Address Resolution Protocol (ARP)

Recap

Past Topics

- Overview of networking and layered architecture
- Wireshark packet sniffer
- Ethernet and WiFi
- IPv4
- Scapy

Today's Topics

- Address Resolution Protocol (ARP)

Address Resolution Protocol

- Find link layer address given a network layer address
  - What is the **Ethernet address** for a given **IP address**?
- Every IP node (hosts and routers) has an ARP table
  - Mapping from IP to Ethernet addresses on their LAN
  - May be incomplete
  - Can include both static and dynamic entries

Dynamic ARP Entries

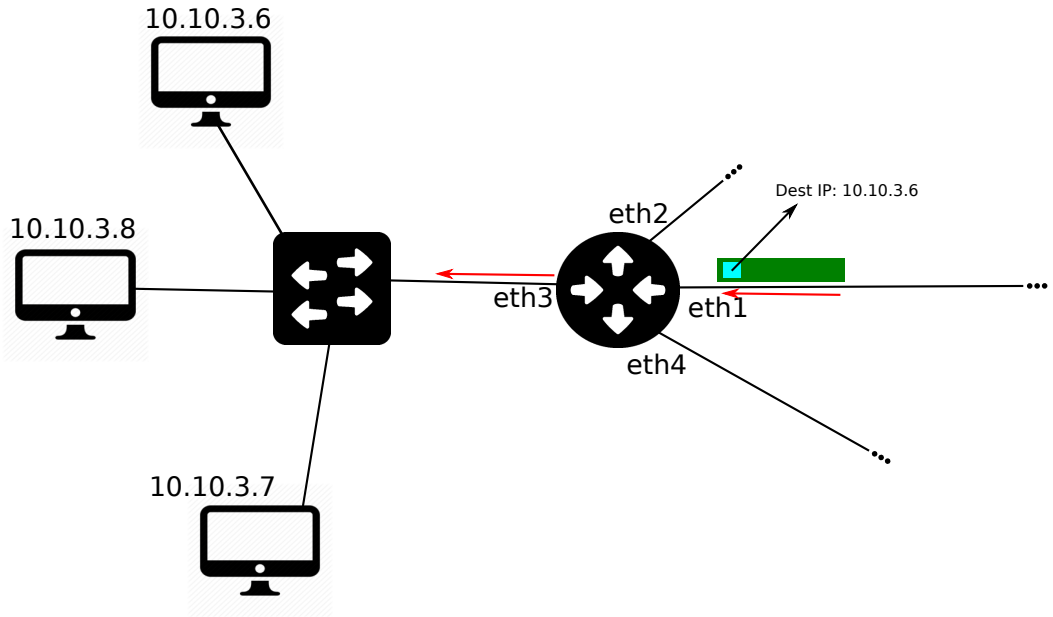
- Systems "discover" IP → Ethernet address mappings, as needed
- Each entry has an IP address, an Ethernet address, and a timeout (typically around 1 minute)
- ARP messages are **broadcast** on the LAN to discover mappings
  - All computers on the network receive the ARP requests

Learning MAC addresses

- Hosts learn IP → Ethernet address mappings
  - ARP responses are stored in ARP tables
  - ARP requests are stored in ARP tables (whether the host is the target or not!)
- ARP entries time out
  - Allow machines to change IP and/or MAC addresses transparently
  - Eliminate stale entries (machines turn off, move, crash, etc.)

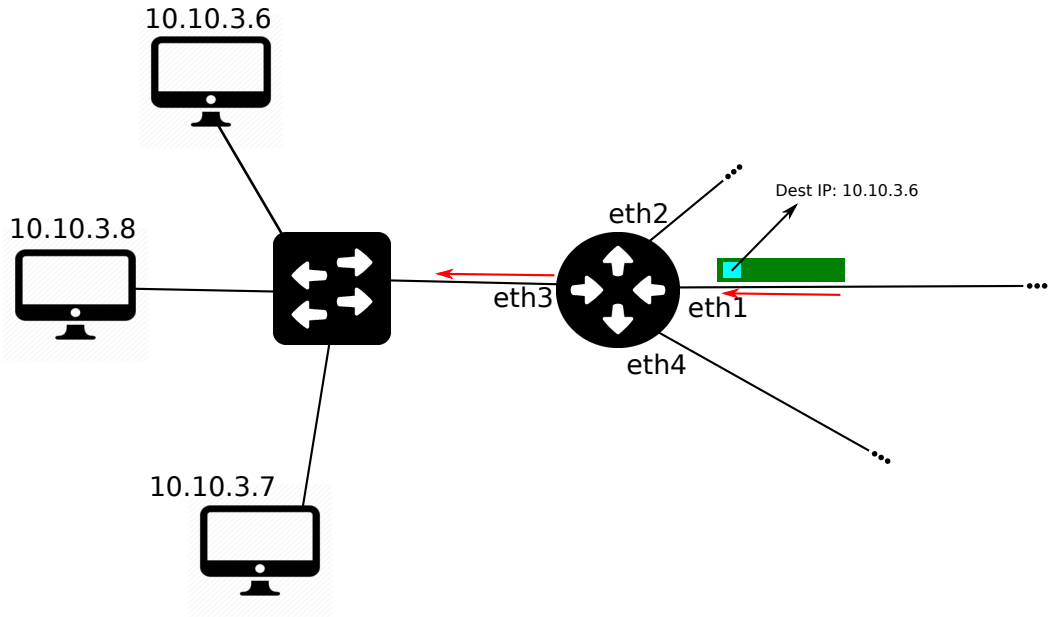
ARP Scenario

- Router receives an IP packet
  - Reads destination IP address
  - Uses *longest prefix match* to determine next hop and egress interface



