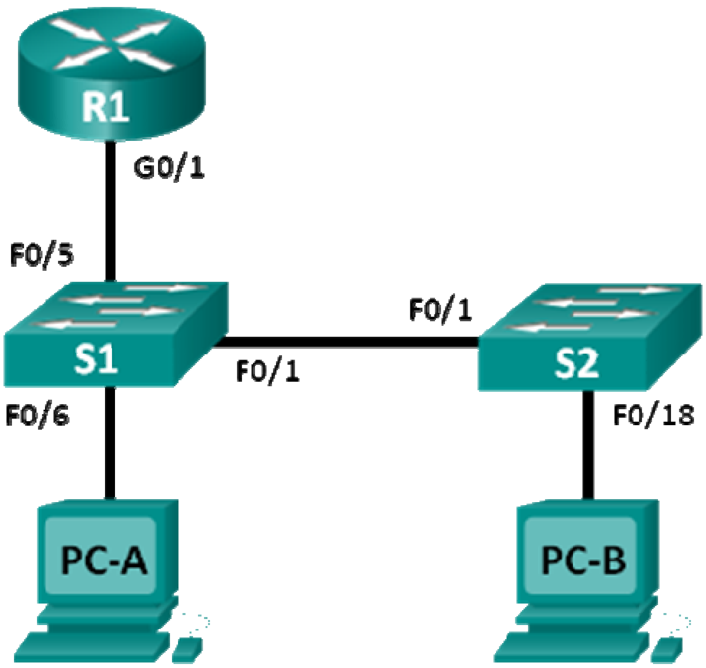
Lab – Observing ARP with the Windows CLI, IOS CLI, and Wireshark (Instructor Version)

Instructor Note: Red font color or Gray highlights indicate text that appears in the instructor copy only.

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
S1	VLAN 1	192.168.1.11	255.255.255.0	192.168.1.1
S2	VLAN 1	192.168.1.12	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.1.2	255.255.255.0	192.168.1.1

Objectives

Part 1: Build and Configure the Network

Part 2: Use the Windows ARP Command

Part 3: Use the IOS Show ARP Command

Part 4: Use Wireshark to Examine ARP Exchanges

Background / Scenario

The Address Resolution Protocol (ARP) is used by TCP/IP to map a Layer 3 IP address to a Layer 2 MAC address. When a frame is placed on the network, it must have a destination MAC address. To dynamically discover the MAC address for the destination device, an ARP request is broadcast on the LAN. The device that contains the destination IP address responds, and the MAC address is recorded in the ARP cache. Every device on the LAN keeps its own ARP cache, or small area in RAM that holds ARP results. An ARP cache timer removes ARP entries that have not been used for a certain period of time.

ARP is an excellent example of performance tradeoff. With no cache, ARP must continually request address translations each time a frame is placed on the network. This adds latency to the communication and could congest the LAN. Conversely, unlimited hold times could cause errors with devices that leave the network or change the Layer 3 address.

A network administrator should be aware of ARP, but may not interact with the protocol on a regular basis. ARP is a protocol that enables network devices to communicate with the TCP/IP protocol. Without ARP, there is no efficient method to build the datagram Layer 2 destination address. Also, ARP is a potential security risk. ARP spoofing, or ARP poisoning, is a technique used by an attacker to inject the wrong MAC address association in a network. An attacker forges the MAC address of a device, and frames are sent to the wrong destination. Manually configuring static ARP associations is one way to prevent ARP spoofing. Finally, an authorized MAC address list may be configured on Cisco devices to restrict network access to only approved devices.

In this lab, you will use the ARP commands in both Windows and Cisco routers to display the ARP table. You will also clear the ARP cache and add static ARP entries.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Instructor Note: Some of the ARP commands in Windows Vista or later operating systems will require administrator privileges.

Required Resources

- 1 Router (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term and Wireshark installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Note: The Fast Ethernet interfaces on Cisco 2960 switches are autosensing and an Ethernet straight-through cable may be used between switches S1 and S2. If using another Cisco switch model, it may be necessary to use an Ethernet crossover cable.

