

ARP

Address Resolution Protocol

SECURITY

Networking Basics

Communication in packet networks rely on several layers, with different identifiers:

- Applications use TCP/UDP ports
- Hosts use IP addresses
- Interface Cards use MAC addresses

Communication is typically made between applications using tuples <IP_Address:Port> and a protocol (e.g. TCP or UDP).

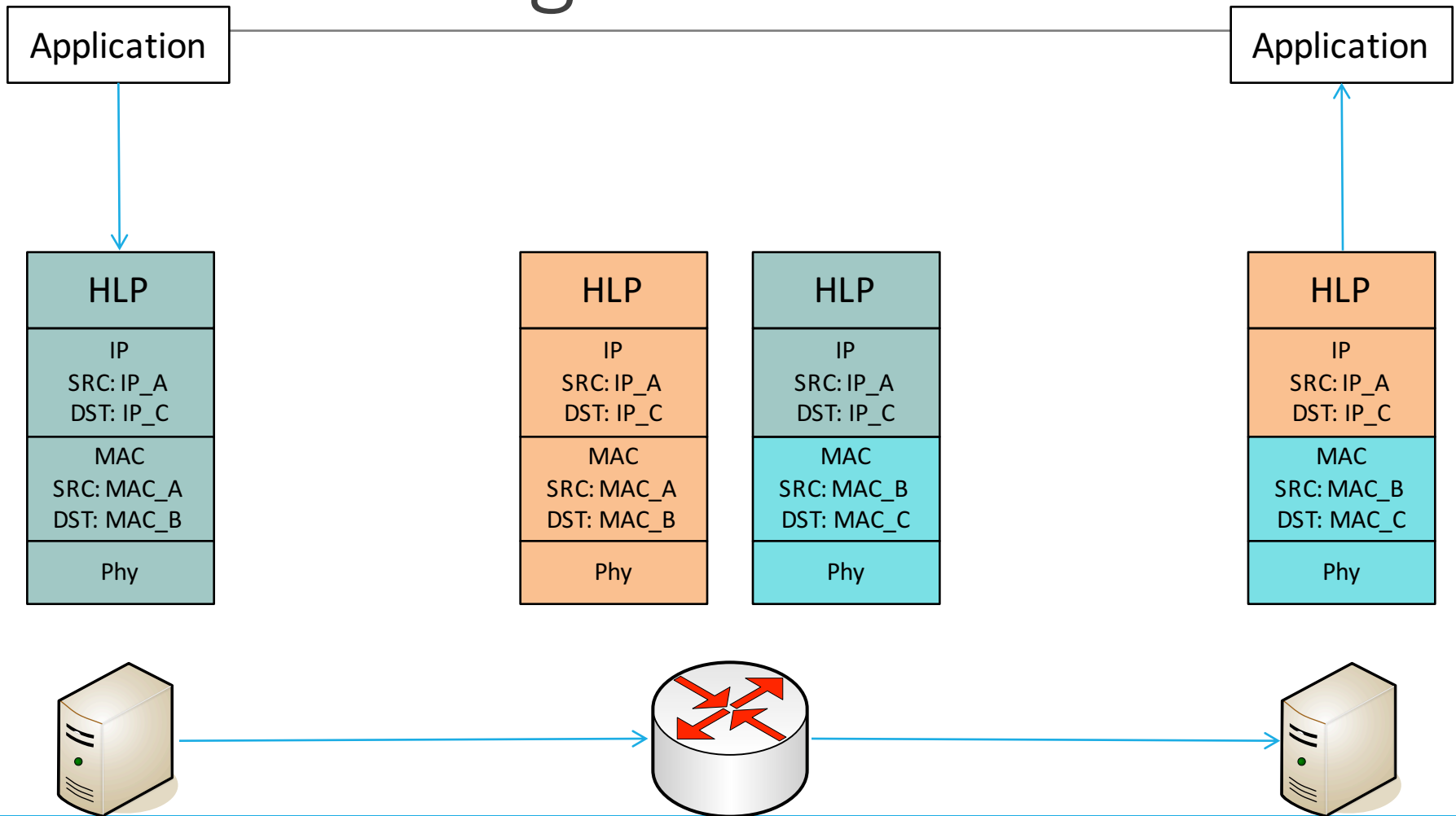
Networking Basics

When a packet is to be routed, two situations may occur:

- The packet is sent to the destination host, which is in the same IP network.
- The packet must be sent to a next hop (gateway), until it reaches the destination IP network.

