**Experiment 1**

**Date:** 20-07-2021

**Aim:** To explore the basic networking commands.

**Software Used:** Command Prompt.

**Theory:**

1. **tracert:** tracert command determines the path taken to a destination by sending Internet Control Message Protocol (ICMP) Echo Request messages to the destination with incrementally increasing Time to Live (TTL) field values. The path displayed is the list of near-side router interfaces of the routers in the path between a source host and a destination. The near-side interface is the interface of the router that is closest to the sending host in the path. Used without parameters, tracert displays help.

This diagnostic tool determines the path taken to a destination by sending ICMP Echo Request messages with varying Time to Live (TTL) values to the destination. Each router along the path is required to decrement the TTL in an IP packet by at least 1 before forwarding it.

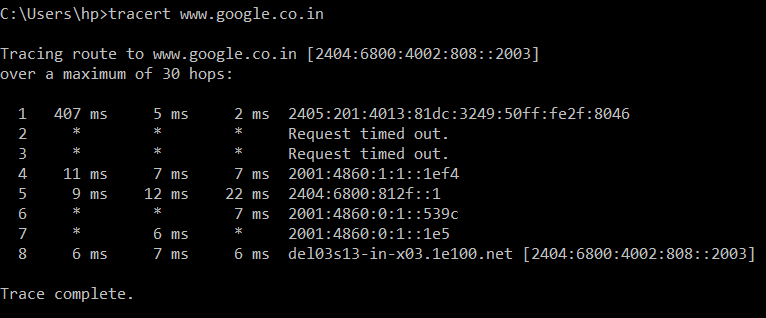
Effectively, the TTL is a maximum link counter. When the TTL on a packet reaches 0, the router is expected to return an ICMP Time Exceeded message to the source computer. Tracert determines the path by sending the first Echo Request message with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum number of hops is reached. The maximum number of hops is 30 by default and can be specified using the -h parameter.

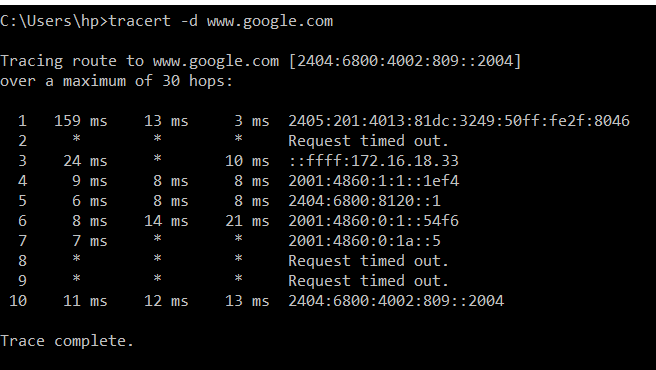
The path is determined by examining the ICMP Time Exceeded messages returned by intermediate routers and the Echo Reply message returned by the destination. However, some routers do not return Time Exceeded messages for packets with expired TTL values and are invisible to the tracert command. In this case, a row of asterisks (\*) is displayed for that hop.

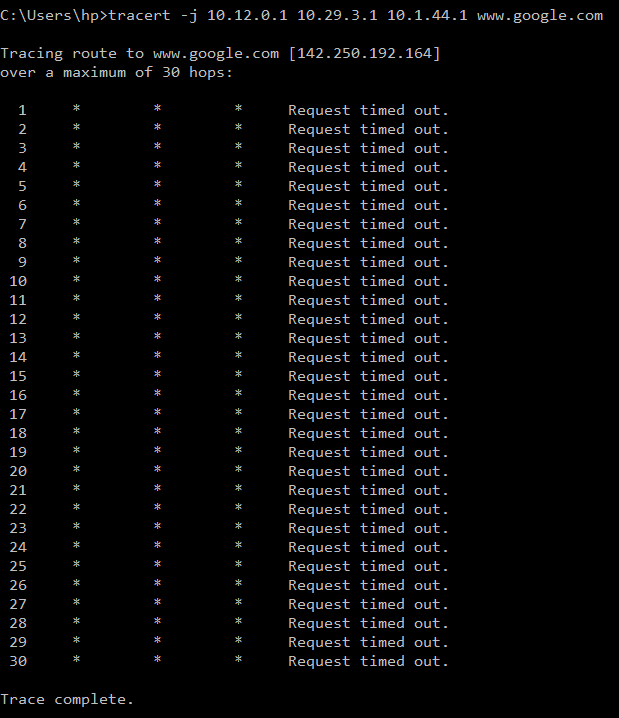
**Syntax:** tracert [-d] [-h MaximumHops] [-j HostList] [-w Timeout] [TargetName]

**Parameters:**

* **-d:** Prevents tracert from attempting to resolve the IP addresses of intermediate routers to their names. This can speed up the display of tracert results.
* **-h:** MaximumHops Specifies the maximum number of hops in the path to search for the target (destination). The default is 30 hops.
* **-j:** HostList Specifies that Echo Request messages use the Loose Source Route option in the IP header with the set of intermediate destinations specified in HostList. With loose source routing, successive intermediate destinations can be separated by one or multiple routers. The maximum number of addresses or names in the host list is 9. The HostList is a series of IP addresses (in dotted decimal notation) separated by spaces.
* **-w:** Timeout Specifies the amount of time in milliseconds to wait for the ICMP Time Exceeded or Echo Reply message corresponding to a given Echo Request message to be received. If not received within the time-out, an asterisk (\*) is displayed. The default time-out is 4000 (4 seconds).