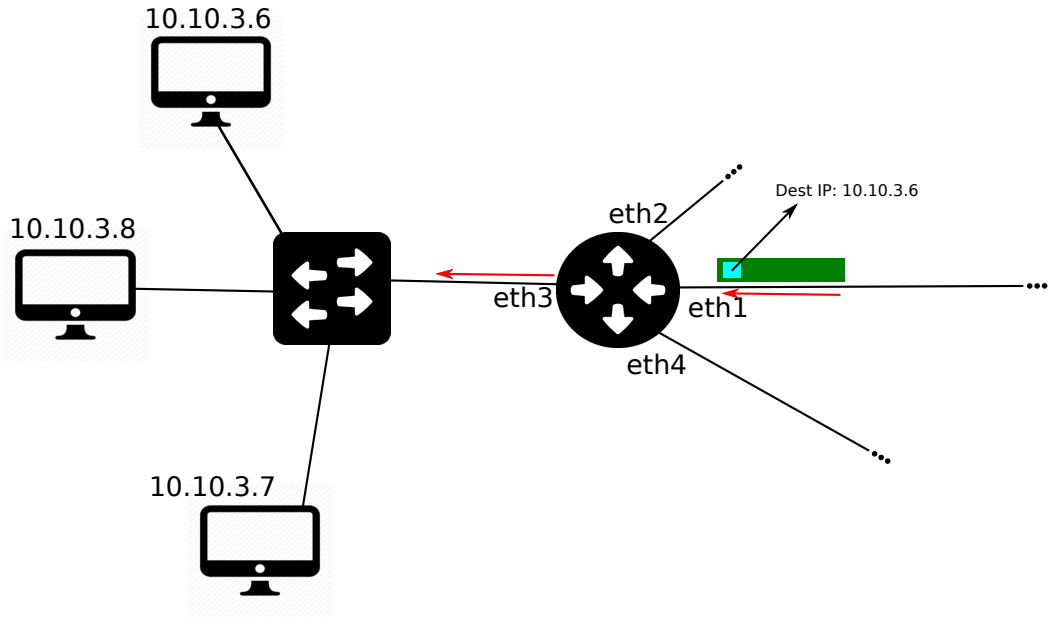
ARP Scenario

- Before forwarding IP packet, router needs to construct a new link layer frame
  - Source MAC = Address of egress interface (local to router)
  - Destination MAC = ????



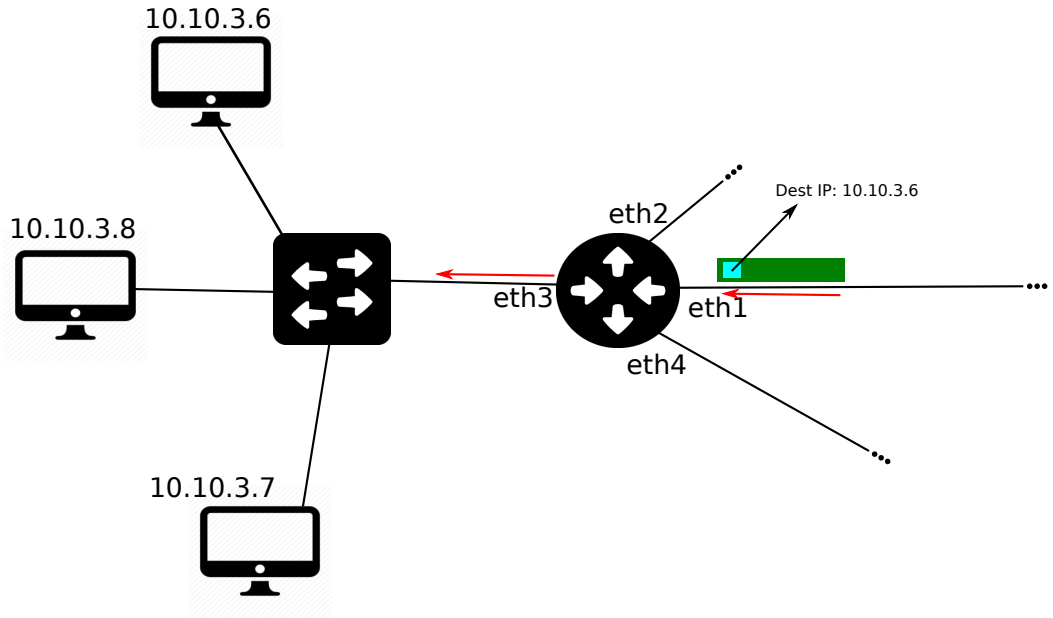
ARP Scenario

- Destination MAC address?
  - If *next hop* is another router, it's the MAC address of that router
    - Current router would have IP address of next hop router
  - If *next hop* is the destination, then it's the MAC address of the receiver
    - Current router has IP address of the destination



