CSC465 – Computer Networks

Dr. J. Harrison

These slides were produced almost entirely from material by Behrouz Forouzan for the text "TCP/IP Protocol Suite (2nd Edition)", McGraw Hill Publisher

Chapter 7

ARP and RARP

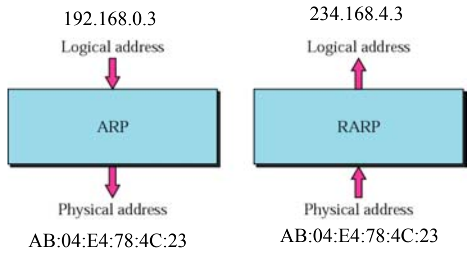
Addresses Revisited

- Logical Address
  - Internet address
  - Jurisdiction is universal; unique universally
  - Usually implemented in software
- Physical addresses
  - Packets pass through physical networks to reach hosts and routers
  - Hosts and routers recognized by physical addresses
  - Jurisdiction is a local (individual) network
  - Usually (not always) implemented in hardware

Address Mapping

- We need to map physical to logical AND logical to physical
- Static Mapping: use a table
- Dynamic Mapping:
  - use a protocol to consult network
- Address Resolution Protocol (ARP)
  - Maps logical address to physical
- Reverse Address Resolution Protocol (RARP)
  - Maps physical address to logical

ARP and RARP



192.168.0.3

Logical address

ARP

Physical address

AB:04:E4:78:4C:23

234.168.4.3

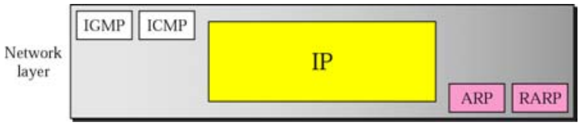
Logical address

RARP

Physical address

AB:04:E4:78:4C:23

Position of ARP and RARP in TCP/IP protocol suite



Network layer

IGMP

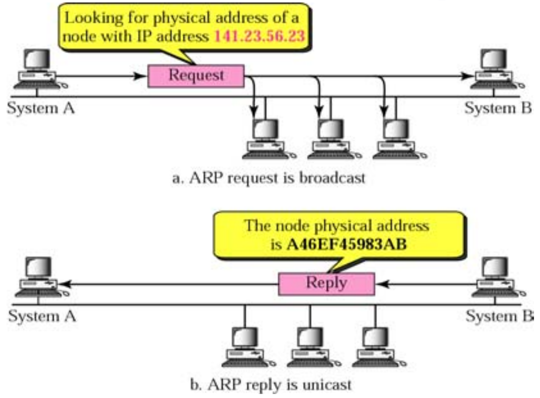
ICMP

IP

ARP

RARP

ARP operation


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A46EF45983AB
Reply
System A
System B

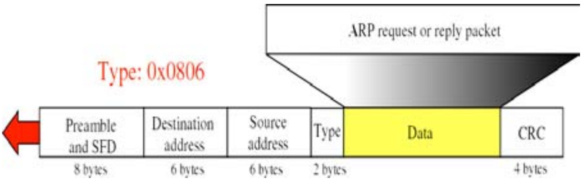
a. ARP request is broadcast

b. ARP reply is unicast

ARP packet

<table>
<tr><td colspan="2">Hardware Type
Ethernet → 1</td><td>Protocol Type
IPv4 → 0x0800</td></tr>
<tr><td>Hardware length</td><td>Protocol IPv4 length → 4</td><td>Operation
Request 1, Reply 2</td></tr>
<tr><td>Ethernet → 6 bytes</td><td>Sender hardware address
(For example, 6 bytes for Ethernet)</td><td>CC:00:FF:FF:EE:EE</td></tr>
<tr><td></td><td>Sender protocol address
(For example, 4 bytes for IP)</td><td>192.168.0.5</td></tr>
<tr><td colspan="3">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)</td></tr>
<tr><td colspan="3">Target protocol address
(For example, 4 bytes for IP)</td></tr>
</table>

