ARP and DNS

Both protocols do conversions of a sort, but the distinct difference is ARP is needed for packet transfers and DNS is not needed but makes things much easier.

ARP – Address Resolution Protocol

Converts IP addresses into MAC addresses

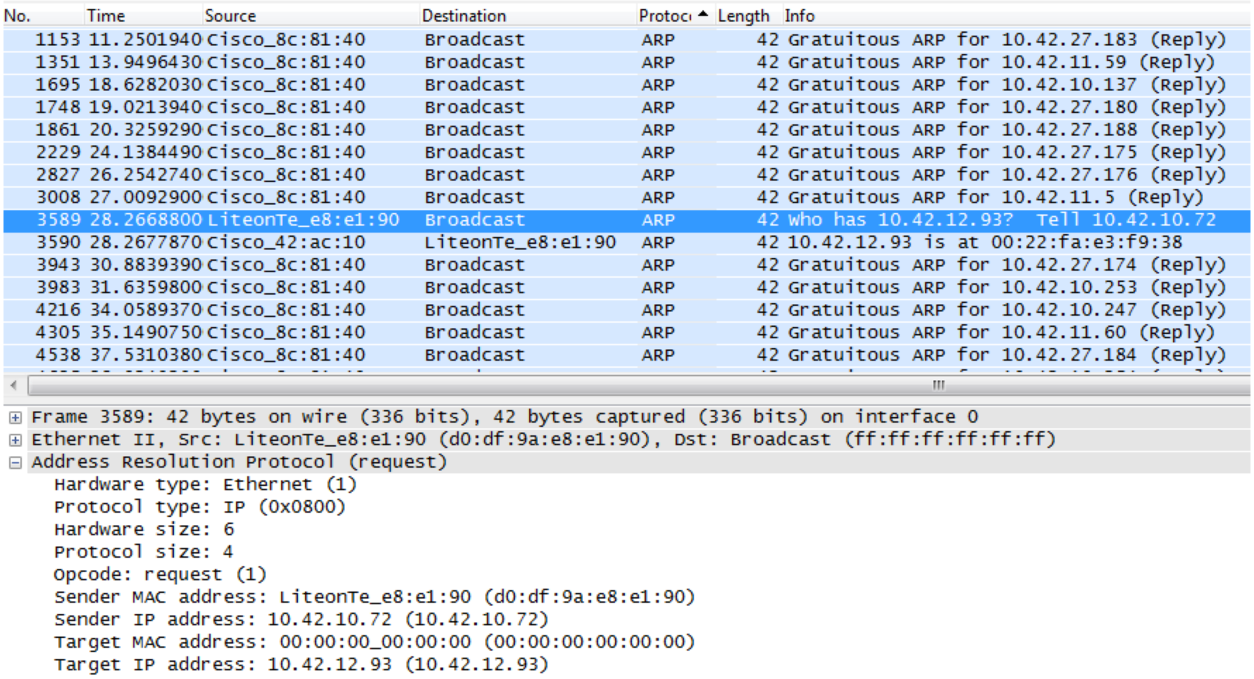
ARP entries are cached by network devices to save time, these cached entries make up a table

When an ARP reply is received the network host updates the protocol, no checks are done to ensure the machine that sent the reply is who it says it is.

There is no security on ARP which is where spoofing comes into play.

When a network device needs to send a packet to an IP, it sends out an ARP Request packet as follows:

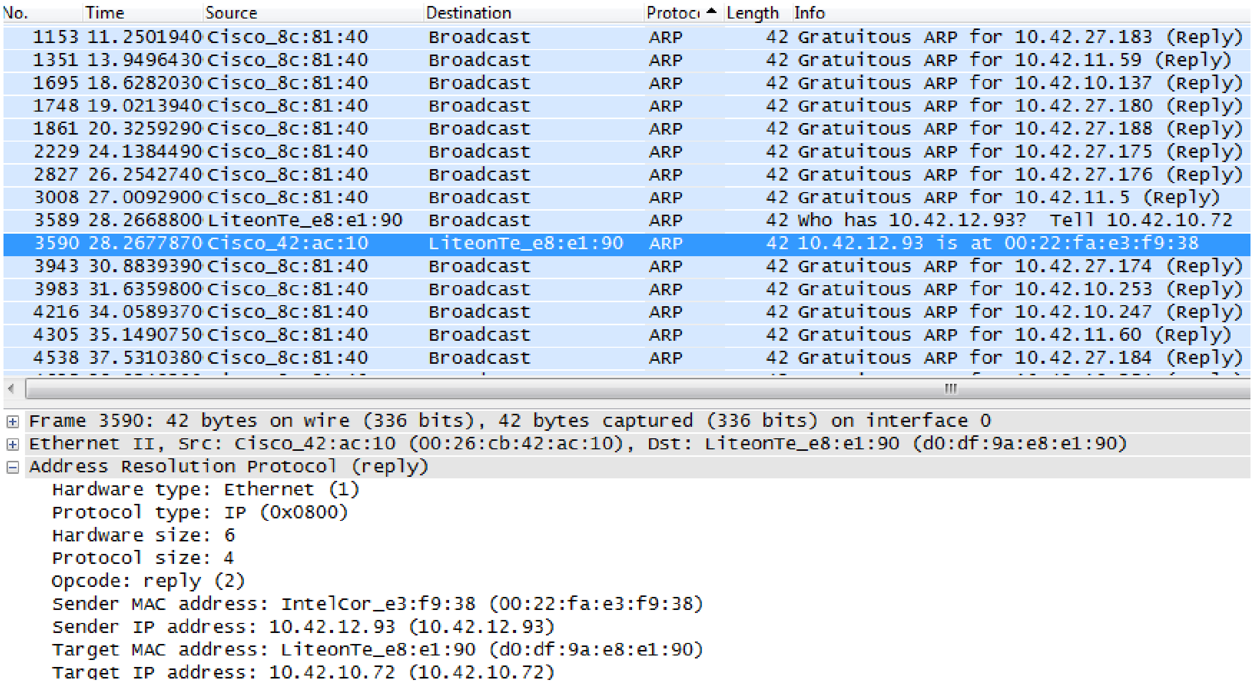
> Here is an example packet capture by Wireshark, as you can see the devices sends out an ARP request and asks who has 10.42.12.93 please tell 10.42.10.72.



```
No.   Time        Source              Destination         Protoc Length Info
 1153 11.2501940  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.183 (Reply)
 1351 13.9496430  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.11.59 (Reply)
 1695 18.6282030  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.10.137 (Reply)
 1748 19.0213940  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.180 (Reply)
 1861 20.3259290  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.188 (Reply)
 2229 24.1384490  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.175 (Reply)
 2827 26.2542740  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.176 (Reply)
 3008 27.0092900  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.11.5 (Reply)
 3589 28.2668800  LiteonTe_e8:e1:90   Broadcast           ARP        42 Who has 10.42.12.93?  Tell 10.42.10.72
 3590 28.2677870  Cisco_42:ac:10      LiteonTe_e8:e1:90   ARP        42 10.42.12.93 is at 00:22:fa:e3:f9:38
 3943 30.8839390  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.174 (Reply)
 3983 31.6359800  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.10.253 (Reply)
 4216 34.0589370  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.10.247 (Reply)
 4305 35.1490750  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.11.60 (Reply)
 4538 37.5310380  Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.184 (Reply)
```

```
⊞ Frame 3589: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
⊞ Ethernet II, Src: LiteonTe_e8:e1:90 (d0:df:9a:e8:e1:90), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: LiteonTe_e8:e1:90 (d0:df:9a:e8:e1:90)
    Sender IP address: 10.42.10.72 (10.42.10.72)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.42.12.93 (10.42.12.93)
```

And on this screen you can see that 10.42.12.93 responds with an ARP Reply back to 10.42.10.72 with its MAC address, 00:22:fa:e3:f9:38.



```
No.   Time        Source              Destination         Protoc▲ Length Info
 1153 11.2501940 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.183 (Reply)
 1351 13.9496430 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.11.59 (Reply)
 1695 18.6282030 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.10.137 (Reply)
 1748 19.0213940 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.180 (Reply)
 1861 20.3259290 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.188 (Reply)
 2229 24.1384490 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.175 (Reply)
 2827 26.2542740 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.176 (Reply)
 3008 27.0092900 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.11.5 (Reply)
 3589 28.2668800 LiteonTe_e8:e1:90   Broadcast           ARP        42 Who has 10.42.12.93?  Tell 10.42.10.72
 3590 28.2677870 Cisco_42:ac:10      LiteonTe_e8:e1:90   ARP        42 10.42.12.93 is at 00:22:fa:e3:f9:38
 3943 30.8839390 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.174 (Reply)
 3983 31.6359800 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.10.253 (Reply)
 4216 34.0589370 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.10.247 (Reply)
 4305 35.1490750 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.11.60 (Reply)
 4538 37.5310380 Cisco_8c:81:40      Broadcast           ARP        42 Gratuitous ARP for 10.42.27.184 (Reply)

⊞ Frame 3590: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
⊞ Ethernet II, Src: Cisco_42:ac:10 (00:26:cb:42:ac:10), Dst: LiteonTe_e8:e1:90 (d0:df:9a:e8:e1:90)
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_e3:f9:38 (00:22:fa:e3:f9:38)
    Sender IP address: 10.42.12.93 (10.42.12.93)
    Target MAC address: LiteonTe_e8:e1:90 (d0:df:9a:e8:e1:90)
    Target IP address: 10.42.10.72 (10.42.10.72)
```

Note: You will see a number of gratuitous ARP packets floating around out there, these are sent to tell other machines to update their ARP tables either because the remote machine has changed its IP or MAC address or when a device starts up or restarts.