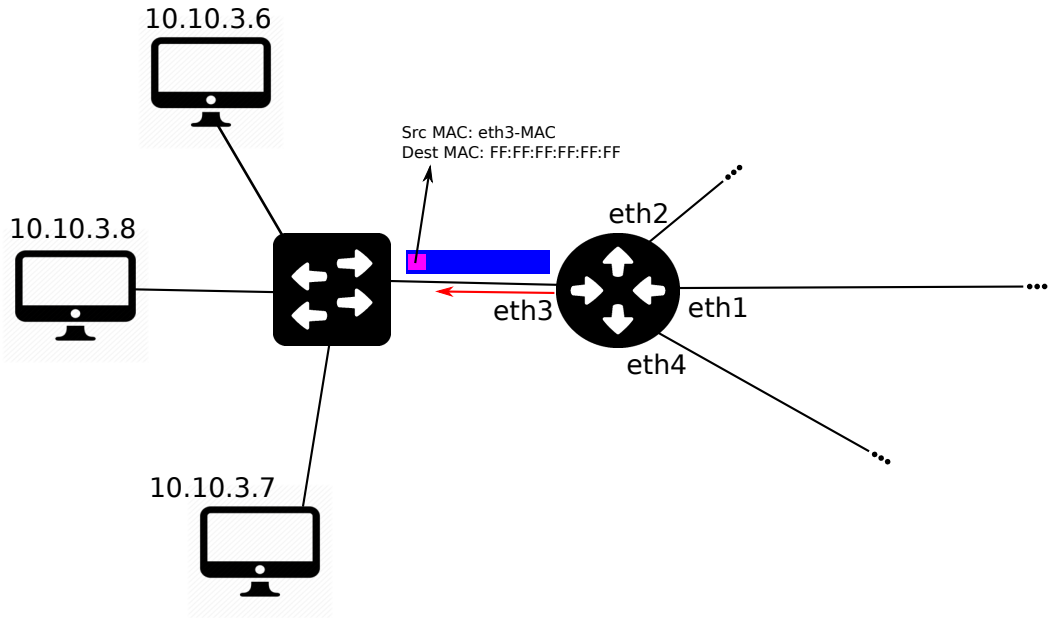
ARP Scenario

- Address Resolution Protocol (ARP) is used to translate between the known IP address and the unknown MAC address
  - Translate from network layer → link layer



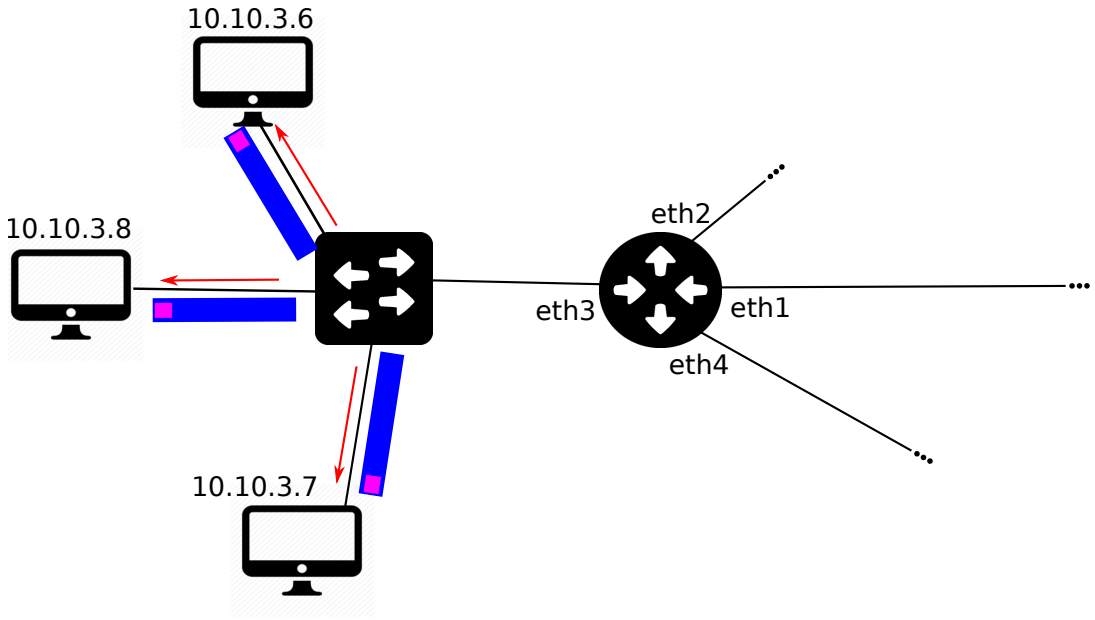
ARP Scenario

- Router constructs a *broadcast* frame containing ARP request for the destination IP address in question