Encapsulation of ARP packet



ARP Process

1. Sender has IP address; needs physical address
2. IP asks ARP to create ARP request
   - …using senders physical & IP addresses and recipients IP
3. ARP uses DL layer; Encapulates with:
   - Physical address of sender as source
   - Physical broadcast address as the destination
4. All host receive (then drop except targeted host)
5. Target replies with IP address
6. Sender then unicasts back ARP response

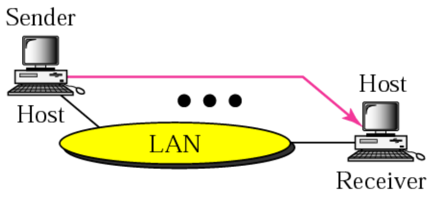
Note

An ARP request is broadcast;

an ARP reply is unicast.

Uses of ARP

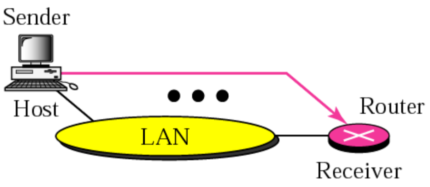
Target IP address:
Destination address in the IP datagram



Case 1. A host has a packet to send to another host on the same network.

Uses of ARP (con't)

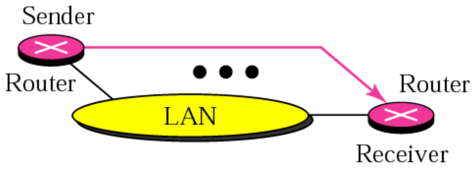
Target IP address:
IP address of a router



Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.

Uses of ARP (con't)

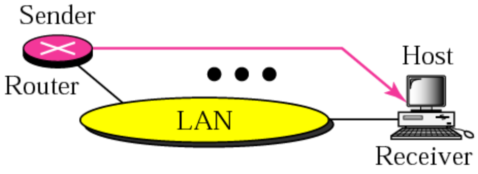
Target IP address:
IP address of the appropriate router found in the routing table



Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.

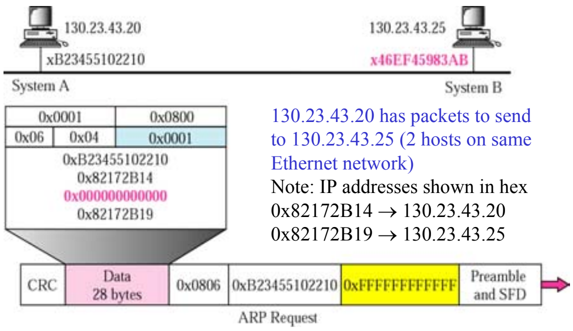
Uses of ARP (con't)

Target IP address:
Destination address in the IP datagram



Case 4. A router receives a packet to be sent to a host on the same network.

Example 1: ARP Request



130.23.43.20 has packets to send to 130.23.43.25 (2 hosts on same Ethernet network)

Note: IP addresses shown in hex
0x82172B14 → 130.23.43.20
0x82172B19 → 130.23.43.25

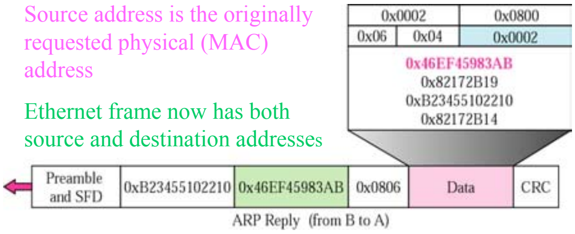
Example 1 (Continued): ARP REPLY

Operation is now "reply"

Note ordering of fields.

