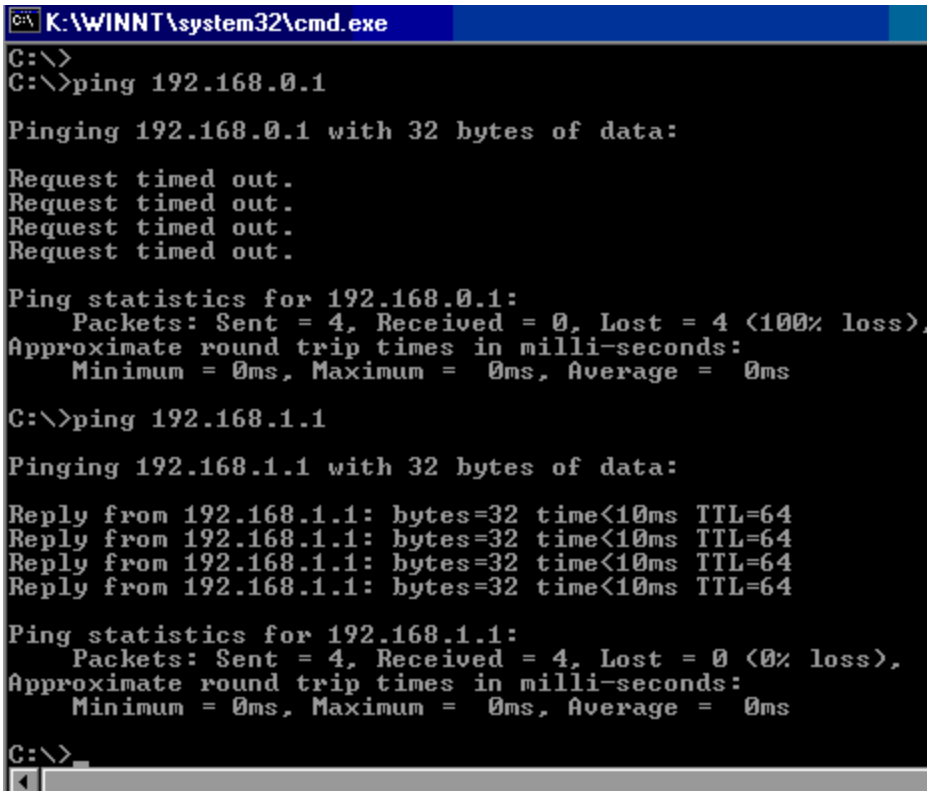
To **ping an IP address** go to a**cmd prompt** and enter:

Ping IP Address e.g. **ping 192.169.0.1** or to ping a **computer name:**

ping computer name e.g. **ping Computer1**

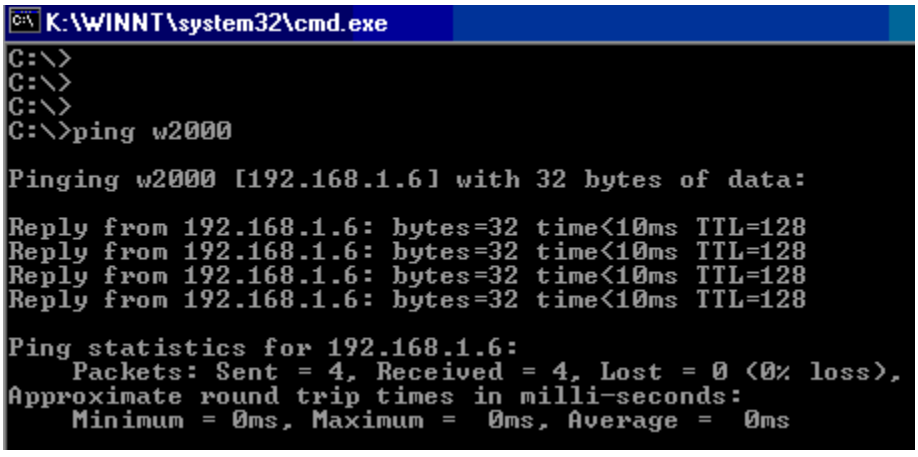
The screen shot below shows how to use the command with an IP address.