In both cases, packet is transmitted between physical interfaces

Networking Basics



Networking Basics

IP addresses do not change between source and destination

MAC addresses are valid for a single network segment

- When packet is routed, MAC address of next hop must be found

IP to MAC mapping

Static configuration

- MAC entries of all hosts configured statically
 - All hosts “know” the MAC address of all interfaces of all other hosts
- Does not scale!
 - Changing a single interface requires updating all other hosts

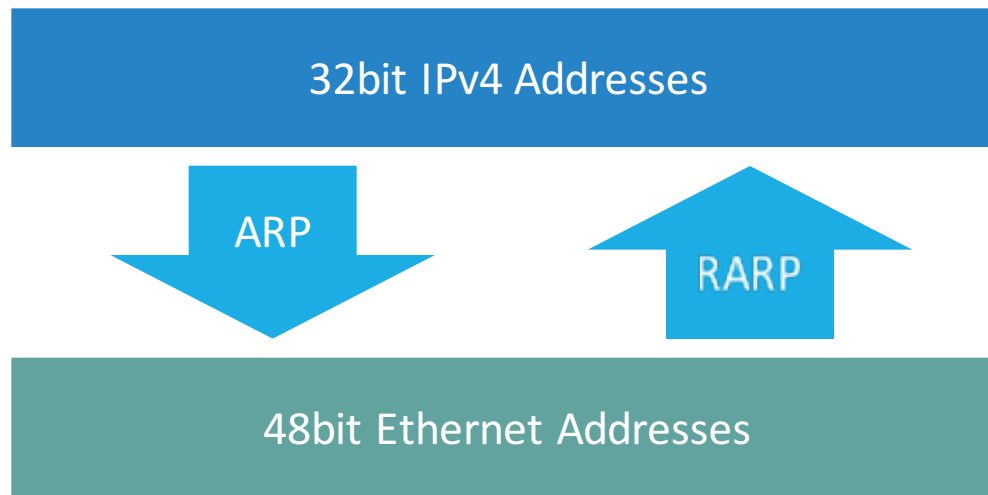
Dynamic configuration: ARP

Address Resolution Protocol

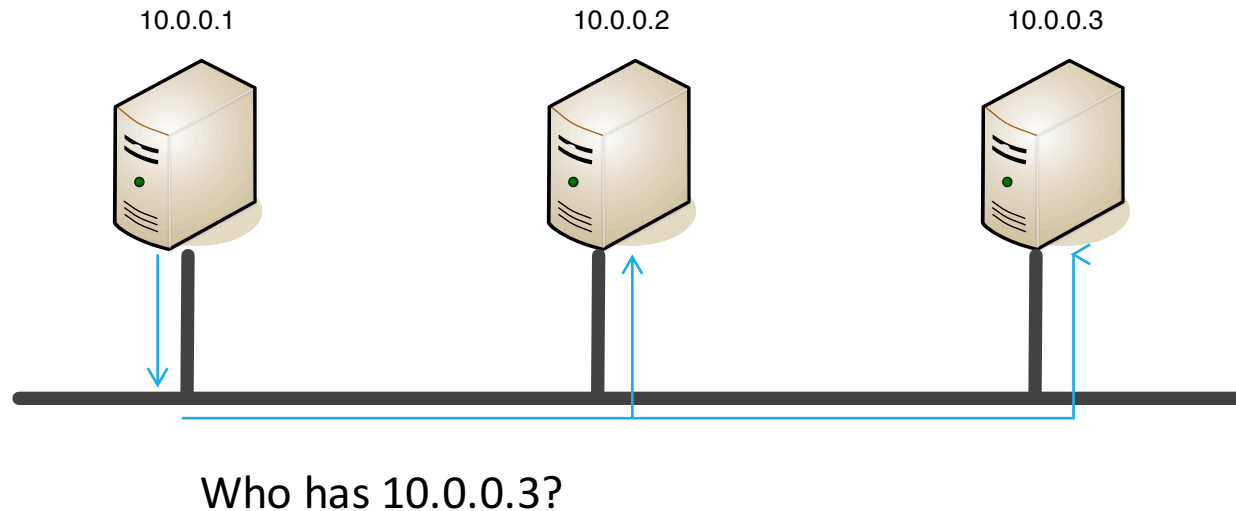
RFC 826

ARP: find MAC address of an Interface which is in a host with IP address

RARP: find IP address of host having an interface with the given MAC