1. **ping:** ping command verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages are displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.

Or

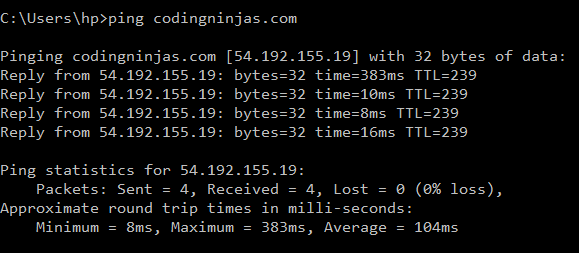
The ping command lets you verify that you have network connectivity with another network device. It is commonly used to help troubleshoot networking issues. To use ping, provide the IP address or machine name of the other device.

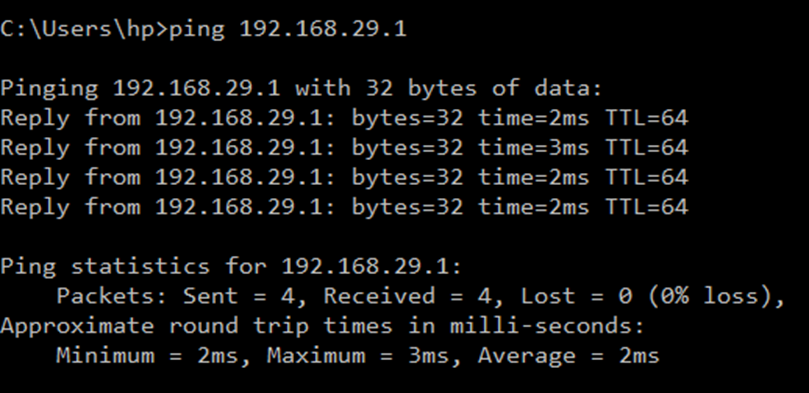
You can use ping to test both the computer name and the IP address of the computer. If pinging the IP address is successful, but pinging the computer name is not, you might have a name resolution problem. In this case, ensure that the computer name you are specifying can be resolved through the local Hosts file, by using Domain Name System (DNS) queries, or through NetBIOS name resolution techniques.

**To test a TCP/IP configuration by using the ping command:**

* To quickly obtain the TCP/IP configuration of a computer, open Command Prompt, and then type **ipconfig**. From the display of the ipconfig command, ensure that the network adapter for the TCP/IP configuration you are testing is not in a Media disconnected state.
* At the command prompt, ping the loopback address by typing ping **127.0.0.1**.
* Ping the IP address of the computer.
* Ping the IP address of the default gateway. If the ping command fails, verify that the default gateway IP address is correct and that the gateway (router) is operational.
* Ping the IP address of a remote host (a host that is on a different subnet). If the ping command fails, verify that the remote host IP address is correct, that the remote host is operational, and that all of the gateways (routers) between this computer and the remote host are operational.
* Ping the IP address of the DNS server. If the ping command fails, verify that the DNS server IP address is correct, that the DNS server is operational, and that all of the gateways (routers) between this computer and the DNS server are operational.

**Syntax:** ping



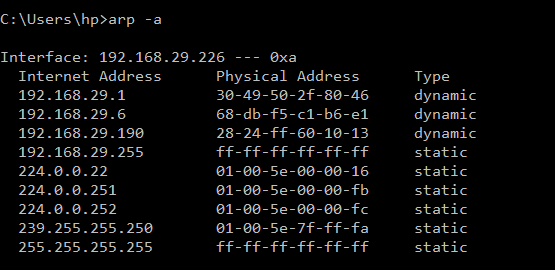


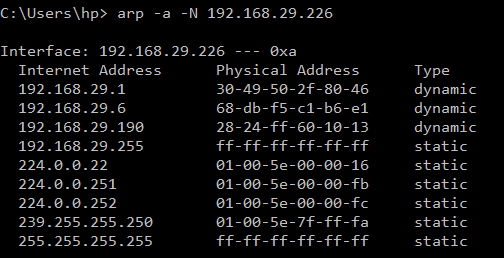
1. **arp:** arp command displays and modifies entries in the Address Resolution Protocol (ARP) cache, which contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer.

**Syntax:** arp [-a [InetAddr] [-N IfaceAddr]] [-g [InetAddr] [-N IfaceAddr]] [-d InetAddr [IfaceAddr]] [-s InetAddr EtherAddr [IfaceAddr]]

**Parameters:**

* **Used without parameters:** displays help
* **-a [InetAddr] [-N IfaceAddr]:** Displays current ARP cache tables for all interfaces. To display the ARP cache entry for a specific IP address, use arp -a with the InetAddr parameter, where InetAddr is an IP address. To display the ARP cache table for a specific interface, use the -N IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. The -N parameter is case-sensitive.
* **-g [InetAddr] [-N IfaceAddr]:** Identical to -a.
* **-d InetAddr [IfaceAddr]:** Deletes an entry with a specific IP address, where InetAddr is the IP address. To delete an entry in a table for a specific interface, use the IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. To delete all entries, use the asterisk (\*) wildcard character in place of InetAddr.
* **-s InetAddr EtherAddr [IfaceAddr]:** Adds a static entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. To add a static ARP cache entry to the table for a specific interface, use the IfaceAddr parameter where IfaceAddr is an IP address assigned to the interface.