I have shown both a **failed ping** (192.168.0.1), and a **successful ping** (192.168.1.1)



Note: a failed ping results in a **request timed out** response, and a success results in the **reply from** message with the **round trip delay** in milliseconds.

The screen shot below shows how to use the ping command with the **computer name**.



Although this is easier to use a computer name than the IP address, it is only good if it works..

If it fails it is **not conclusive** as there is an extra stage called **name resolution** involved, and that could be at fault.

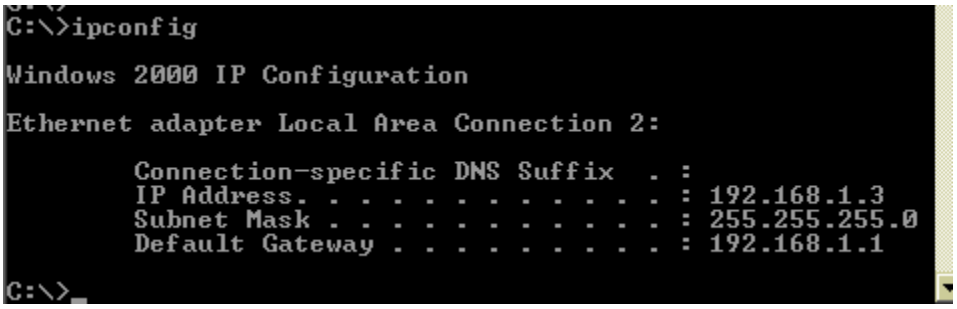
2. ipconfig Command

Another indispensable and frequently used utility that is used for finding network information about your local machine-like IP addresses, DNS addresses etc

Basic Use: Finding Your IP Address and Default Gateway

Type the command ipconfig at the prompt.

The following is displayed



Ip config has a number of switches the most common are**:**

ipconfig /all – displays more information about the network setup on your systems including the MAC address.

ipconfig /release – release the current IP address

ipconfig /renew – renew IP address

ipconfig /? -shows help

ipconfig/flushdns – flush the dns cache

NSlookUp

nslookup is a network administration command-line tool available for many computer operating systems.

It is used for querying the Domain Name System (DNS) to obtain domain name or IP address mapping information.

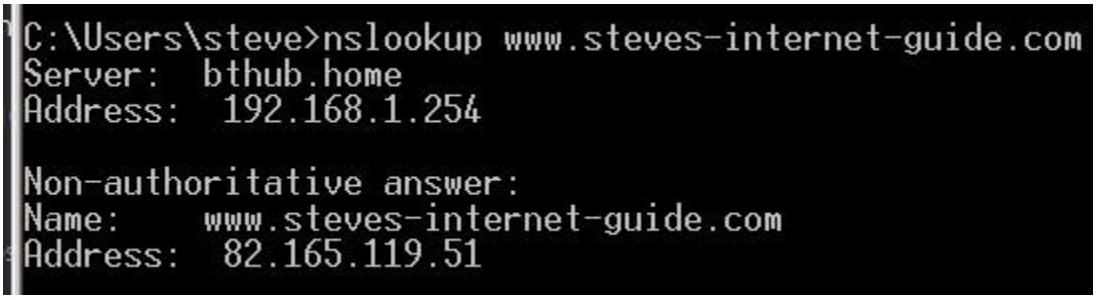
The main use of**nslookup** is for troubleshooting DNS related problems.

Nslookup can be use in **interactive** and **non-interactive** mode.

To use in interactive mode **type nslookup** at the command line and hit return.

You should get an**nslookup command prompt.**

To use in **non-interactive** mode type **nslookup options** at the command prompt.



Using Nslookup

To illustrate the use of nslookup we are going to use it to:

- Find the IP address of a host.

- Find the domain name of an IP address.
- Find mail servers for a domain.