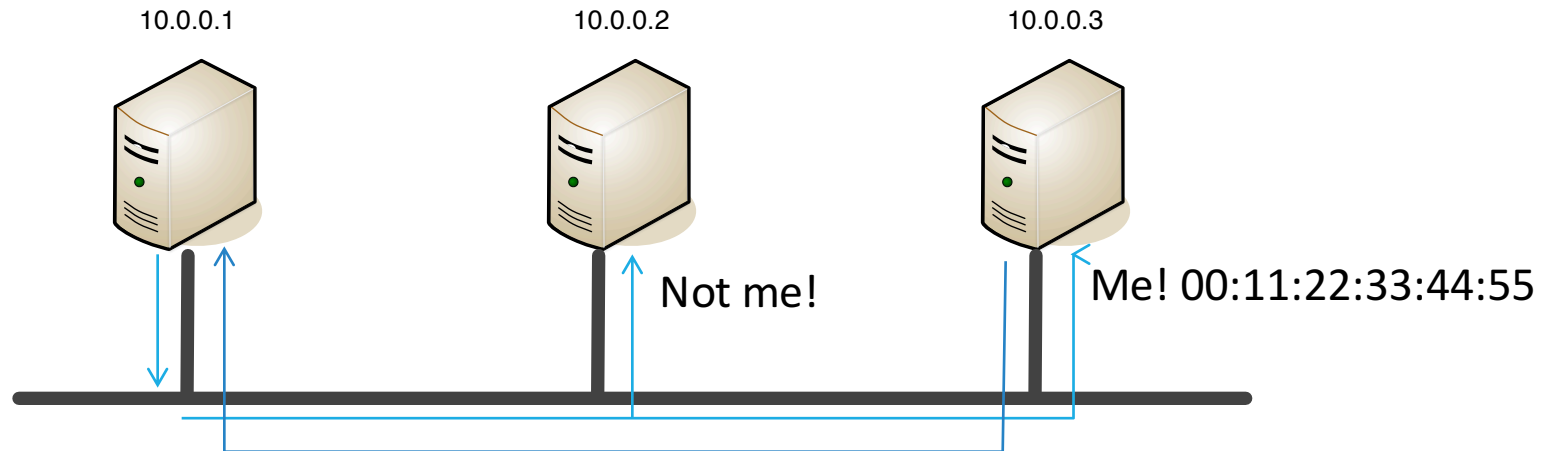


Address Resolution Protocol



Send ARP Request using broadcast.

Address Resolution Protocol



Reply using ARP Response using unicast

Address Resolution Protocol

Every packet sent requires two MAC address

- Source Address is known
- Destination Address must be determined

ARP Cache increases performance

- Caches both known and unknown entries
- Avoid repeating the discovery process per packet
- Entries have a large lifetime (2 minutes)

ARP Cache

```
security@security:~$ arp -a
```

```
fog.av.it.pt      (193.136.92.154) at 00:1e:8c:3e:6a:a6 [ether] on eth0
atnog.av.it.pt    (193.136.92.123) at 00:15:17:e6:6f:67 [ether] on eth0
guarani.av.it.pt  (193.136.92.134) at 00:0c:6e:da:19:87 [ether] on eth0
aeolus.av.it.pt   (193.136.92.136) at bc:ae:c5:1d:c6:53 [ether] on eth0
```