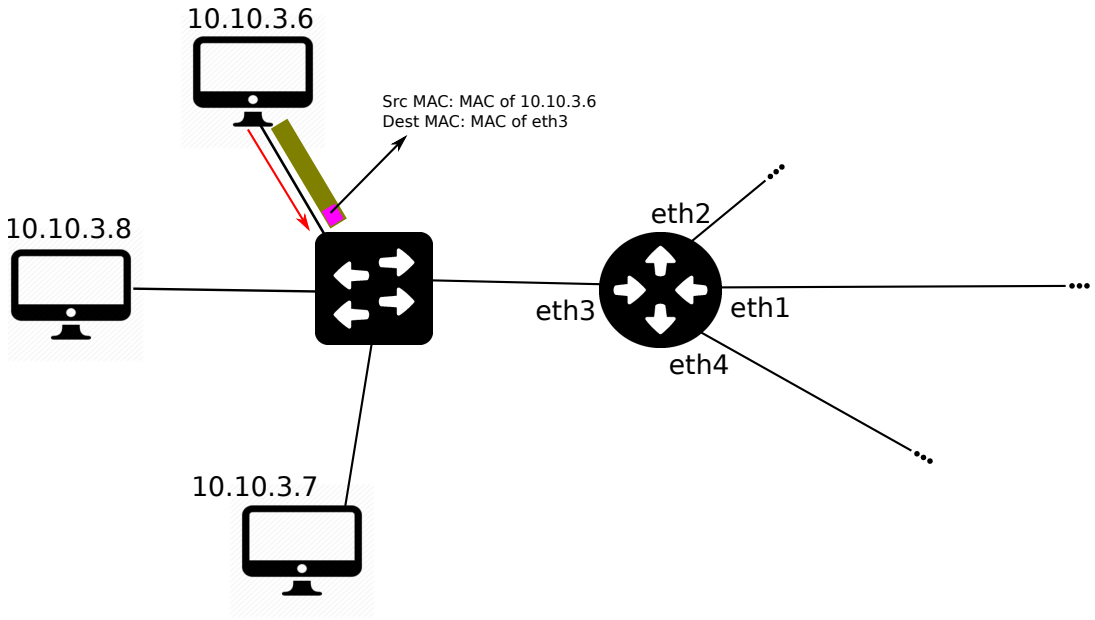
ARP Scenario

- Router constructs a *broadcast* frame containing ARP request for the destination IP address in question
  - Every node in LAN receives the ARP request



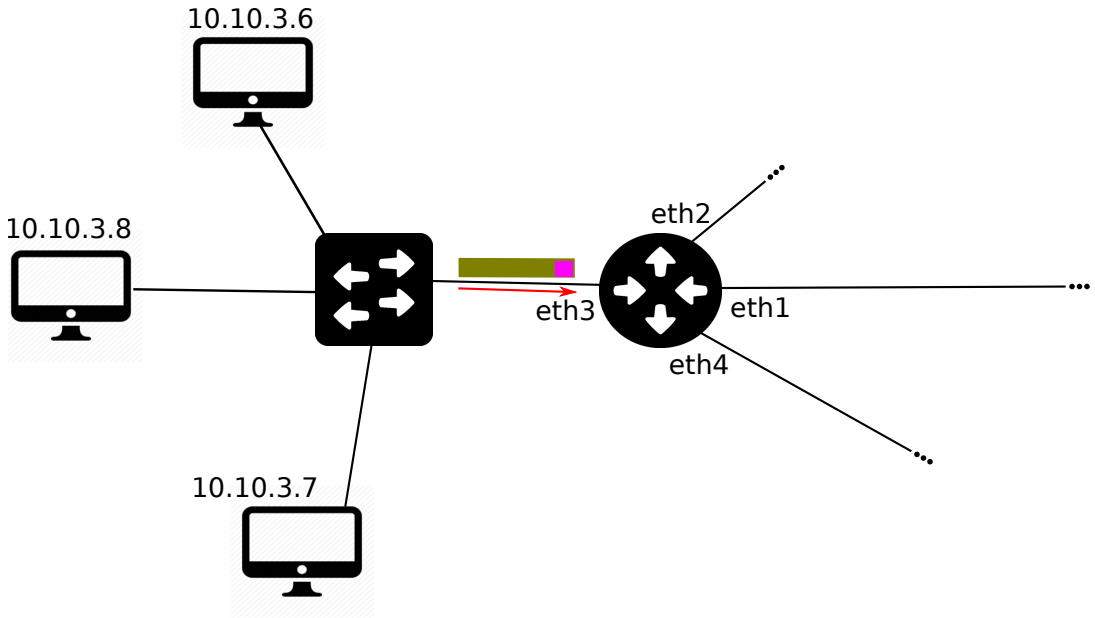
ARP Scenario

- The node with the matching IP address generates an ARP reply (with information on its matching MAC address)
  - Sent directly back to router (not broadcast)



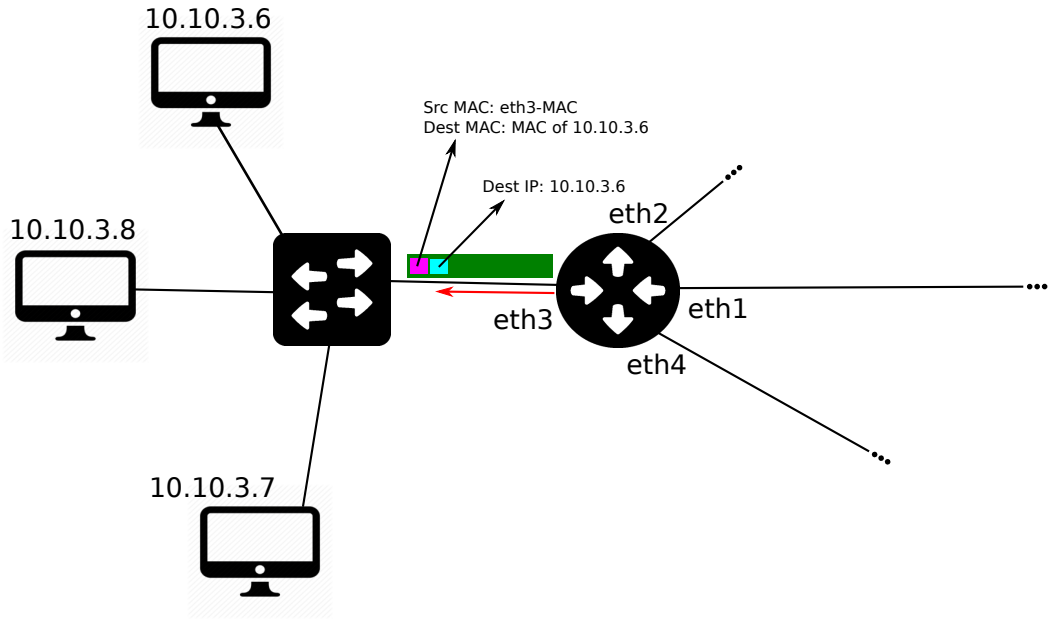
ARP Scenario

- The node with the matching IP address generates an ARP reply (with information on its matching MAC address)
  - Sent directly back to router (not broadcast)