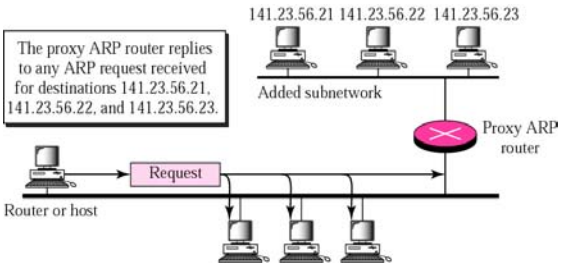
Source address is the originally requested physical (MAC) address

Ethernet frame now has both source and destination addresses



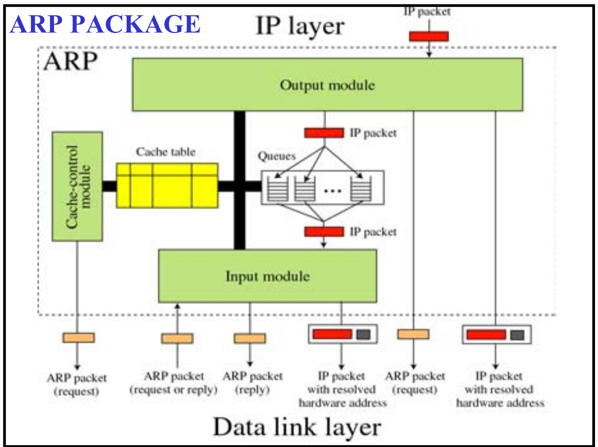
Proxy ARP

Use ARP to emulate a subnet



The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.

ARP PACKAGE



ARP Cache Table

- Typically more than one IP datagram to same host
- Inefficient to use ARP for every datagram
- Cache table is used
- Packets for same destination are enqueued in same queue
- Number of attempts to resolve are recorded
- Time-to-live recorded for cache entry

Input Module

- Sleep until ARP packet (request or reply) arrives
- If request, simply reply
- If “reply” (solicited or not), check cache:
- If found in cache:
  - Update ARP entry
  - Send any queued packets
- If not found in cache:
  - Create cache entry
  - Add entry to table

Output Module

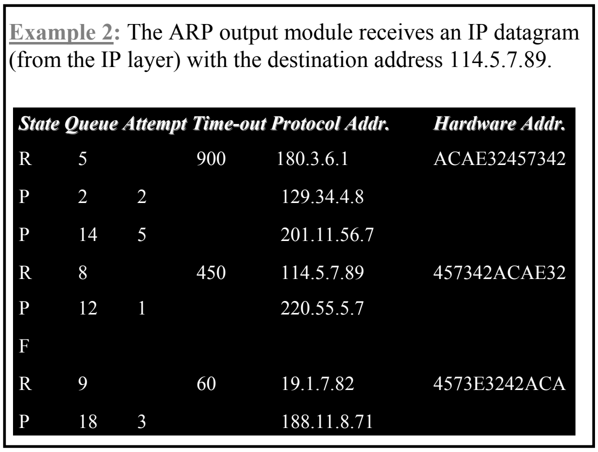
- Sleep until IP packet received from IP (layer 3) software
- Check cache table for an entry for IP dest
- If “found”:
  - If resolved, send packet using DL (layer 2) address
  - If pending, enqueue packet to correct queue
- If not found in cache:
  - Create cache entry with state=Pending, Attempts=1
  - Create queue and enqueue packet
  - Send ARP request

Cache-Control Module

- Sleep until periodic time matures
- Consider all cache entries
- If “Pending”
  - ++attempts, send another ARP request
- If too many attempts,
  - change state to free and destroy queue
- If state “Resolved”
  - Decrement value of time-out by elasped time
  - If time elapsed, change state to free; destroy queue

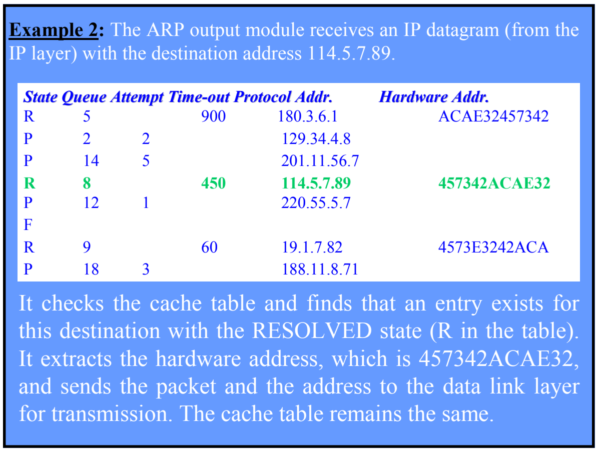
Original ARP Cache Table

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	



Example 2: The ARP output module receives an IP datagram (from the IP layer) with the destination address 114.5.7.89.