1. **netstat:** netstat command displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols).

**Netstat provides statistics for the following:**

* **Proto:** The name of the protocol (TCP or UDP).
* **Local Address:** The IP address of the local computer and the port number being used. The name of the local computer that corresponds to the IP address and the name of the port is shown unless the -n parameter is specified. If the port is not yet established, the port number is shown as an asterisk (\*).
* **Foreign Address:** The IP address and port number of the remote computer to which the socket is connected. The names that corresponds to the IP address and the port are shown unless the -n parameter is specified. If the port is not yet established, the port number is shown as an asterisk (\*).

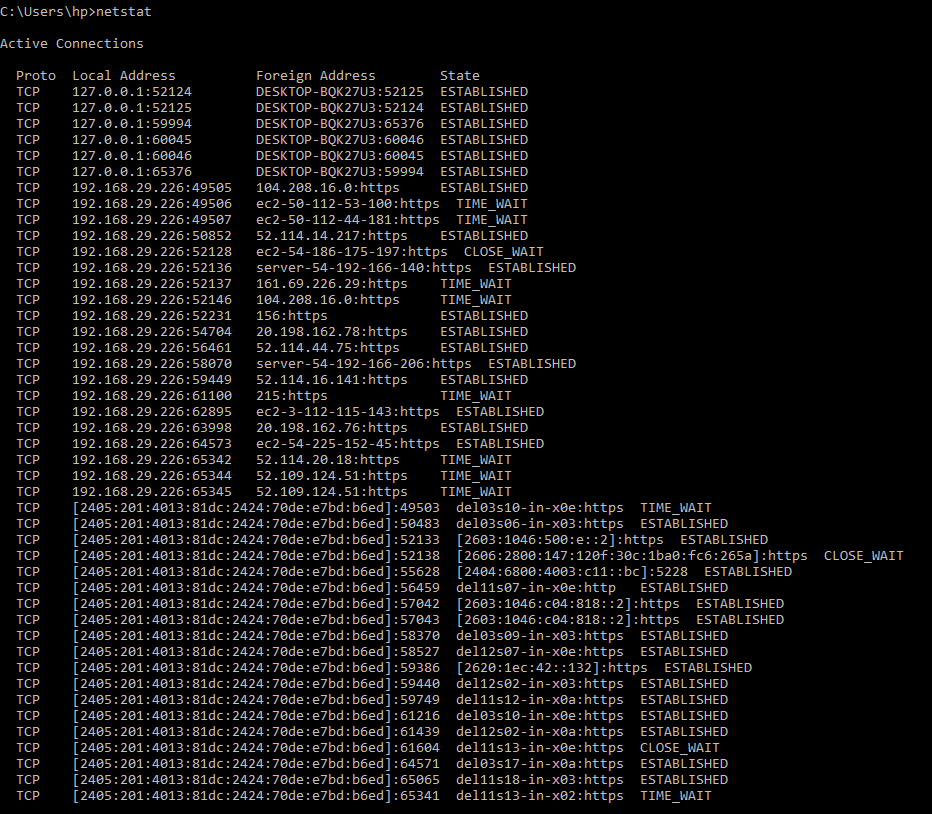
**(state) Indicates the state of a TCP connection. The possible states are as follows:**

* CLOSE\_WAIT
* CLOSED
* ESTABLISHED
* FIN\_WAIT\_1
* FIN\_WAIT\_2
* LAST\_ACK
* LISTEN
* SYN\_RECEIVED
* SYN\_SEND
* TIMED\_WAIT

**Syntax:** netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]

**Parameters:**

* **Used without parameters:** displays active TCP connections.
* **-a:** Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.
* **-e:** Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with -s.
* **-n:** Displays active TCP connections, however, addresses and port numbers are expressed numerically and no attempt is made to determine names.
* **-o:** Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the Processes tab in Windows Task Manager. This parameter can be combined with -a, -n, and -p.
* **-p:** Shows connections for the protocol specified by Protocol. In this case, the Protocol can be tcp, udp, tcpv6, or udpv6. If this parameter is used with -s to display statistics by protocol, Protocol can be tcp, udp, icmp, ip, tcpv6, udpv6, icmpv6, or ipv6.
* **-s:** Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The -p parameter can be used to specify a set of protocols.
* **-r:** Displays the contents of the IP routing table. This is equivalent to the route print command.
* **Interval:** Redisplays the selected information every Interval seconds. Press CTRL+C to stop the redisplay. If this parameter is omitted, netstat prints the selected information only once.
* **/?:** Displays help at the command prompt.