Part 1: Build and Configure the Network

Step 1: Cable the network according to the topology.

Step 2: Configure the IP addresses for the devices according to the Addressing Table.

Step 3: Verify network connectivity by pinging all the devices from PC-B.

Part 2: Use the Windows ARP Command

The **arp** command allows the user to view and modify the ARP cache in Windows. You access this command from the Windows command prompt.

Step 1: Display the ARP cache.

a. Open a command window on PC-A and type **arp**.

```
C:\Users\User1> 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                                    .... Displays the arp table.
```


b. Examine the output.

What command would be used to display all entries in the ARP cache?

____

arp –a

What command would be used to delete all ARP cache entries (flush ARP cache)?

____

arp –d *

What command would be used to delete the ARP cache entry for 192.168.1.11?

____

arp –d 192.168.1.11

c. Type **arp –a** to display the ARP table.

```
C:\Users\User1> arp –a

Interface: 192.168.1.3 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.1           d4-8c-b5-ce-a0-c1     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

Note: The ARP table is empty if you use Windows XP (as displayed below).

```
C:\Documents and Settings\User1> arp -a
No ARP Entries Found.
```

d. Ping from PC-A to PC-B to dynamically add entries in the ARP cache.

```
C:\Documents and Settings\User1> ping 192.168.1.2

Interface: 192.168.1.3 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.2           00-50-56-be-f6-db     dynamic
```

What is the physical address for the host with IP address of 192.168.1.2?

____

00-50-56-be-f6-db

Step 2: Adjust entries in the ARP cache manually.

To delete entries in ARP cache, issue the command **arp –d {inet-addr | *}**. Addresses can be deleted individually by specifying the IP address, or all entries can be deleted with the wildcard *****.

Verify that the ARP cache contains the following entries: the R1 G0/1 default gateway (192.168.1.1), PC-B (192.168.1.2) and both switches (192.168.1.11 and 192.168.1.12).

a. From PC-A, ping all the addresses in the Address Table.

b. Verify that all the addresses have been added to the ARP cache. If the address is not in ARP cache, ping the destination address and verify that the address was added to the ARP cache.

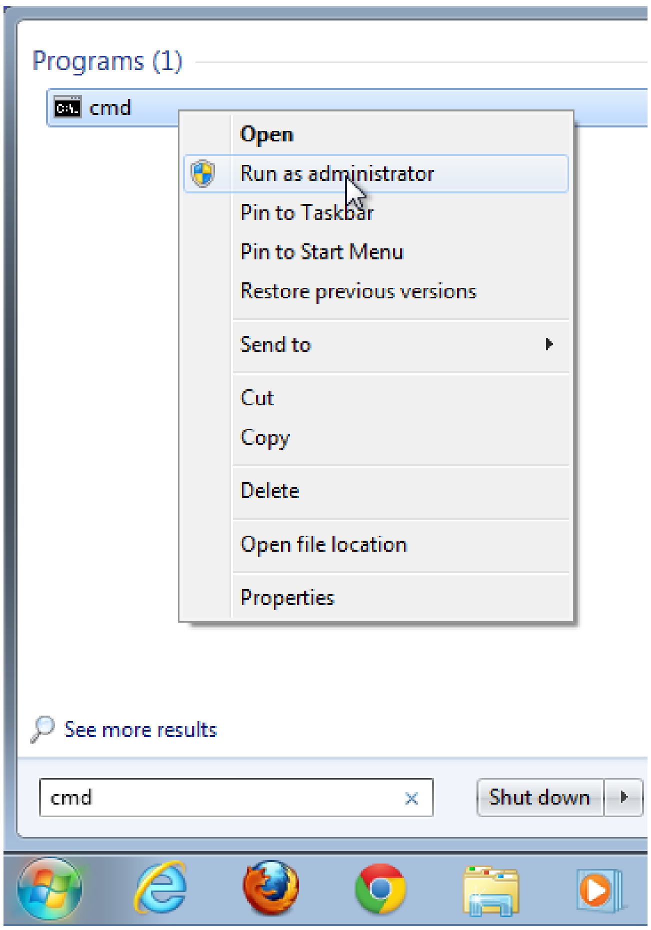
```
C:\Users\User1> arp –a
```