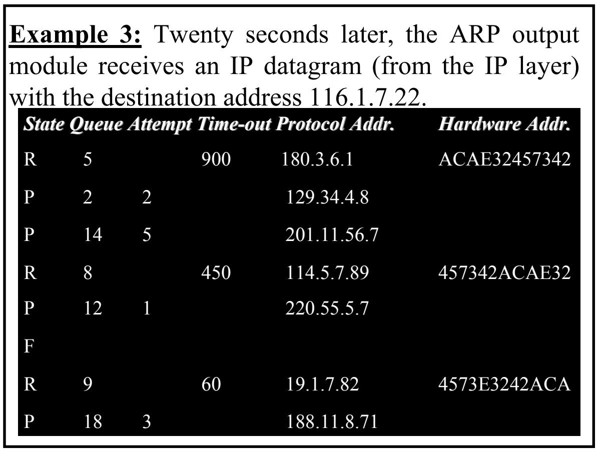
State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	



Example 2: The ARP output module receives an IP datagram (from the IP layer) with the destination address 114.5.7.89.

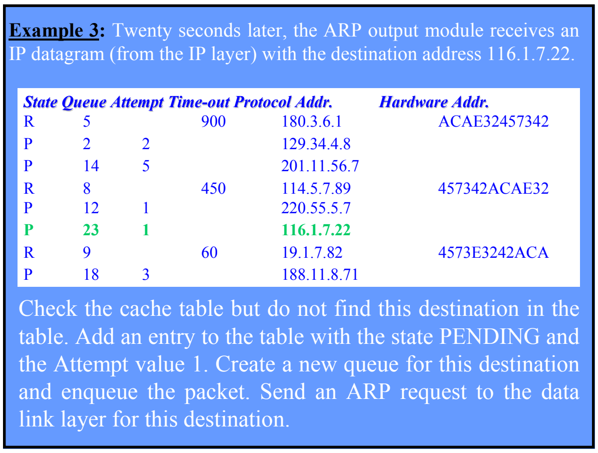
State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

It checks the cache table and finds that an entry exists for this destination with the RESOLVED state (R in the table). It extracts the hardware address, which is 457342ACAE32, and sends the packet and the address to the data link layer for transmission. The cache table remains the same.



Example 3: Twenty seconds later, the ARP output module receives an IP datagram (from the IP layer) with the destination address 116.1.7.22.

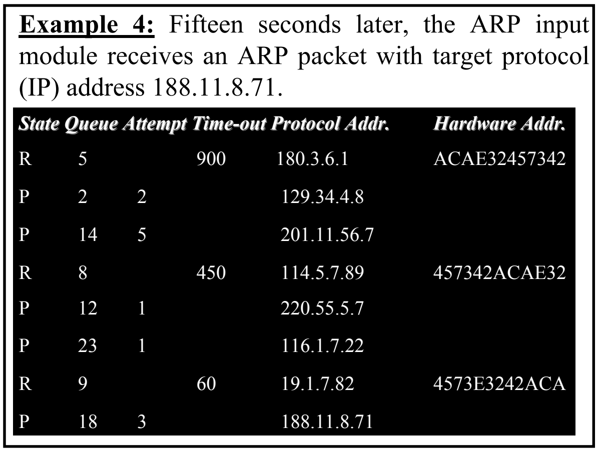
State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	



Example 3: Twenty seconds later, the ARP output module receives an IP datagram (from the IP layer) with the destination address 116.1.7.22.