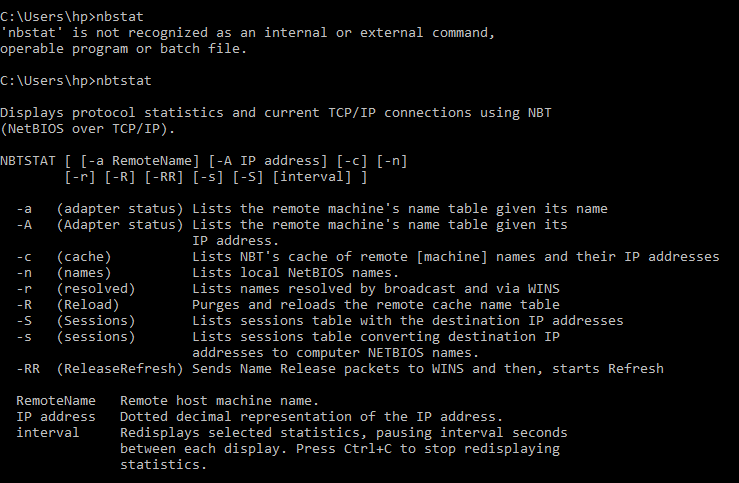
1. **nbtstat:** nbtstat command displays NetBIOS over TCP/IP (NetBT) protocol statistics. NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache. Nbtstat allows a refresh of the NetBIOS name cache and the names registered with Windows Internet Name Service (WINS). Nbtstat command-line parameters are case-sensitive.

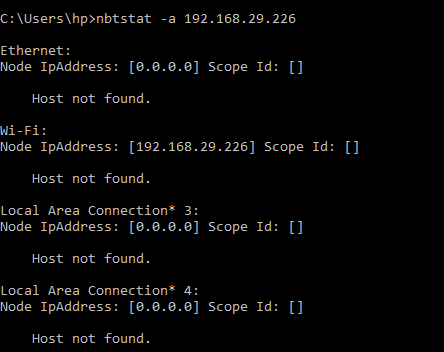
**Syntax:** nbtstat [-a RemoteName] [-A IPAddress] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [Interval]

**Parameters:**

**Used without parameters:** displays help.

* **-a:** RemoteName Displays the NetBIOS name table of a remote computer, where RemoteName is the NetBIOS computer name of the remote computer. The NetBIOS name table is the list of NetBIOS names that corresponds to NetBIOS applications running on that computer.
* **-A:** IPAddress Displays the NetBIOS name table of a remote computer, specified by the IP address (in dotted decimal notation) of the remote computer.
* **-c:** Displays the contents of the NetBIOS name cache, the table of NetBIOS names and their resolved IP addresses.
* **-n:** Displays the NetBIOS name table of the local computer. The status of Registered indicates that the name is registered either by broadcast or with a WINS server.
* **-r:** Displays NetBIOS name resolution statistics. On a Windows XP computer that is configured to use WINS, this parameter returns the number of names that have been resolved and registered using broadcast and WINS.
* **-R:** Purges the contents of the NetBIOS name cache and then reloads the #PRE-tagged entries from the Lmhosts file.
* **-RR:** Releases and then refreshes NetBIOS names for the local computer that is registered with WINS servers.
* **-s:** Displays NetBIOS client and server sessions, attempting to convert the destination IP address to a name.
* **-S:** Displays NetBIOS client and server sessions, listing the remote computers by destination IP address only.
* **Interval:** Redisplays selected statistics, pausing the number of seconds specified in Interval between each display. Press CTRL+C to stop redisplaying statistics. If this parameter is omitted, nbtstat prints the current configuration information only once.
* **/?:** Displays help at the command prompt.





1. **ipconfig:** Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. This command is most useful on computers that are configured to obtain an IP address automatically. This enables users to determine which TCP/IP configuration values have been configured by DHCP, Automatic Private IP Addressing (APIPA), or an alternate configuration.

* If the Adapter name contains any spaces, use quotation marks around the adapter name (that is, "Adapter Name").
* For adapter names, ipconfig supports the use of the asterisk (\*) wildcard character to specify either adapters with names that begin with a specified string or adapters with names that contain a specified string.
* For example, Local\* matches all adapters that start with the string Local and \*Con\* matches all adapters that contain the string Con.

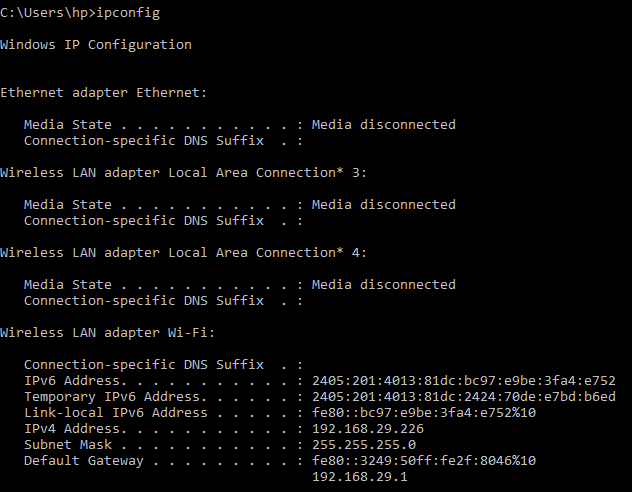
**Syntax:** ipconfig [/all] [/renew [Adapter]] [/release [Adapter]] [/flushdns] [/displaydns] [/registerdns] [/showclassid Adapter] [/setclassid Adapter [ClassID]]