```
Interface: 192.168.1.3 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.1           d4-8c-b5-ce-a0-c1     dynamic
  192.168.1.2           00-50-56-be-f6-db     dynamic
  192.168.1.11          0c-d9-96-e8-8a-40     dynamic
  192.168.1.12          0c-d9-96-d2-40-40     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

c. As an administrator, access the command prompt. Click the **Start** icon, and in the *Search programs and file* box, type **cmd**. When the **cmd** icon appears, right-click the icon and select **Run as administrator**. Click **Yes** to allow this program to make changes.

Note: For Windows XP users, it is not necessary to have administrator privileges to modify ARP cache entries.



d. In the Administrator command prompt window, type **arp –d ***. This command deletes all the ARP cache entries. Verify that all the ARP cache entries are deleted by typing **arp –a** at the command prompt.

```
C:\windows\system32> arp –d *
C:\windows\system32> arp –a
```

```
No ARP Entries Found.
```

e. Wait a few minutes. The Neighbor Discovery protocol starts to populate the ARP cache again.

```
C:\Users\User1> arp –a

Interface: 192.168.1.3 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
```

Note: The Neighbor Discovery protocol is not implemented in Windows XP.

f. From PC-A, ping PC-B (192.168.1.2) and the switches (192.168.1.11 and 192.168.1.12) to add the ARP entries. Verify that the ARP entries have been added to the cache.

```
C:\Users\User1> arp –a

Interface: 192.168.1.3 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.2           00-50-56-be-f6-db     dynamic
  192.168.1.11          0c-d9-96-e8-8a-40     dynamic
  192.168.1.12          0c-d9-96-d2-40-40     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
```

g. Record the physical address for switch S2.

____________________________________ Answers will vary. 0c-d9-96-d2-40-40 in this case.

h. Delete a specific ARP cache entry by typing **arp –d *inet-addr***. At the command prompt, type **arp -d 192.168.1.12** to delete the ARP entry for S2.

```
C:\windows\system32> arp –d 192.168.1.12
```

i. Type **arp –a** to verify that the ARP entry for S2 has been removed from the ARP cache.

```
C:\Users\User1> arp –a

Interface: 192.168.1.3 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.2           00-50-56-be-f6-db     dynamic
  192.168.1.11          0c-d9-96-e8-8a-40     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
```

j. You can add a specific ARP cache entry by typing **arp –s *inet_addr mac_addr***. The IP address and MAC address for S2 will be used in this example. Use the MAC address recorded in step g.

```
C:\windows\system32> arp –s 192.168.1.12 0c-d9-96-d2-40-40
```

k. Verify that the ARP entry for S2 has been added to the cache.

Part 3: Use the IOS show arp Command

The Cisco IOS can also display the ARP cache on routers and switches with the **show arp** or **show ip arp** command.

Step 1: Display ARP entries on router R1.

```
R1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1             -   d48c.b5ce.a0c1  ARPA   GigabitEthernet0/1
```



```
Internet  192.168.1.2             0   0050.56be.f6db  ARPA   GigabitEthernet0/1
Internet  192.168.1.3             0   0050.56be.768c  ARPA   GigabitEthernet0/1
R1#
```

Notice there is no Age (-) for the first entry, router interface G0/1 (the LAN default gateway). The Age is the number of minutes (min) that the entry has been in ARP cache and is incremented for the other entries. The Neighbor Discovery protocol populates the PC-A and PC-B IP and MAC address ARP entries.

Step 2: Add ARP entries on router R1.