ARP Spoofing

MAC addresses can be modified

```
ifconfig eth0 hw ether 00:11:22:33:44:55
```

Using a colliding MAC address will allow reception of network traffic for other hosts

- Some switches limit MAC addresses to single ports

Sending ARP packets with spoofed addresses may poison the cache of other stations

- ARP Poisoning

ARP Poisoning

Hosts cache information directly from ARP packets

- No other verification is done

New information will replace existing entries

- Great for allowing network dynamism
- Very bad for security

Possible to send specially crafted packets to create specific entries in remote hosts

ARP Poisoning

When receiving an ARP Request:

```
▶ Frame 10: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▶ Ethernet II, Src: Apple_1b:1f:42 (e0:f8:47:1b:1f:42), 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: Apple_1b:1f:42 (e0:f8:47:1b:1f:42)
  Sender IP address: 10.0.0.3 (10.0.0.3)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 10.0.0.2 (10.0.0.2)
```

10.0.0.2 will send an ARP Reply

But... 10.0.0.2 will also “learn” that 10.0.0.3 is at e0:f8:47:1b:1f:42

ARP Poisoning

When receiving an ARP Reply

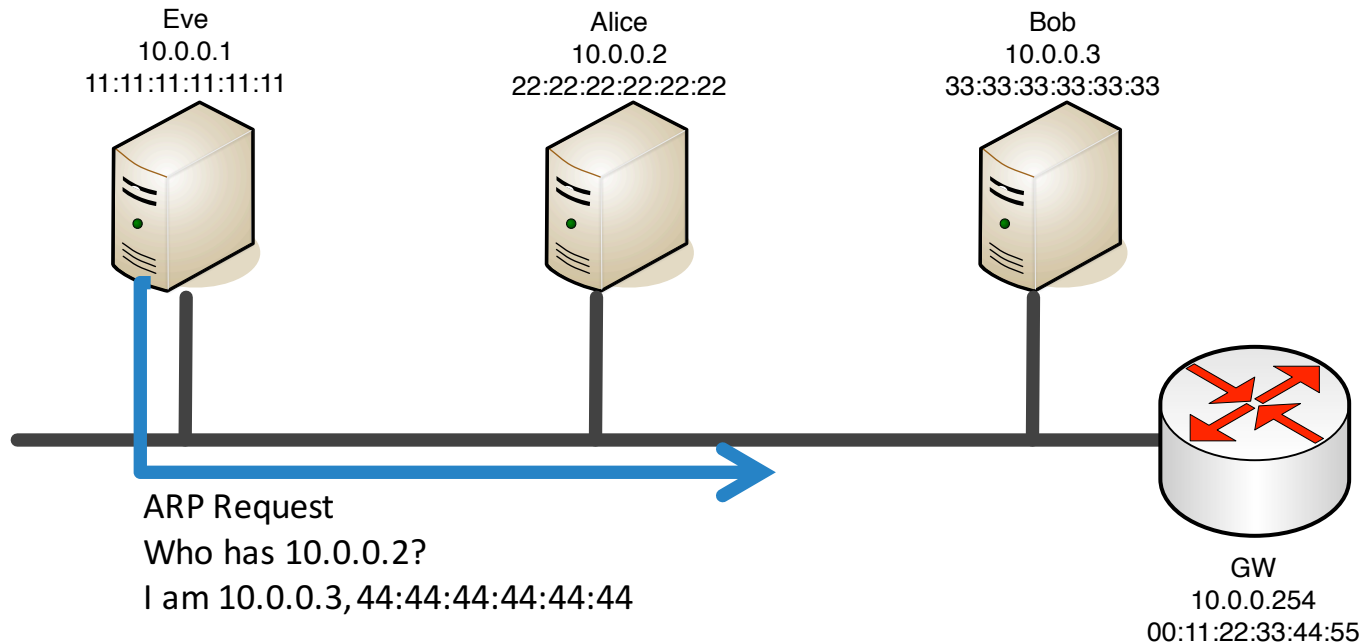
```
▶ Frame 123: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▶ Ethernet II, Src: Tp-LinkT_f2:77:62 (90:f6:52:f2:77:62), Dst: Apple_1b:1f:42 (e0:f8:47:1b:1f:42)
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Tp-LinkT_f2:77:62 (90:f6:52:f2:77:62)
  Sender IP address: 10.0.0.246 (10.0.0.246)
  Target MAC address: Apple_1b:1f:42 (e0:f8:47:1b:1f:42)
  Target IP address: 10.0.0.3 (10.0.0.3)
```