ARP Spoofing / ARP Poisoning

Attack example:

Let's say there are two users on an internal network Adam and Bob, a third person Eve is trying to eavesdrop on the conversation Adam and Bob are having. Eve is running Wireshark on the same network Adam and Bob are on, as she watches she gets both their IP address and MAC address of their respective machines. She then sends an ARP Reply to Adam's machine saying to update his ARP table with Bob's IP address and Eve's MAC address. Adam's ARP table is updated and the next time he tries to send a message to Bob, Eve receives the message because her MAC address is now associated with Bob's IP address. Now if Eve wants to be really tricky she can then send the packet onto Bob so that Bob has no idea the intercept has occurred.

To write it out the attack would look like this

Adam (MAC = 00:11:22:33:44:55 IP = 192.168.1.2)

Bob (MAC = FF:EE:DD:CC:BB:AA IP = 192.168.1.3)

Eve (MAC = 55:44:33:22:11:00 IP = 192.168.1.4)

Prior to attack:

Adam's ARP Table:

192.168.1.3 -> FF:EE:DD:CC:BB:AA

Bob's ARP Table:

192.168.1.2 -> 00:11:22:33:44:55

Note: Adam's IP points to his MAC address and Bob's IP points to Bob's MAC address

After attack:

Adam's ARP Table:

192.168.1.3 -> 55:44:33:22:11:00

Bob's ARP Table:

192.168.1.2 -> 55:44:33:22:11:00

Note: Adam's and Bob's IP now points to Eve's MAC address, which means whenever Adam tries to send a packet to Bob or Bob tries to send a packet to Adam, Eve is the one getting it. This is the most basic form of a man in the middle attack.

Note: Not all ARP spoofing is illegitimate, if you got to a hotel or wireless hotspot that requires you to register what is actually happening to take you to that registration page is an ARP spoofing redirect. Your browser when opened asks for a webpage and the local router actually serves you back the registration page.

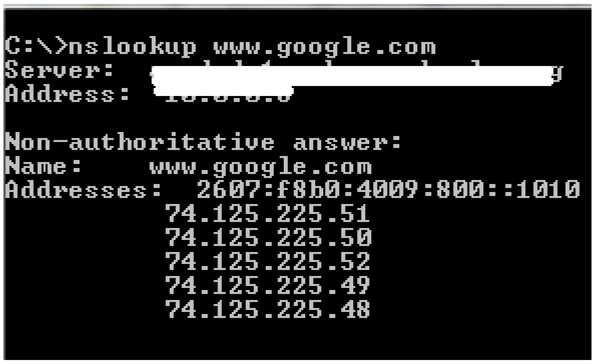
DNS – Domain Name Service

Converts hostnames into IP addresses and vice versa, it is made in a hierarchical system meaning that there are approximately 13 root DNS servers worldwide. There are many more smaller servers that rely on these root servers to get the job done.

A DNS Request:

When you type in www.google.com, a DNS request is generated because the OSI model we discussed earlier does not understand what www.google.com is, it only understands IP addresses and MAC addresses. So your computer needs to go out and grab a DNS entry for www.google.com, normally your local ISP has a DNS server so it first checks there. If that DNS server does not have an entry for Google (very unlikely) it will go to the next level up on the DNS tree and ask if it has an entry, and so on and so forth. If none of the lower DNS servers have an entry for whatever website you are going to it will ask one of the root servers which contain all of the web addresses available.

Here is what a DNS entry looks like, which can be found by doing an nslookup in a command prompt:



```
C:\>nslookup www.google.com
Server:   [redacted]
Address:

Non-authoritative answer:
Name:     www.google.com
Addresses:  2607:f8b0:4009:800::1010
          74.125.225.51
          74.125.225.50
          74.125.225.52
          74.125.225.49
          74.125.225.48
```

At the top you will see what DNS you are using (a popular one if your DNS is slow is Google's public DNS located at 8.8.8.8) and then you will see that we got a non-authoritative response from the server. This means that this entry has been cached on the local DNS server and the request was not sent out to an authoritative server for a response.

As you can see Google has quite a few web servers running right now, normally on a smaller site you will only get one maybe two IP's back for one entry.

Note: The nslookup program will do conversions either from IP to hostname or from hostname to IP.

A great analogy that is used for DNS is that DNS is the phone book of the Internet; it looks up the name of a domain and gives you back an IP. The problem is, by default, there is no security checks made in DNS. And again this is where either spoofing or DNS poisoning comes into play.

Attack Example:

Let's say you are on your local network again and somebody else has already done ARP poisoning between you and your DNS server using the method we described above. Now every time your computer makes a DNS request on port 53, it is sent to the attacker. The attacker then sends back the IP address of a malware site or something of the like and now when you try to go to Google you will get directed elsewhere. You can see how this can be very dangerous for the normal user.

This problem could be mitigated by using a more secure DNS system, and one has been designed but not really implemented yet. It is called DNSSEC which stands for Domain Name System Security Extensions. In this system the DNS server actually digitally signs each DNS response it gives meaning that the computer that makes the request can confirm this came from the correct DNS server.

Fake antivirus programs have been doing this for a while now by placing rootkits on infected machines, these rootkits do DNS poisoning even before the OS starts up and make it nearly impossible to go anywhere online without being redirected. This problem would not be mitigated by DNSSEC as it is all being done internally on the local machine, no network request are being made for this to happen.