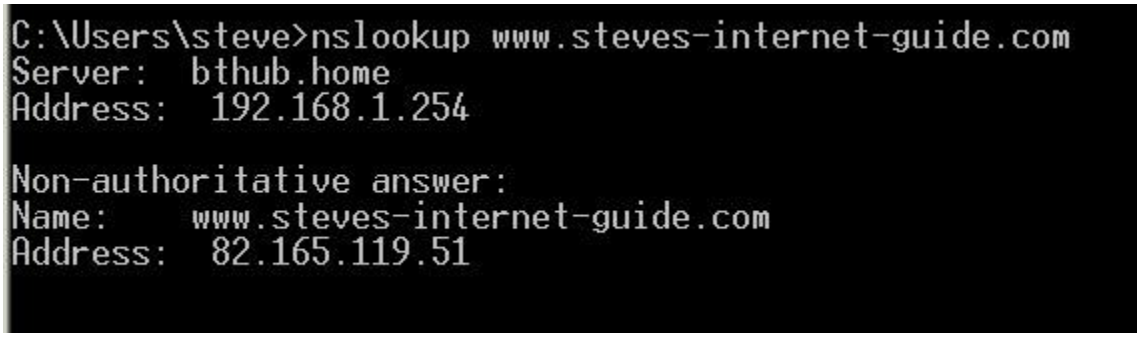
These are probably the most common usage scenarios.

Finding The IP Address of an Host-

To find the ip address of a host e.g. www.steves-internet-guide.com type:

nslookup www.steves-internet-guide.com

at a command prompt.



for an interactive lookup:

Reverse Lookup IP address to domain name

Type**nslookup** IP address

```
C:\Users\steve>nslookup 82.165.119.51
Server:  bthub.home
Address:  192.168.1.254

Name:    kundenserver.de
Address:  82.165.119.51
```

Find Mail Servers for a Domain

Type **nslookup -querytype=mx domain name**

```
C:\Users\steve>nslookup -querytype=mx steves-internet-guide.com
Server:  bthub.home
Address:  192.168.1.254

Non-authoritative answer:
steves-internet-guide.com       MX preference = 10, mail exchanger = mx00.1and1.co.uk
steves-internet-guide.com       MX preference = 10, mail exchanger = mx01.1and1.co.uk

mx01.1and1.co.uk        internet address = 217.72.192.67
mx00.1and1.co.uk        internet address = 212.227.15.41

C:\Users\steve>
```

General Usage Notes:

1. By default nslookup will use the domain server that is currently configured for your system.

You can switch DNS servers using **server name** or **server IP address** option.

To switch to using the **open dns server** address **208.67.222.222** then go to an interactive prompt and type:

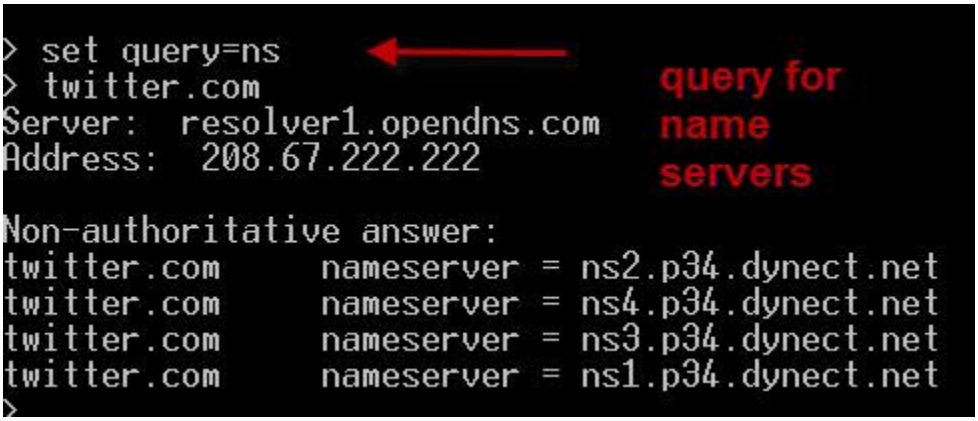
server208.67.222.222

2. You may notice that you get **non-authoritative answers**.

This is nothing to worry about as all it means is that the DNS server has already recently resolved this query.