ARP Scenario

- Now the router can finally complete the link layer frame for the packet waiting to be forwarded



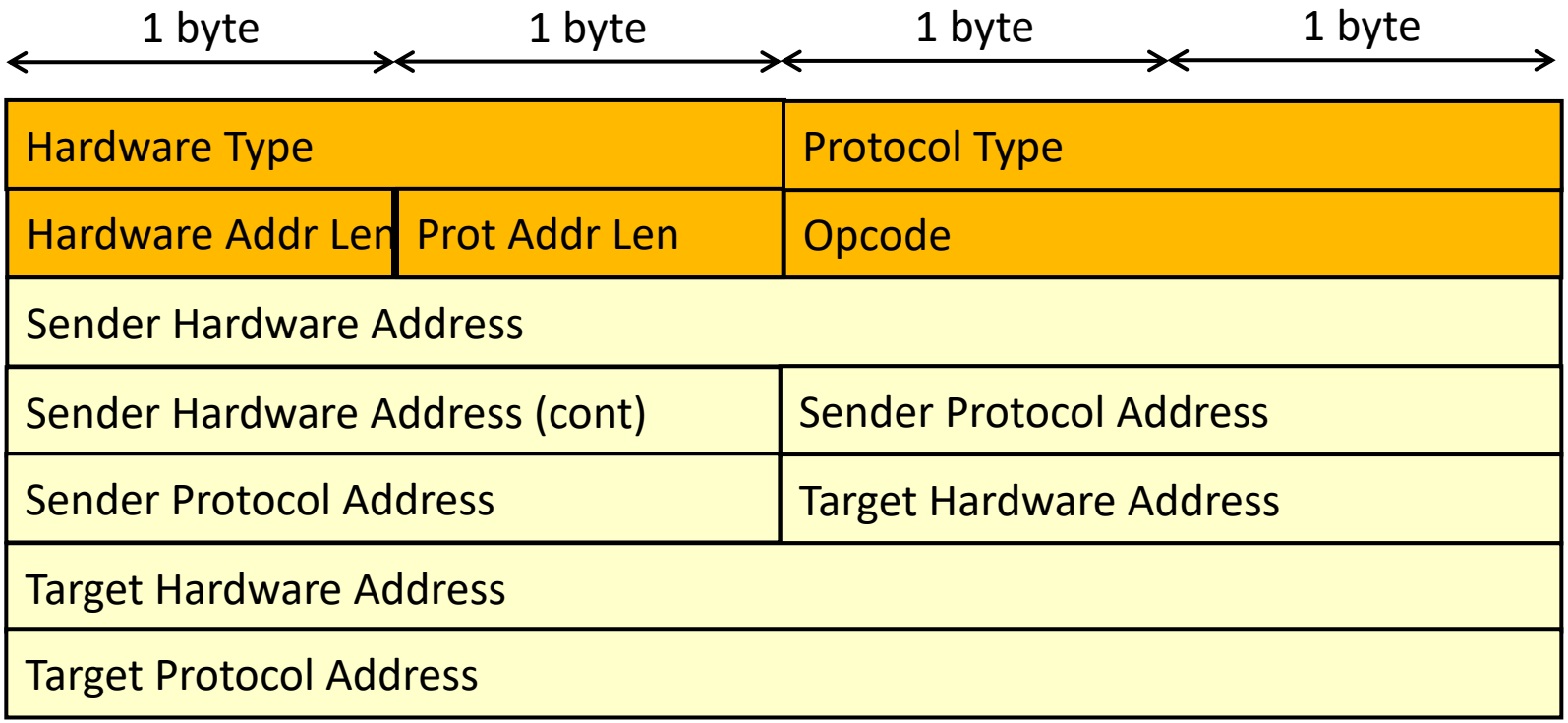
ARP Cache

- Each host maintains an ARP cache
  - An ARP cache is a table that maps IP addresses to MAC addresses
- ARP cache entries expire and need to be updated
  - Expiration time ranges from seconds to a few minutes
  - Upon receiving an ARP reply, the ARP cache is updated