You can add ARP entries to the ARP table of the router by pinging other devices.

a. Ping switch S1.

```
R1# ping 192.168.1.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.11, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms
```

b. Verify that an ARP entry for switch S1 has been added to the ARP table of R1.

```
R1# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1             -   d48c.b5ce.a0c1  ARPA   GigabitEthernet0/1
Internet  192.168.1.2             6   0050.56be.f6db  ARPA   GigabitEthernet0/1
Internet  192.168.1.3             6   0050.56be.768c  ARPA   GigabitEthernet0/1
Internet  192.168.1.11            0   0cd9.96e8.8a40  ARPA   GigabitEthernet0/1
R1#
```

Step 3: Display ARP entries on switch S1.

```
S1# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1            46   d48c.b5ce.a0c1  ARPA   Vlan1
Internet  192.168.1.2             8   0050.56be.f6db  ARPA   Vlan1
Internet  192.168.1.3             8   0050.56be.768c  ARPA   Vlan1
Internet  192.168.1.11            -   0cd9.96e8.8a40  ARPA   Vlan1
S1#
```

Step 4: Add ARP entries on switch S1.

By pinging other devices, ARP entries can also be added to the ARP table of the switch.

a. From switch S1, ping switch S2.

```
S1# ping 192.168.1.12
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.12, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/8 ms
```

b. Verify that the ARP entry for switch S2 has been added to ARP table of S1.

```
S1# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
```

```
Internet  192.168.1.1            5   d48c.b5ce.a0c1  ARPA   Vlan1
Internet  192.168.1.2           11   0050.56be.f6db  ARPA   Vlan1
Internet  192.168.1.3           11   0050.56be.768c  ARPA   Vlan1
Internet  192.168.1.11           -   0cd9.96e8.8a40  ARPA   Vlan1
Internet  192.168.1.12           2   0cd9.96d2.4040  ARPA   Vlan1
S1#
```

Part 4: Use Wireshark to Examine ARP Exchanges

In Part 4, you will examine ARP exchanges by using Wireshark to capture and evaluate the ARP exchange. You will also examine network latency caused by ARP exchanges between devices.

Step 1: Configure Wireshark for packet captures.

a. Start Wireshark.

b. Choose the network interface to use for capturing the ARP exchanges.

Step 2: Capture and evaluate ARP communications.

a. Start capturing packets in Wireshark. Use the filter to display only ARP packets.

b. Flush the ARP cache by typing the **arp –d *** command at the command prompt.

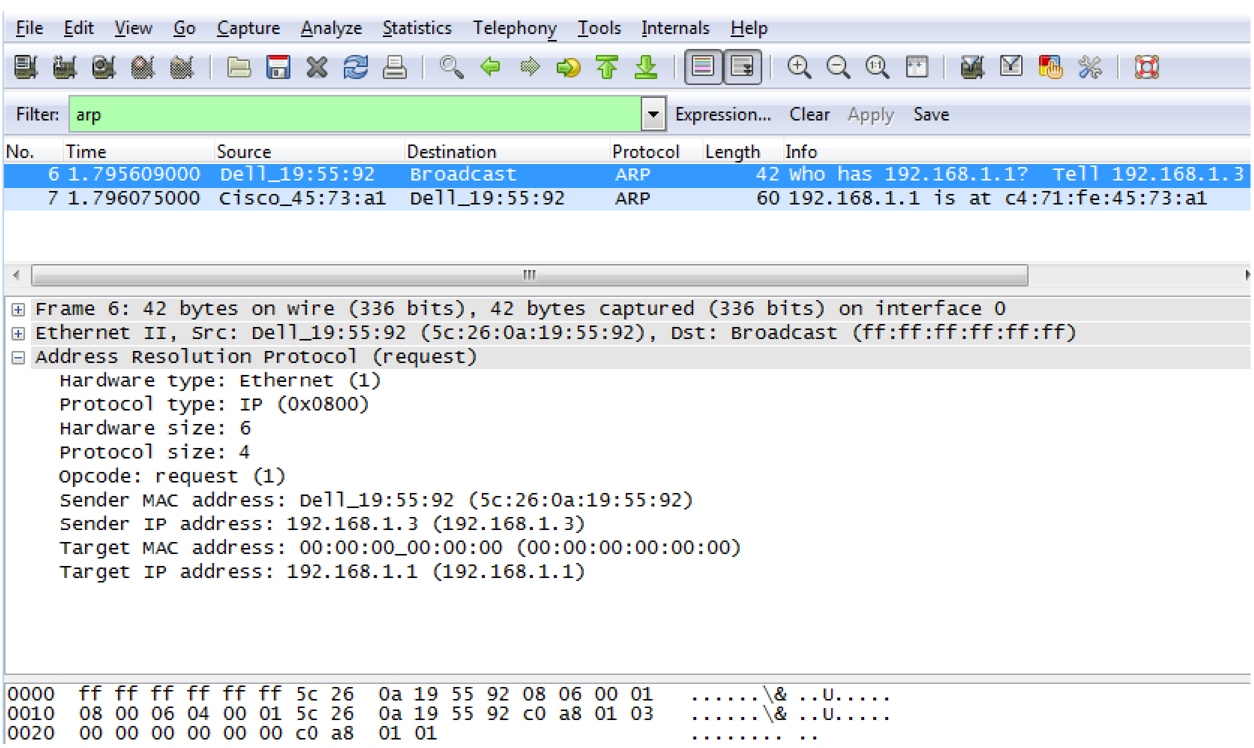
c. Verify that the ARP cache has been cleared.