It can retrieve the results from cache and doesn't need to contact the authoritative name server.

You can find out which name servers are responsible (**authoritative**) for a domain by setting the **query type to NS** and entering the domain name as shown below:



Common Questions

Q- Can I use dig instead of Nslookup?

A– Yes **Dig** was initially meant to replace nslookup but didn't. See Wiki – Dig isn't found on Windows.

Summary

Nslookup is a very handy tool for troubleshooting DNS related network problems.

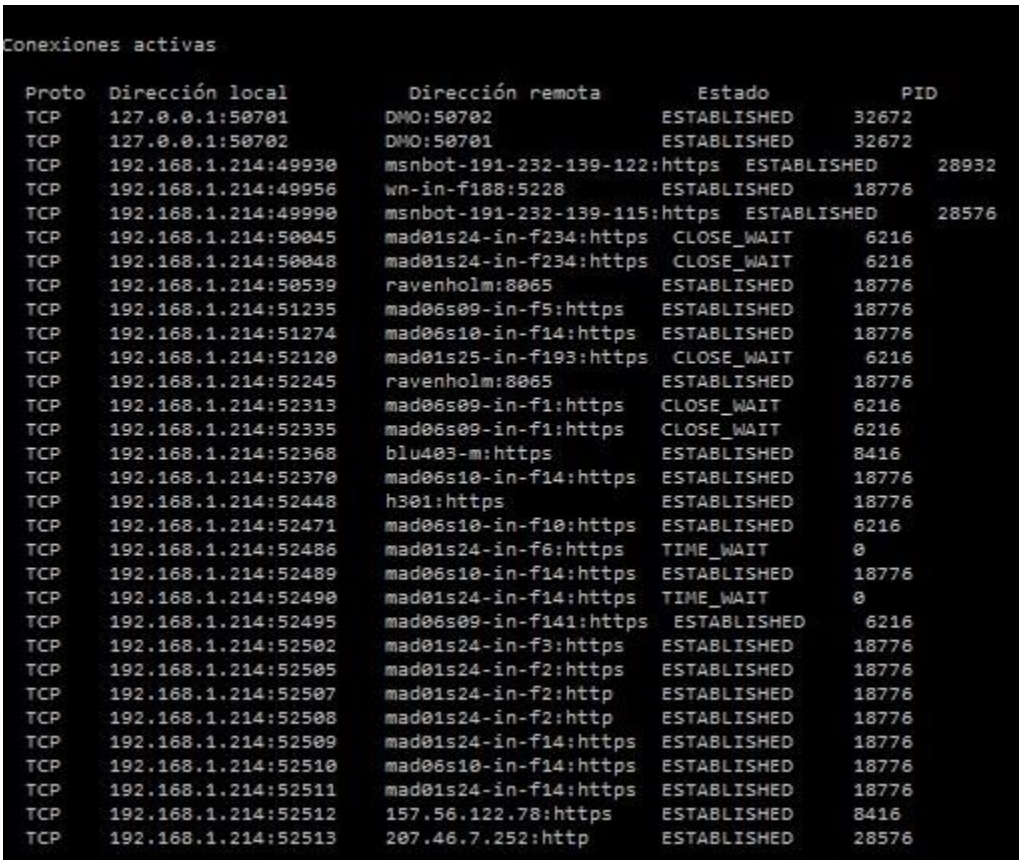
It is available on all the main Operating systems and can be used in interactive and command line mode.

Netstat (Unix/Windows)

Network command identifies all TCP connections and UDP open on a machine. Besides this, it allows us to know the following information:

- Routing tables to meet our network interfaces and its outputs.
- Ethernet statistics that show sent and received packages and possible errors.
- To know the id of the process that is being used by the connection.

Netstat is another basic command as Ping that meets many elementary functions. **Some of the elements, that Pandora FMS agents use to get information of the system, are the traffic statistics, the number of open connections and most importantly, the number of closing pending connections or in a settlement process.** An unusual growth in these metrics can be a serious problem , and it may be due to a performance problem on our server or even an external attack.