10.0.0.3 will learn that 10.0.0.246 is at 90:f6:52:f2:77:62
.... even if no matching request as made...

ARP Poisoning: Consequences

Hosts can be isolated from the network

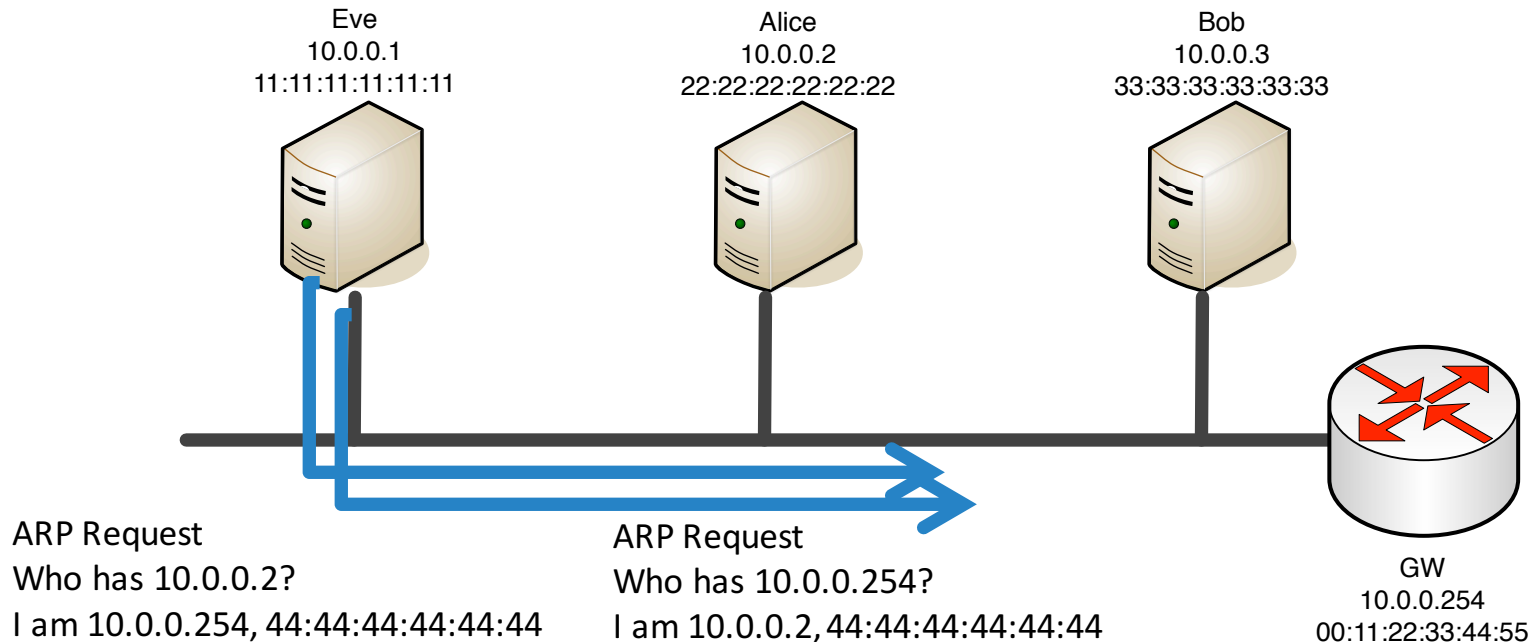
- Create fake entries for all other hosts