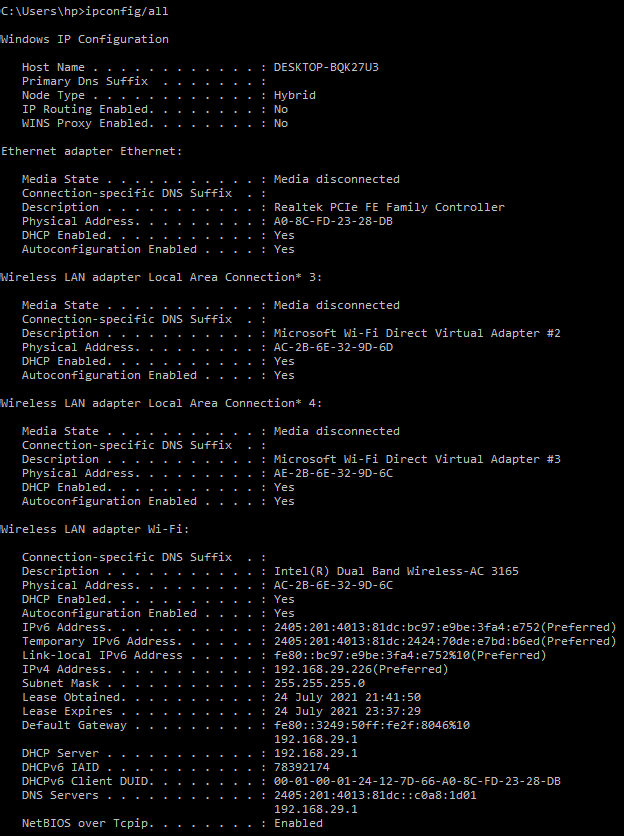
**Parameters:**

* **Used without parameters:** displays the IP address, subnet mask, and default gateway for all adapters.
* **/all:** Displays the full TCP/IP configuration for all adapters. Without this parameter, ipconfig displays only the IP address, subnet mask, and default gateway values for each adapter. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.
* **/renew [Adapter]:** Renews DHCP configuration for all adapters (if an adapter is not specified) or for a specific adapter if the Adapter parameter is included. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use ipconfig without parameters.
* **/release [Adapter]:** Sends a DHCPRELEASE message to the DHCP server to release the current DHCP configuration and discard the IP address configuration for either all adapters (if an adapter is not specified) or for a specific adapter if the Adapter parameter is included. This parameter disables TCP/IP for adapters configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use ipconfig without parameters.
* **/flushdns:** Flushes and resets the contents of the DNS client resolver cache. During DNS troubleshooting, you can use this procedure to discard negative cache entries from the cache, as well as any other entries that have been added dynamically.
* **/displaydns:** Displays the contents of the DNS client resolver cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records for name queries resolved by the computer. The DNS Client service uses this information to resolve frequently queried names quickly, before querying its configured DNS servers.
* **/registerdns:** Initiates manual dynamic registration for the DNS names and IP addresses that are configured at a computer. You can use this parameter to troubleshoot a failed DNS name registration or resolve a dynamic update problem between a client and the DNS server without rebooting the client computer. The DNS settings in the advanced properties of the TCP/IP protocol determine which names are registered in DNS.
* **/showclassid:** Adapter Displays the DHCP class ID for a specified adapter. To see the DHCP class ID for all adapters, use the asterisk (\*) wildcard character in place of Adapter. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically.
* **/setclassid:** Adapter [ClassID] Configures the DHCP class ID for a specified adapter. To set the DHCP class ID for all adapters, use the asterisk (\*) wildcard character in place of Adapter. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. If a DHCP class ID is not specified, the current class ID is removed.

**Examples:**

* **ipconfig:** To display the basic TCP/IP configuration for all adapters
* **ipconfig /all:** To display the full TCP/IP configuration for all adapters
* **ipconfig /renew "Local Area Connection":** To renew a DHCP-assigned IP address configuration for only the Local Area Connection adapter
* **ipconfig /flushdns:** To flush the DNS resolver cache when troubleshooting DNS name resolution problems
* **ipconfig /showclassid Local:** To display the DHCP class ID for all adapters with names that start with Local
* **ipconfig /setclassid "Local Area Connection" TEST:** To set the DHCP class ID for the Local Area Connection adapter to TEST

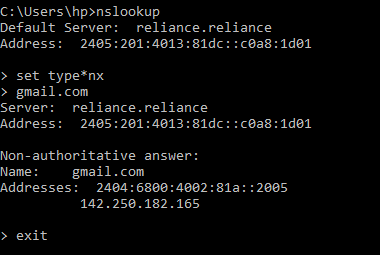




1. **nslookup:** nslookup (Name Server lookup) is a UNIX shell command to query Internet domain name servers.