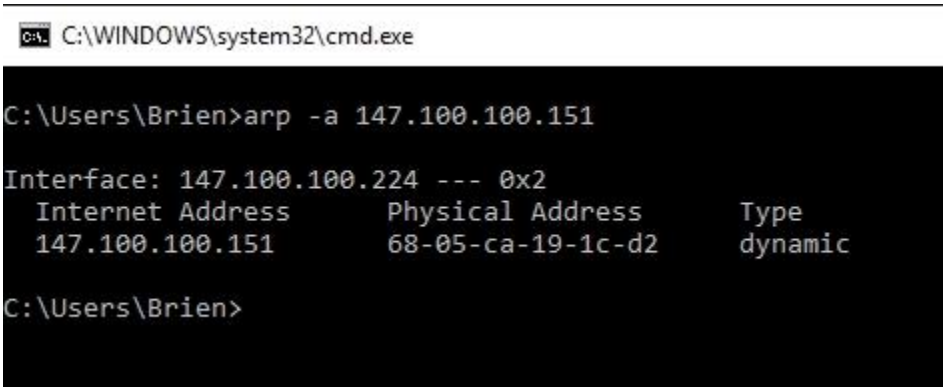
```
Conexiones activas

  Proto  Dirección local          Dirección remota        Estado           PID
  TCP    127.0.0.1:50701         DMO:50702              ESTABLISHED     32672
  TCP    127.0.0.1:50702         DMO:50701              ESTABLISHED     32672
  TCP    192.168.1.214:49930     msnbot-191-232-139-122:https  ESTABLISHED     28932
  TCP    192.168.1.214:49956     wn-in-f188:5228        ESTABLISHED     18776
  TCP    192.168.1.214:49990     msnbot-191-232-139-115:https  ESTABLISHED     28576
  TCP    192.168.1.214:50045     mad01s24-in-f234:https  CLOSE_WAIT      6216
  TCP    192.168.1.214:50048     mad01s24-in-f234:https  CLOSE_WAIT      6216
  TCP    192.168.1.214:50539     ravenholm:8065         ESTABLISHED     18776
  TCP    192.168.1.214:51235     mad06s09-in-f5:https   ESTABLISHED     18776
  TCP    192.168.1.214:51274     mad06s10-in-f14:https  ESTABLISHED     18776
  TCP    192.168.1.214:52120     mad01s25-in-f193:https  CLOSE_WAIT      6216
  TCP    192.168.1.214:52245     ravenholm:8065         ESTABLISHED     18776
  TCP    192.168.1.214:52313     mad06s09-in-f1:https   CLOSE_WAIT      6216
  TCP    192.168.1.214:52335     mad06s09-in-f1:https   CLOSE_WAIT      6216
  TCP    192.168.1.214:52368     blu403-m:https         ESTABLISHED     8416
  TCP    192.168.1.214:52370     mad06s10-in-f14:https  ESTABLISHED     18776
  TCP    192.168.1.214:52448     h301:https             ESTABLISHED     18776
  TCP    192.168.1.214:52471     mad06s10-in-f10:https  ESTABLISHED     6216
  TCP    192.168.1.214:52486     mad01s24-in-f6:https   TIME_WAIT       0
  TCP    192.168.1.214:52489     mad06s10-in-f14:https  ESTABLISHED     18776
  TCP    192.168.1.214:52490     mad01s24-in-f14:https  TIME_WAIT       0
  TCP    192.168.1.214:52495     mad06s09-in-f141:https  ESTABLISHED     6216
  TCP    192.168.1.214:52502     mad01s24-in-f3:https   ESTABLISHED     18776
  TCP    192.168.1.214:52505     mad01s24-in-f2:https   ESTABLISHED     18776
  TCP    192.168.1.214:52507     mad01s24-in-f2:http    ESTABLISHED     18776
  TCP    192.168.1.214:52508     mad01s24-in-f2:http    ESTABLISHED     18776
  TCP    192.168.1.214:52509     mad01s24-in-f14:https  ESTABLISHED     18776
  TCP    192.168.1.214:52510     mad06s10-in-f14:https  ESTABLISHED     18776
  TCP    192.168.1.214:52511     mad01s24-in-f14:https  ESTABLISHED     18776
  TCP    192.168.1.214:52512     157.56.122.78:https    ESTABLISHED     8416
  TCP    192.168.1.214:52513     207.46.7.252:http      ESTABLISHED     28576
```