d. Send a ping to the default gateway, using the **ping 192.168.1.1** command.

e. Stop the Wireshark capture after pinging to the default gateway is finished.

f. Examine the Wireshark captures for the ARP exchanges in the packet details pane.

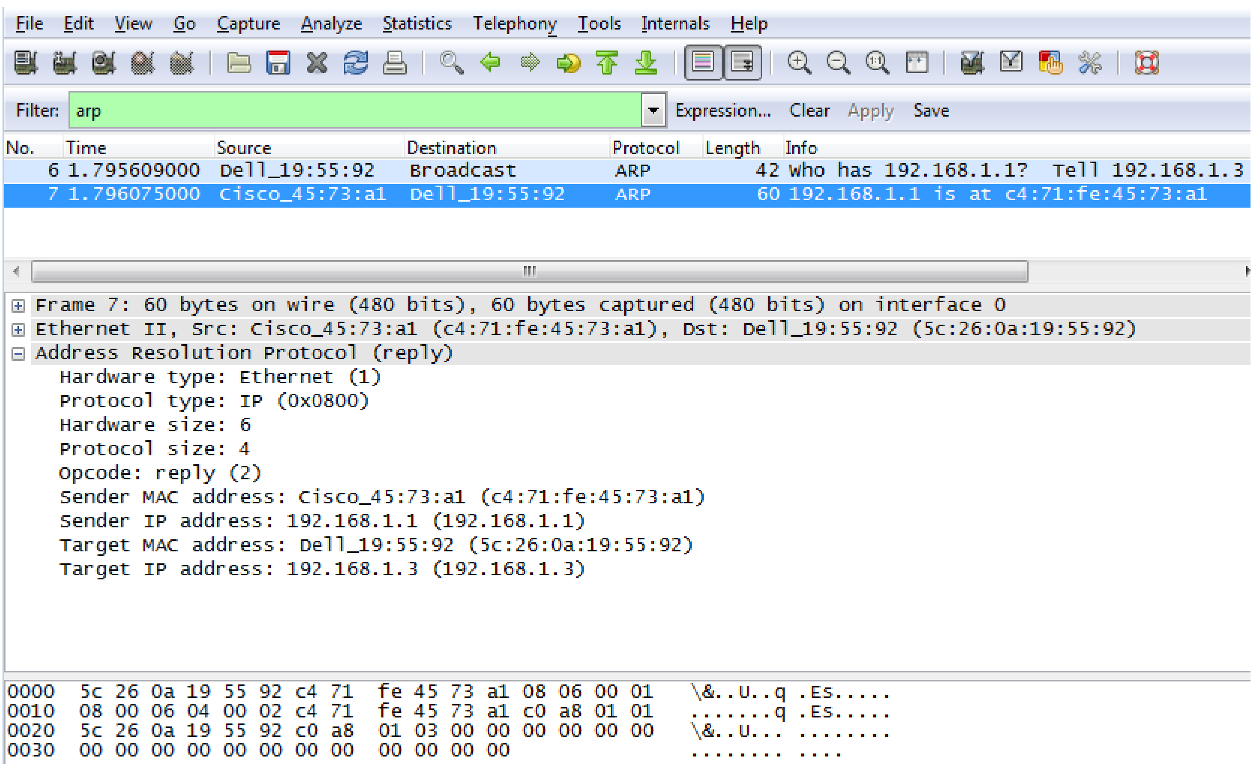
What was the first ARP packet? ____________________________ ARP request