**Definitions:**

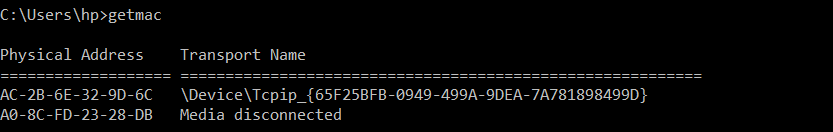
* **Nameserver:** These are the servers that the internet uses to find out more about the domain. Usually they are an ISP's computer.
* **Mailserver:** Where email is sent to.
* **Webserver:** The domains website.
* **FTPserver:** FTP is file transfer protocol, this server is where files may be stored.
* **Hostname:** The name of the host as given by the domain.
* **Real Hostname:** This is hostname that you get by reverse resolving the IP address, may be different to the given hostname.
* **IP Address:** Unique four numbered identifier that is obtained by resolving the hostname.



1. **getmac:** getmac command is used for quickly finding out your MAC address.

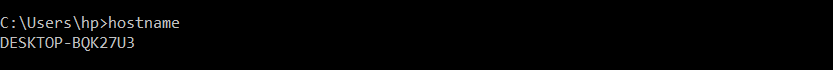
In order to be compliant with the IEEE 802 standards, each device must have a unique MAC (Media Access Control) address. The manufacturer of your device will assign it a MAC address and store it within the hardware. The getmac command provides an easy way to find the MAC address of your device. If you see more than one MAC address for your device, it will have multiple network adapters. As an example, a laptop with both Ethernet and Wi-Fi will have two separate MAC addresses. Some administrators will use the unique MAC addresses of devices to limit what can and cannot connect to a network.

**Syntax:** getmac



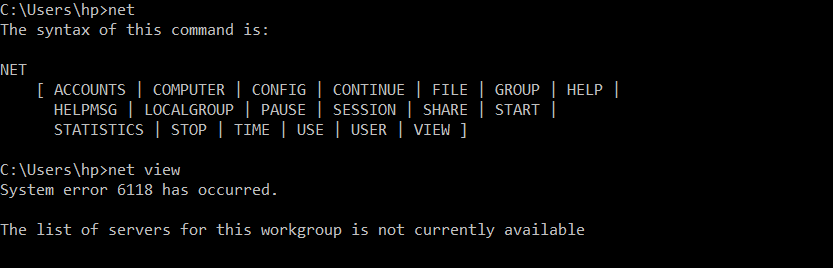
1. **hostname:** hostname command provides you with an easy way of identifying the hostname that has been assigned to your Windows device. There are ways of being able to find this through Windows but using the command line is much quicker. Simply type hostname into the command prompt and it will present you with the local computer name of your device.

**Syntax:** hostname



1. **net:** net command is definitely a versatile one, allowing you to manage many different aspects of a network and its settings such as network shares, users and print jobs, as just a few examples. Running just net won’t do much, but it will present you with a list of all the switches that are available. These include accounts to set password and logon requirements, file to show a list of open files and sessions to list, or even disconnect, sessions on the network. If you are ever in doubt as to what task each switch performs, run net help and you’ll find the answer.

**Syntax:** net

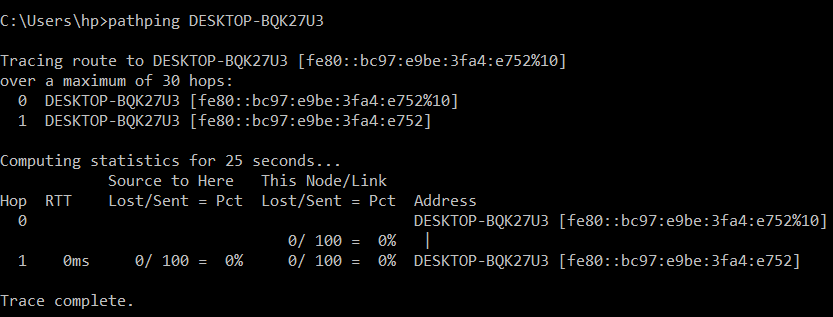


1. **pathping:** pathping command is used troubleshooting the network connection issues. It combines that best of both ping and tracert into a single utility. Enter pathping followed by a hostname into the command prompt and it will initiate what looks like a regular old tracert command. Let the process finish, however, and you will be provided with more detail than either ping or tracert can provide, such as latency reports and statistics on packet loss. Be patient when using the pathping command as it will take five minutes in order to gather all of the statistics for you.

**Syntax:** pathping [-g host-list] [-h maximum\_hops] [-i address] [-n] [-p period] [-q num\_queries] [-w timeout] [-4] [-6] target\_name

**Parameters:**

* **-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.



1. **netsh:** netsh command is used for displaying and configuring network adapters. Netsh is another very powerful command, allowing you to view and configure almost all of the network adapters in your device in much greater detail compared with some other commands. When you run the netsh command on its own, the command prompt will be shifted into network shell mode. Within this mode, there are several different “contexts”, such as one for DHCP-related commands, one for diagnostics and one for routing. It is possible to still run individual commands from netsh, though.