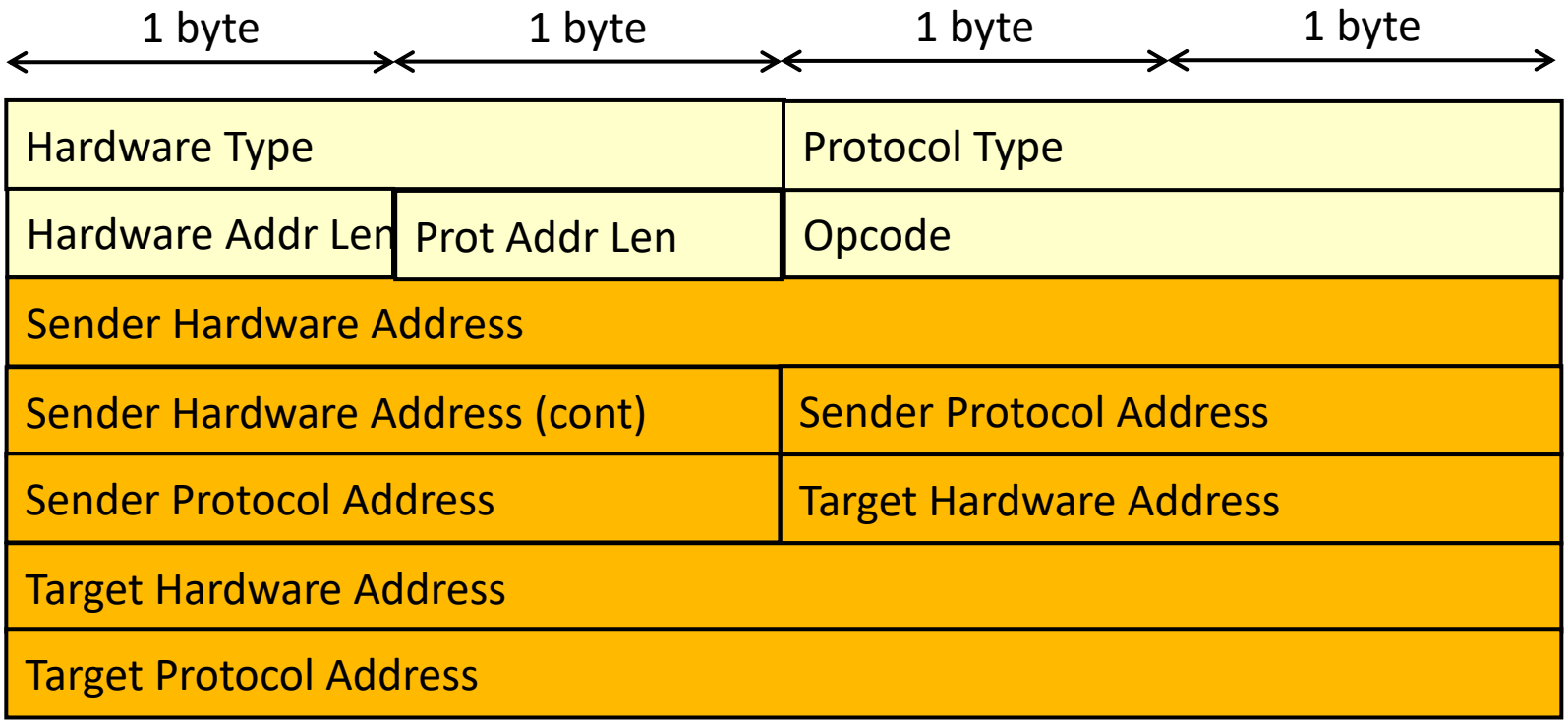
IP Address	MAC Address
10.0.1.1	5a:77:28:e3:ff:26
10.0.1.2	62:a0:4b:19:34:6d
10.0.1.3	96:03:37:be:73:cc
...	...

ARP Cache

- The *ARP Requests* provide a useful benefit to other hosts on the network (beyond the specific query)
- Every node within a LAN that receives the broadcast ARP request:
  - Checks its ARP cache to see if there is a match between the source of the ARP request and an entry in the cache
  - If so, updates the cache with the potentially new MAC address
  - Helps to avoid *stale cache entries*



- **Hardware Type** (2 bytes): Link layer used (e.g. Ethernet, `0x0001`)
- **Protocol Type** (2 bytes): Network layer used (e.g. IP, `0x0800`)
- **Hardware Addr Len** (1 byte): Size of link layer address in bytes
- **Protocol Addr Len** (1 byte): Size of network layer address in bytes
- **Opcode** (2 bytes): Type of ARP message (1=Request, 2=Reply)



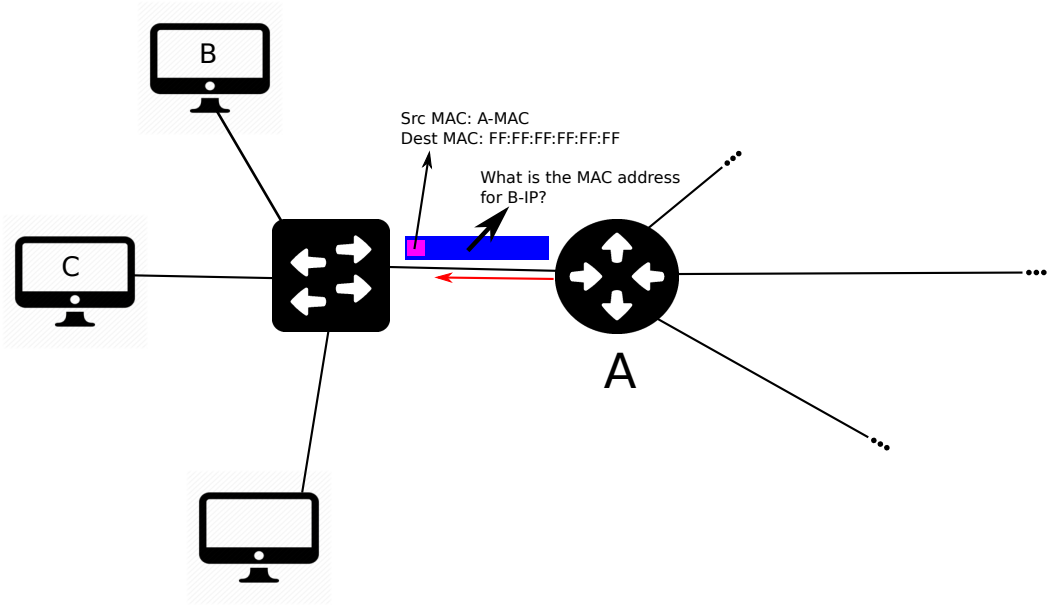
- **Sender Hardware Address**: Sender Link Layer address
- **Sender Protocol Address**: Sender Network Layer address
- **Target Hardware Address**: Target Link Layer address
- **Target Protocol Address**: Target Network Layer address