Fill in the following table with information about your first captured ARP packet.

Field	Value
Sender MAC address	5C:26:0A:19:55:92 - Answers will vary.
Sender IP address	192.168.1.3
Target MAC address	00:00:00:00:00:00
Target IP address	192.168.1.1

What was the second ARP packet? ______________________________ ARP reply



Fill in the following table with information about your second captured ARP packet.

Field	Value
Sender MAC address	C4:71:FE:45:73:A1 - Answers will vary.
Sender IP address	192.168.1.1
Target MAC address	5C:26:0A:19:55:92 - Answers will vary.
Target IP address	192.168.1.3

Step 3: Examine network latency caused by ARP.

a. Clear the ARP entries on PC-A.

b. Start a Wireshark capture.

c. Ping switch S2 (192.168.1.12). The ping should be successful after the first echo request.

   Note: If all the pings were successful, S1 should be reloaded to observe network latency with ARP.

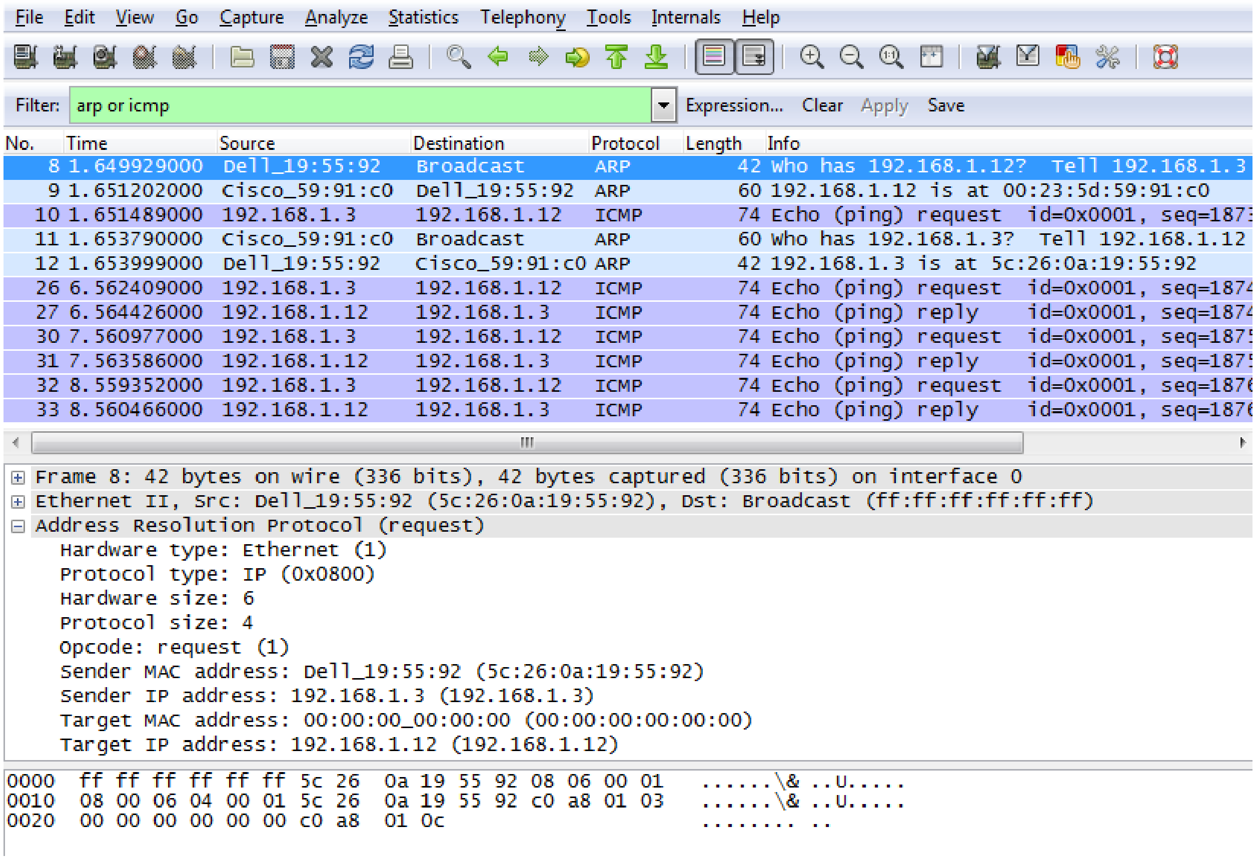
```
C:\Users\User1> ping 192.168.1.12
Request timed out.
Reply from 192.168.1.12: bytes=32 time=2ms TTL=255
Reply from 192.168.1.12: bytes=32 time=2ms TTL=255
Reply from 192.168.1.12: bytes=32 time=2ms TTL=255

Ping statistics for 192.168.1.12:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms
```