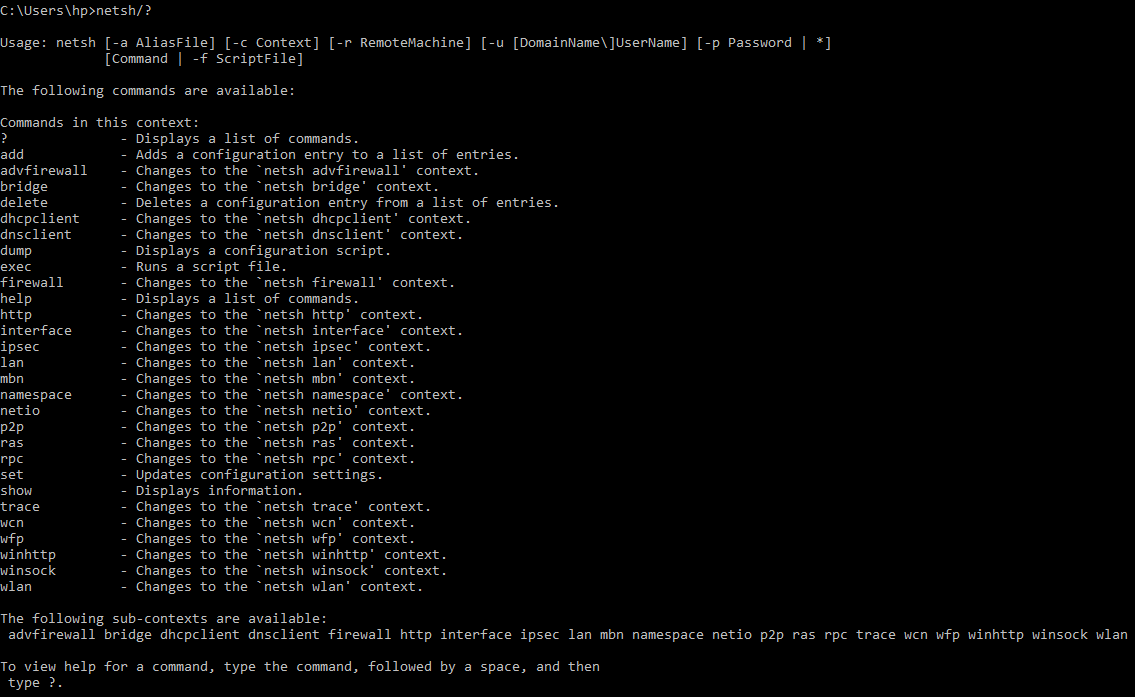
* In order to see all of the available netsh contexts, run **netsh /?**
* To see all of the commands available within a context, run **netsh contextname /?** Subcommands are available within certain commands.
* To view these, run netsh contextname **show /?**
* As an example, you can run the **netsh wlan** show drivers command to view all of the wireless network drivers on your device and their properties.

**Syntax:** netsh [-a AliasFile] [-c Context] [-r RemoteMachine] [-u [DomainName\]UserName] [-p Password | \*] [Command | -f ScriptFile]

**Parameters:**

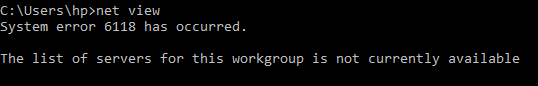
* **?:** Displays a list of commands.
* **add:** Adds a configuration entry to a list of entries.
* **advfirewall:** Changes to the `netsh advfirewall' context.
* **bridge:** Changes to the `netsh bridge' context.
* **delete:** Deletes a configuration entry from a list of entries.
* **dhcpclient:** Changes to the `netsh dhcpclient' context.
* **dnsclient:** Changes to the `netsh dnsclient' context.
* **dump:** Displays a configuration script.
* **exec:** Runs a script file.
* **firewall:** Changes to the `netsh firewall' context.
* **help:** Displays a list of commands.
* **http:** Changes to the `netsh http' context.
* **interface:** Changes to the `netsh interface' context.
* **ipsec:** Changes to the `netsh ipsec' context.
* **lan:** Changes to the `netsh lan' context.
* **mbn:** Changes to the `netsh mbn' context.
* **namespace:** Changes to the `netsh namespace' context.
* **netio:** Changes to the `netsh netio' context.
* **p2p:** Changes to the `netsh p2p' context.
* **ras:** Changes to the `netsh ras' context.
* **rpc:** Changes to the `netsh rpc' context.
* **set:** Updates configuration settings.
* **show:** Displays information.
* **trace:** Changes to the `netsh trace' context.
* **wcn:** Changes to the `netsh wcn' context.
* **wfp:** Changes to the `netsh wfp' context.
* **winhttp:** Changes to the `netsh winhttp' context.
* **winsock:** Changes to the `netsh winsock' context.
* **wlan:** Changes to the `netsh wlan' context.





1. **net view:** net view command is used for viewing devices connected to a network. There may be a time where you want to see what devices are connected to your network. This is where the net view command comes in. Simply run the net view command and after a short while you will be presented with a list of devices that are connected to the same network as you. The caveat with this command is that it may not show all of the devices connected to your network. It works well enough for private networks but will fail to identify devices such as smartphones and printers, and it can have trouble identifying devices running a different operating system to Windows. This simple command may work perfectly for you and your home network, but if not, you can always use the arp command we discussed earlier instead.