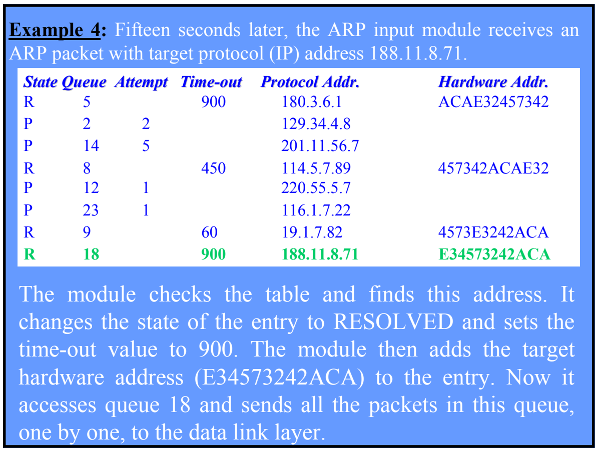
State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

Check the cache table but do not find this destination in the table. Add an entry to the table with the state PENDING and the Attempt value 1. Create a new queue for this destination and enqueue the packet. Send an ARP request to the data link layer for this destination.



Example 4: Fifteen seconds later, the ARP input module receives an ARP packet with target protocol (IP) address 188.11.8.71.

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	



Example 4: Fifteen seconds later, the ARP input module receives an ARP packet with target protocol (IP) address 188.11.8.71.

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
R	18		900	188.11.8.71	E34573242ACA

The module checks the table and finds this address. It changes the state of the entry to RESOLVED and sets the time-out value to 900. The module then adds the target hardware address (E34573242ACA) to the entry. Now it accesses queue 18 and sends all the packets in this queue, one by one, to the data link layer.

Example 5: Twenty-five seconds later, the cache-control module updates every entry.

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	**23**	**1**		**116.1.7.22**	
R	9		60	19.1.7.82	4573E3242ACA
R	18		900	188.11.8.71	E34573242ACA

Example: 25 secs later, the CC module updates every entry.

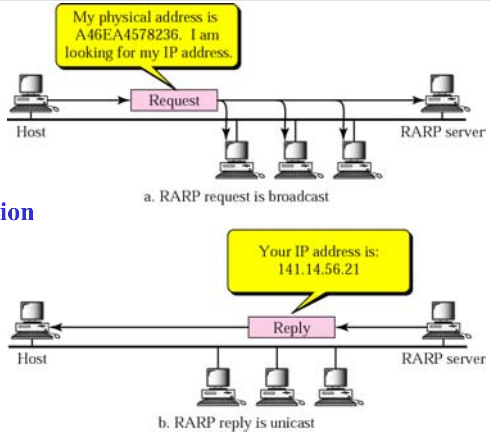
State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		840	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
F					
R	8		390	114.5.7.89	457342ACAE32
P	12	2		220.55.5.7	
P	23	2		116.1.7.22	
F					
R	18		875	188.11.8.71	E34573242ACA

The time-out values for the first three resolved entries are decremented by 60. The time-out value for the last resolved entry is decremented by 25. The state of the next-to-the last entry is changed to FREE because the time-out is zero. For each of the three entries, the value of the attempts field is incremented by one. After incrementing, the attempts value for one entry (the one with IP protocol address 201.11.56.7) is more than the maximum; the state is changed to FREE, the queue is deleted.

Final Cache Table

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		840	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
F					
R	8		390	114.5.7.89	457342ACAE32
P	12	2		220.55.5.7	
P	**23**	**2**		**116.1.7.22**	
F					
R	18		875	188.11.8.71	E34573242ACA

RARP Operation


My physical address is A46EA4578236. I am looking for my IP address.
Request
Host
RARP server
a. RARP request is broadcast
Your IP address is: 141.14.56.21
Reply
Host
RARP server
b. RARP reply is unicast

The RARP request packets are ***broadcast;***
the RARP reply packets are ***unicast.***

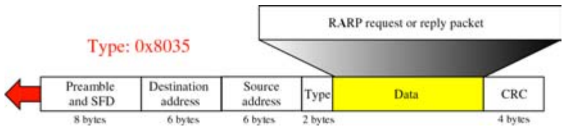
Same as ARP

RARP Packet Format

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

Exactly the same as ARP

Encapsulation of RARP packet



Alternative Solutions to RARP

- When a diskless computer is booted, it needs more information in addition to its IP address.
- Subnet mask, the IP address of a router, and the IP address of a name server are also needed.
- RARP cannot provide this extra information.
- New protocols have been developed to provide this information,e.g.,BOOTP and DHCP.