ARP

The ARP command corresponds to the Address Resolution Protocol. Although it is easy to think of network communications in terms of IP addressing, packet delivery is ultimately dependent on the Media Access Control (MAC) address of the device's network adapter. This is where the Address Resolution Protocol comes into play. Its job is to map IP addresses to MAC addresses.

Windows devices maintain an ARP cache, which contains the results of recent ARP queries. You can see the contents of this cache by using the ARP -A command. If you are having problems communicating with one specific host, you can append the remote host's IP address to the ARP -A command.



How ARP works

When a device needs to send data to another device on an IP network, it has to be able to determine the recipient device's MAC address. This is where the Address Resolution Protocol, or ARP, comes into play. ARP's job is to determine the MAC address that corresponds to a given IP address.

Of course, network communications would be really slow if an ARP translation had to be performed for every single packet that was sent across a network. To improve performance, devices make use of an ARP cache. The ARP cache is simply a list of known IP address to MAC address mappings. Therefore, if a device needs to determine the MAC address that corresponds to a particular IP address, it looks at the ARP cache to determine whether the recipient's MAC address is already known.

Missing MAC

If the recipient's MAC address does not appear in the ARP cache, then the device that needs to transmit data to a given recipient sends an ARP request across the local subnet. The ARP request is essentially a request for the recipient to respond with its MAC address.

This of course, raises a big question. How is the sender able to send an ARP request to the recipient when it does not know the recipient's MAC address? After all, the whole reason why the sender needs the recipient's MAC address in the first place is so that the sender can send data to the recipient.

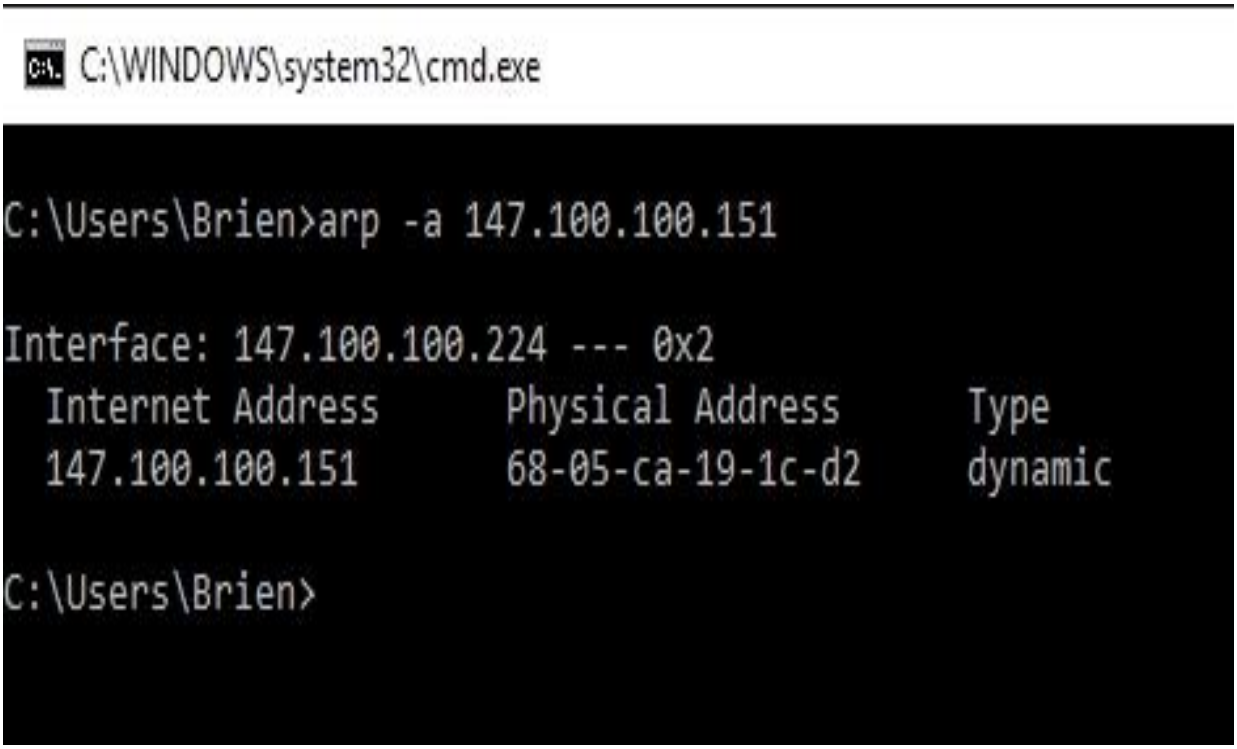
If the recipient's MAC address is unknown, then the sender cannot transmit data directly to the recipient. That being the case, the sender transmits a broadcast message across the local subnet. Broadcast messages are different from normal IP communications because they are sent to every host on the subnet. Therefore, an ARP request involves broadcasting the recipient's IP address and asking the hosts on the subnet to check to see if they are using the IP address that is contained within the ARP request. The host whose IP address is referenced within the ARP request replies with an ARP reply, containing its MAC address. Upon receiving this reply, the device that initiated the request updates its ARP cache, and is able to begin communicating with the recipient.