d. Stop the Wireshark capture after the pinging is finished. Use the Wireshark filter to display only ARP and ICMP outputs. In Wireshark, type **arp or icmp** in the **Filter:** entry area.

e. Examine the Wireshark capture. In this example, frame 10 is the first ICMP request sent by PC-A to S1. Because there is no ARP entry for S1, an ARP request was sent to the management IP address of S1 asking for the MAC address. During the ARP exchanges, the echo request did not receive a reply before the request was timed out. (frames 8 – 12)

   After the ARP entry for S1 was added to the ARP cache, the last three ICMP exchanges were successful, as displayed in frames 26, 27 and 30 – 33.

   As displayed in the Wireshark capture, ARP is an excellent example of performance tradeoff. With no cache, ARP must continually request address translations each time a frame is placed on the network. This adds latency to the communication and could congest the LAN.



Reflection

1. How and when are static ARP entries removed?

   They are deleted manually.

2. Why do you want to add static ARP entries in the cache?

   A static ARP entry can mitigate ARP spoofing or poisoning in the network.

3. If ARP requests can cause network latency, why is it a bad idea to have unlimited hold times for ARP entries?

   Unlimited hold times could cause errors with devices that leave the network or change the Layer 3 address.

Router Interface Summary Table

Router Interface Summary				
Router Model	**Ethernet Interface #1**	**Ethernet Interface #2**	**Serial Interface #1**	**Serial Interface #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs

Router R1

```
R1#show run
Building configuration...

Current configuration : 1165 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
memory-size iomem 15
!
!
!
```



```
!
!
!
!
ip cef
no ipv6 cef
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
!
!
```



```
!
control-plane
!
!
!
line con 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Switch S1

```
S1#show run
Building configuration...

Current configuration : 1305 bytes
!
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname S1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
!
!
!
!
!
!
```



```
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
```

```
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 192.168.1.11 255.255.255.0
!
ip http server
ip http secure-server
!
!
!
line con 0
line vty 5 15
!
end
```

Switch S2

```
S2#show run
Building configuration...

Current configuration : 1313 bytes
!
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname S2
!
boot-start-marker
boot-end-marker
!
!
```



```
no aaa new-model
system mtu routing 1500
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
```

```
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 192.168.1.12 255.255.255.0
!
ip http server
ip http secure-server
!
!
line con 0
line vty 5 15
!
end
```