**Syntax:** net view



1. **route:** route tool displays the routing table that allows Windows 10 to understand the network and communicate with other devices and services. The tool also offers some options to modify and clear the table as needed. Like the arp tool, you typically do not have to worry about the routing table, but the command-line tool will come in handy when troubleshooting related problems.

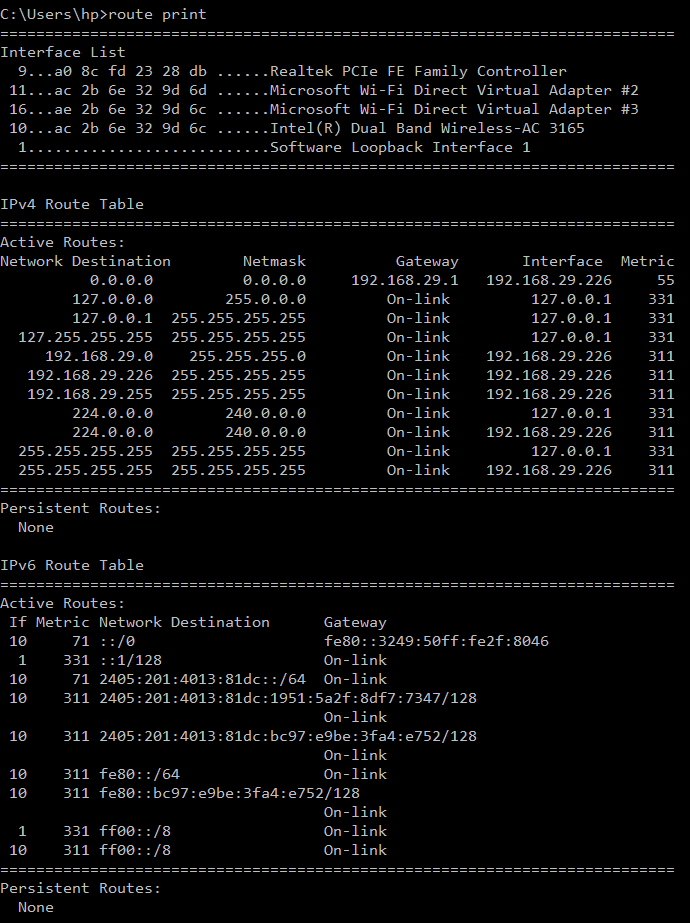
**Syntax:** ROUTE [-f] [-p] [-4|-6] command [destination] [MASK netmask] [gateway] [METRIC metric] [IF interface]

**Parameters:**

* **-f:** Clears the routing tables of all gateway entries. If this is used in conjunction with one of the commands, the tables are cleared prior to running the command.
* **-p:** When used with the ADD command, makes a route persistent across boots of the system. By default, routes are not preserved when the system is restarted. Ignored for all other commands, which always affect the appropriate persistent routes.
* **-4:** Force using IPv4.
* **-6:** Force using IPv6.

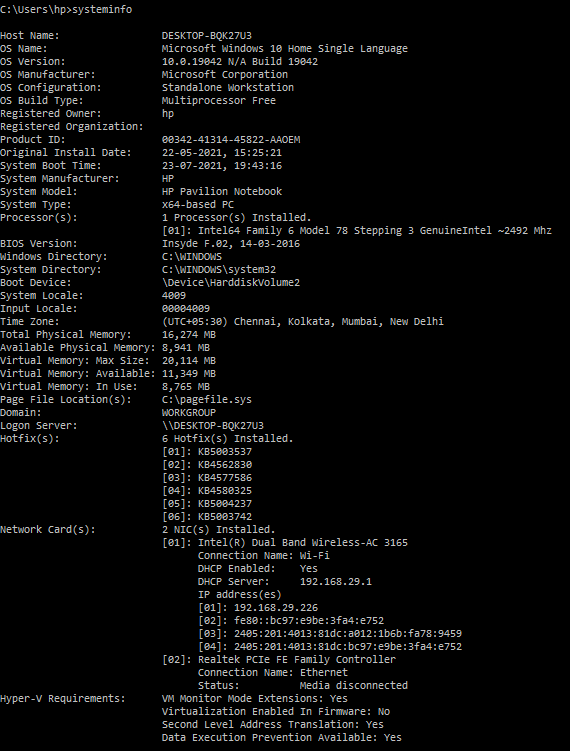
**Commands:**

* **PRINT:** Prints a route
* **ADD:** Adds a route
* **DELETE:** Deletes a route
* **CHANGE:** Modifies an existing route
* **destination:** Specifies the host.
* **MASK:** Specifies that the next parameter is the 'netmask' value.
* **netmask:** Specifies a subnet mask value for this route entry. If not specified, it defaults to 255.255.255.255.
* **gateway:** Specifies gateway.
* **interface:** the interface number for the specified route.
* **METRIC:** specifies the metric, ie. cost for the destination.



1. **systeminfo:** systeminfo command is used to display system information. If you need to know anything about the device you are using, be it details of the processor used, the version of Windows you are operating on, or what the boot device is configured as, you can find it all through the Windows GUI. But this command will poll your device and display the most important information in a clean, easy to read format.

**Syntax:** systeminfo



**Result:** Basic Networking commands has been executed successfully.