ARP "Security"

- ARP does not provide *authentication* for the two ends of communication
  - A malicious node can impersonate another one
- ARP does not provide *integrity* of data within the packet
  - A malicious entity can set the fields arbitrarily
- These deficiencies allow *ARP spoofing attacks*

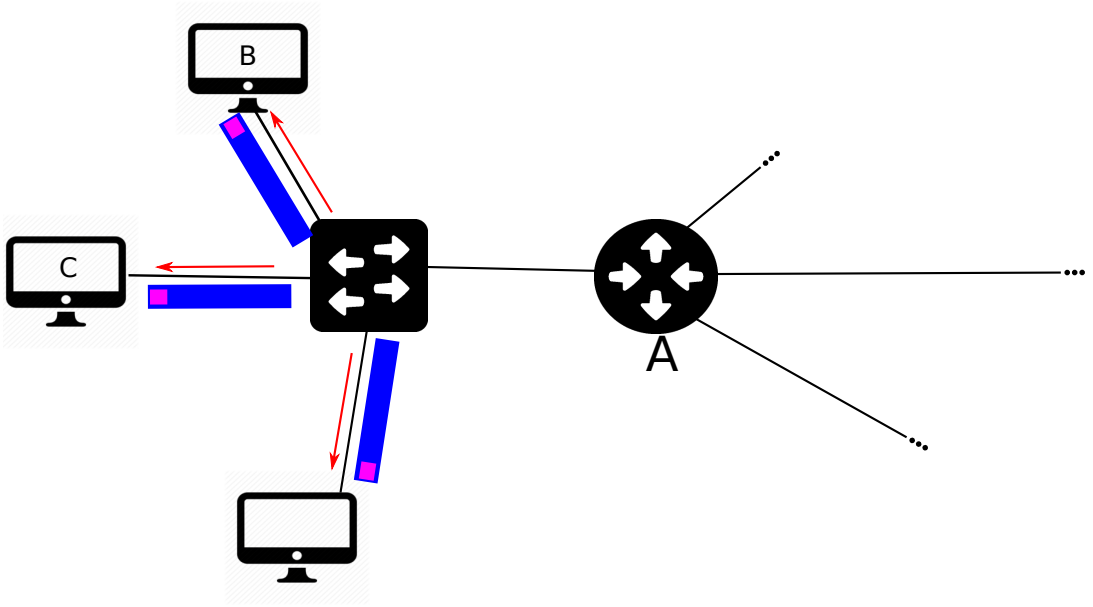
ARP Spoofing Example

- Assume nodes A, B, and C are within a LAN
- A wants to discover the MAC address of B in order to send an IP packet
  - A broadcasts an ARP request



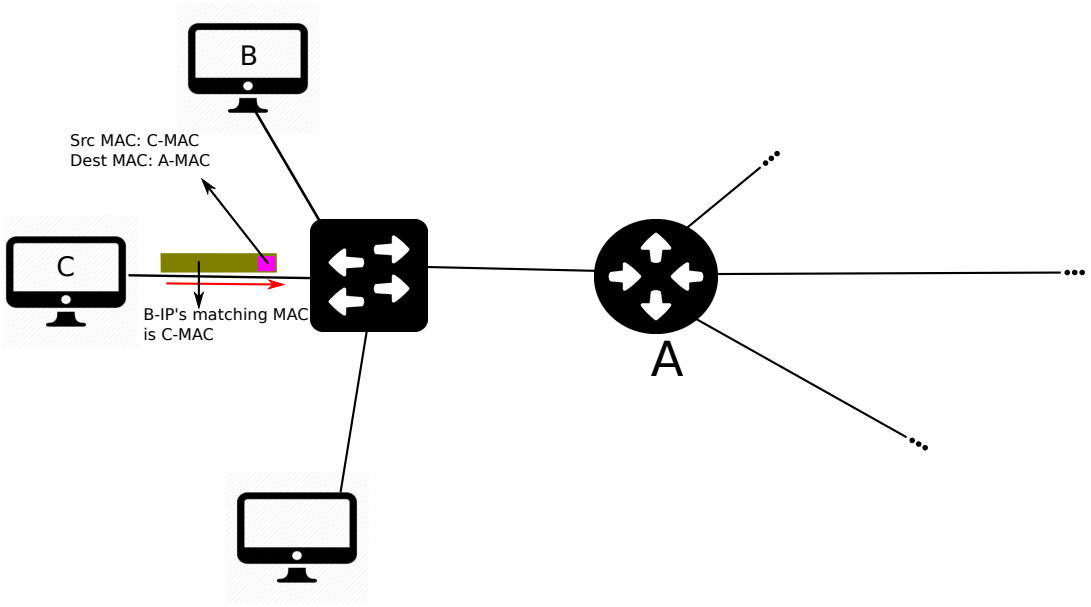
ARP Spoofing Example

- Assume nodes A, B, and C are within a LAN
- A wants to discover the MAC address of B in order to send an IP packet
  - A broadcasts an ARP request