Example ARP Vulnerabilities

- Network administrators must be prepared to defend against misuse of ARP components
- Here we address one type of ARP vulnerability

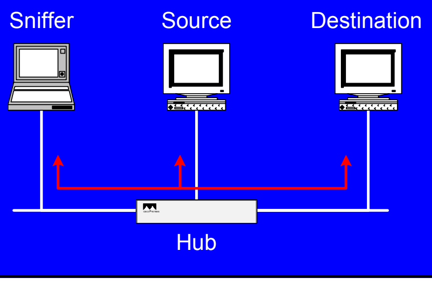
Unsolicited ARP Reply

- Any system can “spoof” (impersonate) an ARP reply to an ARP request
- Receiving system will cache the reply
  - Overwrite existing entry
  - Adds entry if one does not exist
- Usually called ARP “poisoning”
- Network administrators should monitor IP and MAC address mappings to check for anomalies

Some Types of Attacks to Defend Against

- Sniffing Attacks
- Session Hijacking
- Denial of Service

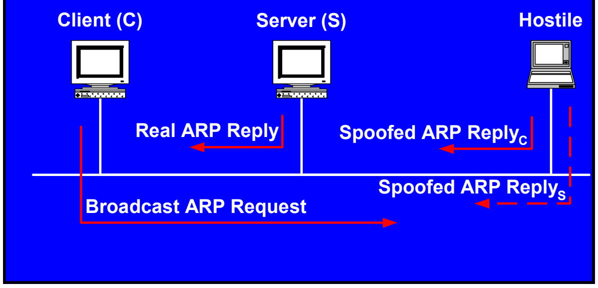
Sniffing on a Hub



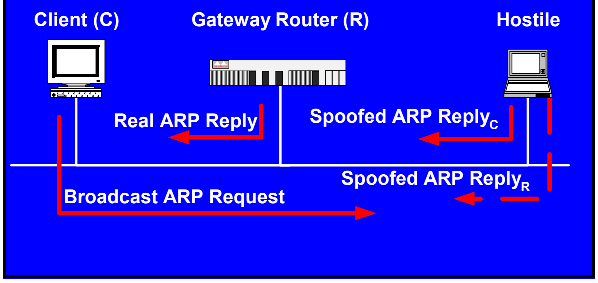
Switch Sniffing

- Normal switched networks
  - Switches relay traffic between two stations based on MAC addresses
  - Stations only see broadcast or multicast traffic
- Compromised switched networks
  - Attacker spoofs destination and source addresses
  - Forces all traffic between two stations through its system

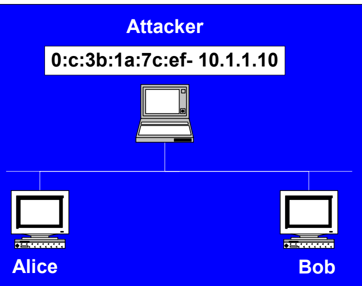
Host to Host Exploit



Host to Router Exploit



Relay Configuration

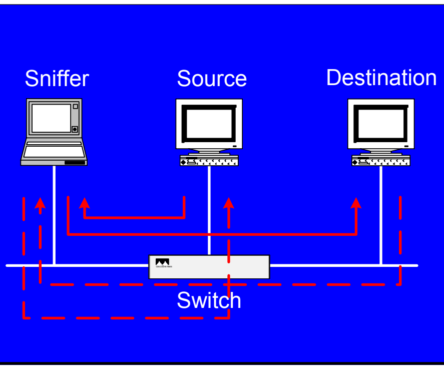


Attacker: 0:c:3b:1a:7c:ef- 10.1.1.10

Alice:
0:c:3b:1c:2f:1b- 10.1.1.2
0:c:3b:1a:7c:ef- 10.1.1.7

Bob:
0:c:3b:9:4d:8- 10.1.1.7
0:c:3b:1a:7c:ef- 10.1.1.2

Relay Configuration (cont.)


Sniffer
Source
Destination
Switch