Alice will use 44:44:44:44:44:44 when talking to Bob

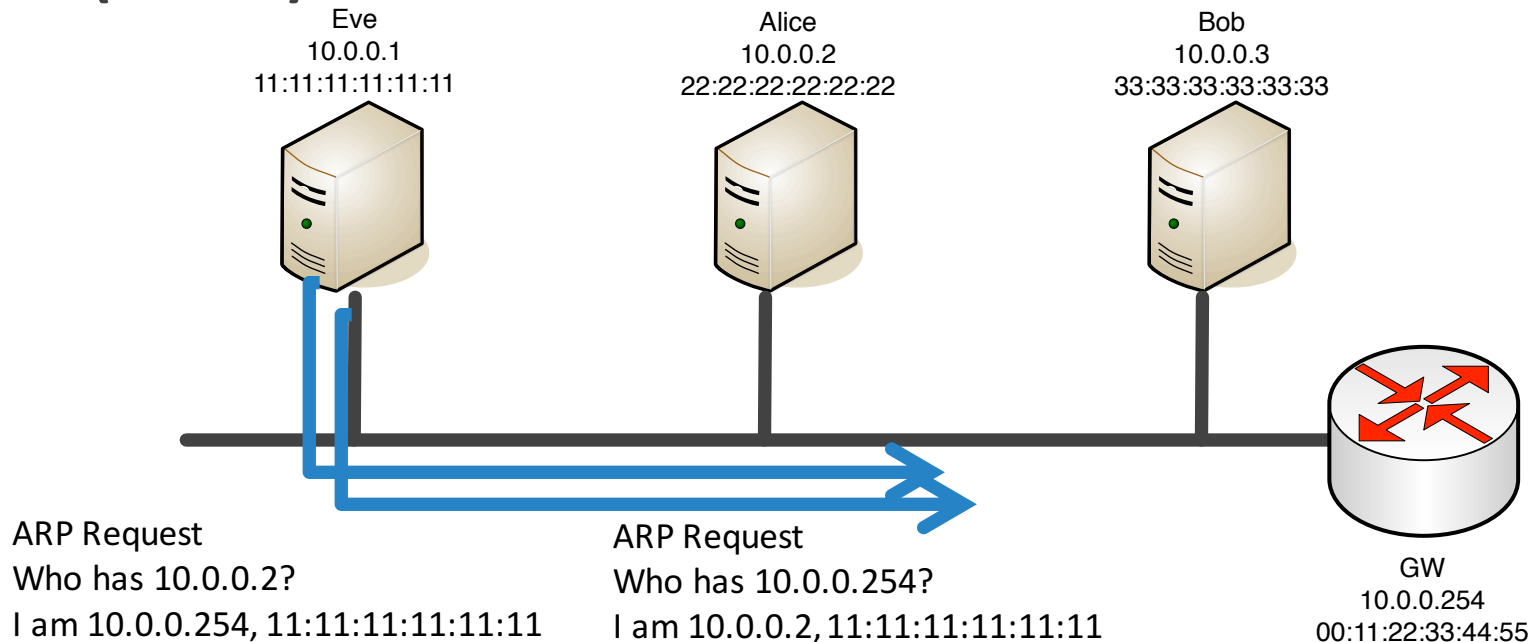
ARP Poisoning: Consequences

Hosts can be denied communication with the outside world



ARP Poisoning: Consequences

Traffic between two hosts can be intercepted (MitM)



- Then Eve will forward traffic

ARP Poisoning: Avoidance

Use static entries

- No resolution process is triggered
- Colliding Information from ARP packets is discarded

Behavior detection

- Detect ARP Replies without Request
- Detect repeated Requests from same host.

ARP Poisoning: Avoidance

Use monitoring software

- Software watches for MAC changes
 - Network administrator is notified
- ARP Poison is not actually avoided!

Port based packet filtering at switch ingress

- Spoofed ARP packets are dropped
- Only possible in static scenarios