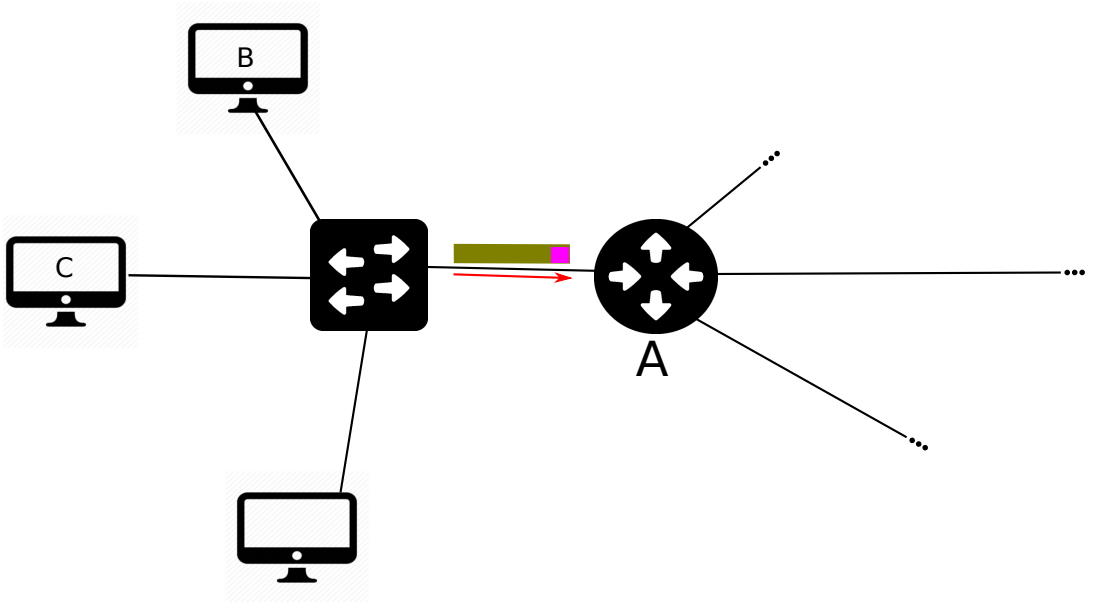
ARP Spoofing Example

- B seems to be offline for some reason (or doesn't respond quickly)
- C responds with unicast ARP reply asserting that B's IP address matches its own MAC address



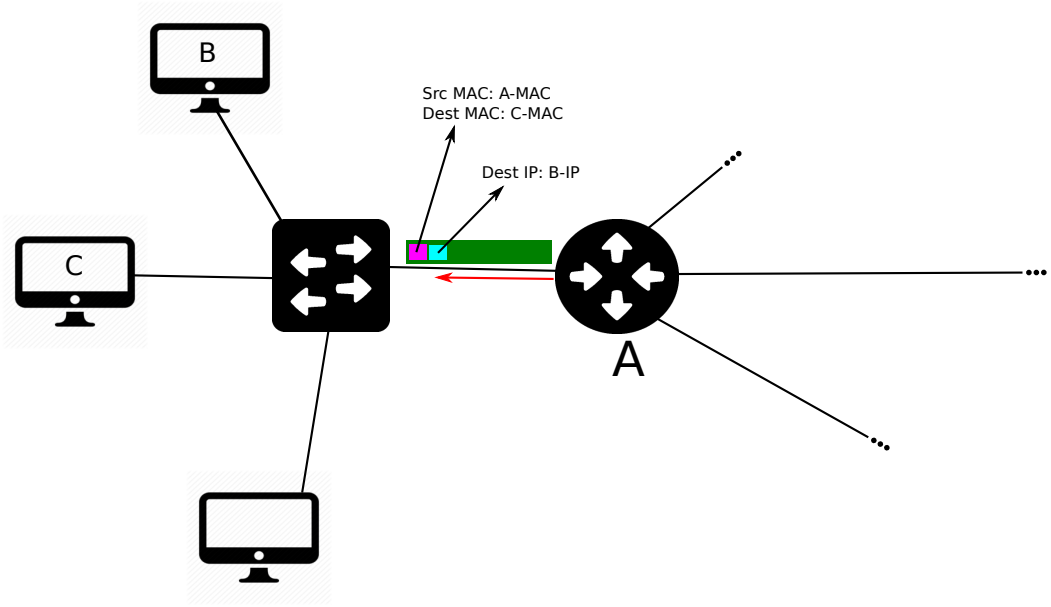
ARP Spoofing Example

- B seems to be offline for some reason (or doesn't respond quickly)
- C responds with unicast ARP reply asserting that B's IP address matches its own MAC address



ARP Spoofing Example

- There is no way that A can verify this assertion
  - A accepts the ARP reply and adds to the ARP cache a mapping from B's IP address to C's MAC address
  - Then, the IP packet is encapsulated in a link layer frame with the destination MAC address set to C's and sent
  - C receives the packet that was supposed to be received by B



Closing Thoughts

Recap

- Today we discussed ARP
  - How ARP works
  - ARP packet format
  - ARP spoofing

Next Class

- Virtual LANs (VLANs)

Class Activity

CA.7 – ARP & Wireshark

Due tonight at 11:59pm

Homework 2

Due Oct 2nd at 11:59pm

Project 1

Due Sept 30th at 11:59pm