On the surface, the workings of the Address Resolution Protocol can seem to be completely theoretical. However, the Windows operating system exposes much of this functionality through a command that is appropriately named ARP. In fact, the ARP command has been a part of the Windows operating system for decades.

The ARP command lets you view and modify a device's ARP cache. To show you how this works, consider that before I started writing this article, I spent a bit of time communicating with a server on my network with an IP address of 147.100.100.151. Because my PC recently communicated with this server, the server's information should be in my PC's ARP cache. Therefore, if I wanted to look up that server's MAC address, I could do so by entering the following command:

```
ARP -A 147.100.100.151
```

You can see the command's output in the figure below:



In this case, I specified a particular server's IP address, but it is possible to examine the ARP cache in its entirety. To do so, just enter the `ARP -A` command without specifying an IP address. Doing so will cause the entire cache to be displayed.

OK, that makes for a nice party trick, but you're probably wondering if there is any practical, real-world use for this utility. Believe it or not, there is something that the ARP command works really well for.

Let's suppose for a moment that an IP address conflict occurs on your network, as a result of two hosts being assigned the same IP address. Let's also pretend that you manage to track down the offending host, and assign it a new IP address, but that a Windows device on your network is having trouble communicating with one of those hosts even though the IP address conflict has been resolved. The problem is most likely related to an invalid entry being written to the ARP cache as a result of the IP address conflict. You could therefore use the ARP command to verify and to correct the problem.

A moment ago, I showed you how to use the ARP command to look at a particular IP address within the ARP cache. The information that was displayed for this address in the previous screen capture was correct, because no IP address conflict has occurred on my network. For the sake of discussion, however, let's pretend that the information that was returned by the cache lookup was incorrect. In such a situation, I could use the -D switch to remove the entry from the ARP cache. The actual command would be:

```
ARP -D 147.100.100.151
```

If I wanted to create a new ARP cache entry with the correct MAC address, I could use the -S switch to create a static entry. Static entries are entries that are manually added to the ARP cache. It is worth noting, however, that static entries are removed when the system is rebooted.

I have to be honest with you in that I have never once had to create a static ARP table entry. Remember, ARP table entries are created automatically as a result of trying to communicate with a host. Therefore, if you remove an offending entry from the ARP cache, a new entry will take its place the next time that you attempt communications with the host. Assuming that the IP address conflict has been resolved, the new ARP cache entry should contain information that is correct.