**TROUBLESHOOTING TOOLS**

**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(ayıklanır).

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 (köklenir) from multiple DHCP scopes or servers doling out(dağıtmak) 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 Command (pathping)**

**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(sonda yapar,araştırır) routers on the route path by limiting the number of hops individual packets are allowed to traverse(çaprazlayarak) 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.

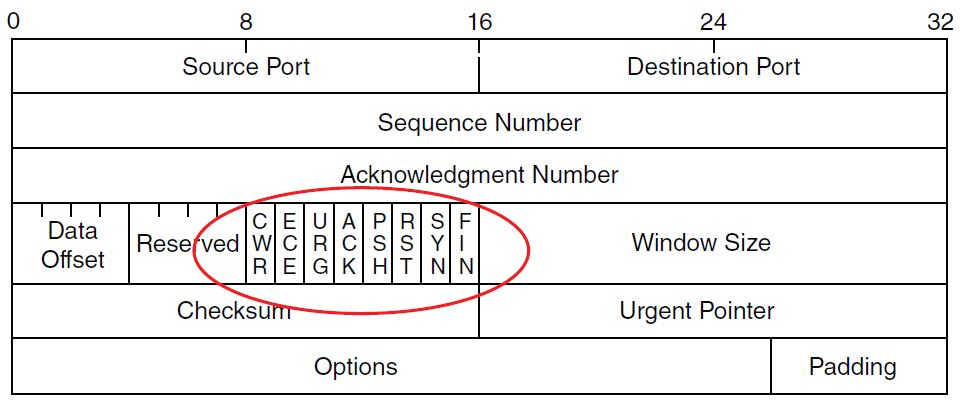
-w timeout Wait timeout milliseconds for each reply.

-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

**Using The route Command**

The biggest reason for manipulating the routing table on a server is to create a firewall. For instance, let’s say we’re running an Application layer firewall on a server located between the demilitarized zone (DMZ) and the internal network.

This scenario would mean the routing that’s happening on the server or hosts located in the DMZ wouldn’t be able to reach the internal network’s hosts and vice versa. To circumvent this problem, we would need to employ both static and default routing because running routing protocols on hosts and servers wouldn’t be a good solution for today’s networks.

To view the routing table on a Windows device, use the route print command, as shown below.

C:\Users\clarusway>route print

===========================================================================

Interface List

14...9c 5c 8e ce d9 c9 ......Intel(R) I211 Gigabit Network Connection

18...9c 5c 8e ce d9 ca ......Intel(R) Ethernet Connection (2) I219-V

15...76 c6 3b 00 62 86 ......Microsoft Wi-Fi Direct Virtual Adapter

8...76 c6 3b 00 6a 86 ......Microsoft Wi-Fi Direct Virtual Adapter #2

10...74 c6 3b 00 62 86 ......Broadcom 802.11ac Network Adapter

1...........................Software Loopback Interface 1

17...00 00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter

===========================================================================

IPv4 Route Table

===========================================================================

Active Routes:

Network Destination Netmask Gateway Interface Metric

0.0.0.0 0.0.0.0 192.168.1.1 192.168.1.22 35

127.0.0.0 255.0.0.0 On-link 127.0.0.1 331

127.0.0.1 255.255.255.255 On-link 127.0.0.1 331

127.255.255.255 255.255.255.255 On-link 127.0.0.1 331

192.168.1.0 255.255.255.0 On-link 192.168.1.22 291

192.168.1.22 255.255.255.255 On-link 192.168.1.22 291

192.168.1.255 255.255.255.255 On-link 192.168.1.22 291

224.0.0.0 240.0.0.0 On-link 127.0.0.1 331

224.0.0.0 240.0.0.0 On-link 192.168.1.22 291

255.255.255.255 255.255.255.255 On-link 127.0.0.1 331

255.255.255.255 255.255.255.255 On-link 192.168.1.22 291

===========================================================================

Persistent Routes:

None

IPv6 Route Table

===========================================================================

Active Routes:

If Metric Network Destination Gateway

17 331 ::/0 On-link

1 331 ::1/128 On-link

17 331 2001::/32 On-link

17 331 2001:0:2851:782c:148e:f3fd:6aff:55b8/128

On-link

10 291 fe80::/64 On-link

17 331 fe80::/64 On-link

17 331 fe80::148e:f3fd:6aff:55b8/128

On-link

10 291 fe80::19ac:8efb:2c6e:f512/128

On-link

1 331 ff00::/8 On-link

10 291 ff00::/8 On-link

17 331 ff00::/8 On-link

===========================================================================

Persistent Routes:

None

In this output, you can see that each of the routes was added automatically when the system booted up(açıldığında). To see all the options available with the route command, type route.

**route Command Options**

To add a route to your routing table, use the following syntax:

route [-f] [-p] [Command] [Destination] [mask Netmask] [Gateway] [metric Metric] [if Interface]

* **-f**: Using this command with any of the options like add, change, or delete will clear the routing table of all entries that aren’t host routes, the loopback network route or routes, and any multicast routes
* **-p**: If you use this with the add command, the individual route will be added to the Registry and then used to initialize the IP routing table whenever TCP/IP is started. Important to remember is that by default, the routes you’ve statically added won’t remain in the routing table the next time TCP/IP boots. And if you use -p with the print command, you’ll get shown a list of the persistent routes that are stored in the Registry location of HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes.

Now, let’s take a look at how and when you would use the route command. The below table shows the command options available and what they do when you are using the route command with them.

| **Command** | **Purpose** |
| --- | --- |
| add | Adds a route |
| change | Modifies an existing route |
| delete | Deletes a route (or routes) |
| print | Prints a route (or routes) |

Here’s a description of some other tasks you can accomplish via the rest of the command’s options:

* **Destination**: This will give you the network destination of a given route. If the host bits of the network address are set to 0, it will be depicted with the destination’s IP network address, an IP address for a specific host route, or the default route of 0.0.0.0.
* **mask netmask**: This will provide you with the subnet mask that’s associated with the destination network. The default destination subnet mask is 0.0.0.0, and typically you’ll see 255.255.255.255 representing a host route.
* **Gateway**: The gateway depends on the network address and subnet mask. It defines the next-hop IP address. For routes located on a local subnet, the gateway address maps directly to a particular interface. If the destination is on a remote network, the gateway IP address will direct packets to the router.
* **metric**: Metric refers to the cost of a given route from the sender to the receiver device, and it has a value between 1 and 9999. Devices use this value to choose the best, or most efficient, routes among those in its routing table—the route with the lowest value wins. This decision can also include factors like the number of hops and the speed, reliability, and available bandwidth of the path being considered.
* **if interface**: This tool depends on information from the gateway address and determines the interface index for the specific interface that needs to receive the data. You can get a list of interfaces along with their relevant interface indexes by typing the route print command.
* **/?**: Using this will allow you to view help at the command prompt.

**Some Examples of The route Command**

It is recommended that you spend some time practicing them on a nonproduction server.

* To display the entire IP routing table, type:

route print

* To add a default route with the default gateway address 192.168.10.1, type:

route add 0.0.0.0 mask 0.0.0.0 192.168.10.1

* To add a route to the destination 10.1.1.0 with the subnet mask 255.255.255.0 and the next-hop address 10.2.2.2, type:

route add 10.1.1.0 mask 255.255.255.0 10.2.2.2

* If you want to add a persistent route to the destination 10.100.0.0 with the subnet mask 255.255.0.0 and the next-hop address 10.2.0.1, type:

route -p add 10.100.0.0 mask 255.255.0.0 10.2.0.1

* If you want to delete the route to the destination 10.100.0.0 with the subnet mask 255.255.0.0, enter:

route delete 10.100.0.0 mask 255.255.0.0

* If you want to change the next-hop address of a route with the destination 10.100.0.0 and the subnet mask 255.255.0.0 from 10.2.0.1 to 10.7.0.5, type:

route change 10.100.0.0 mask 255.255.0.0 10.7.0.5