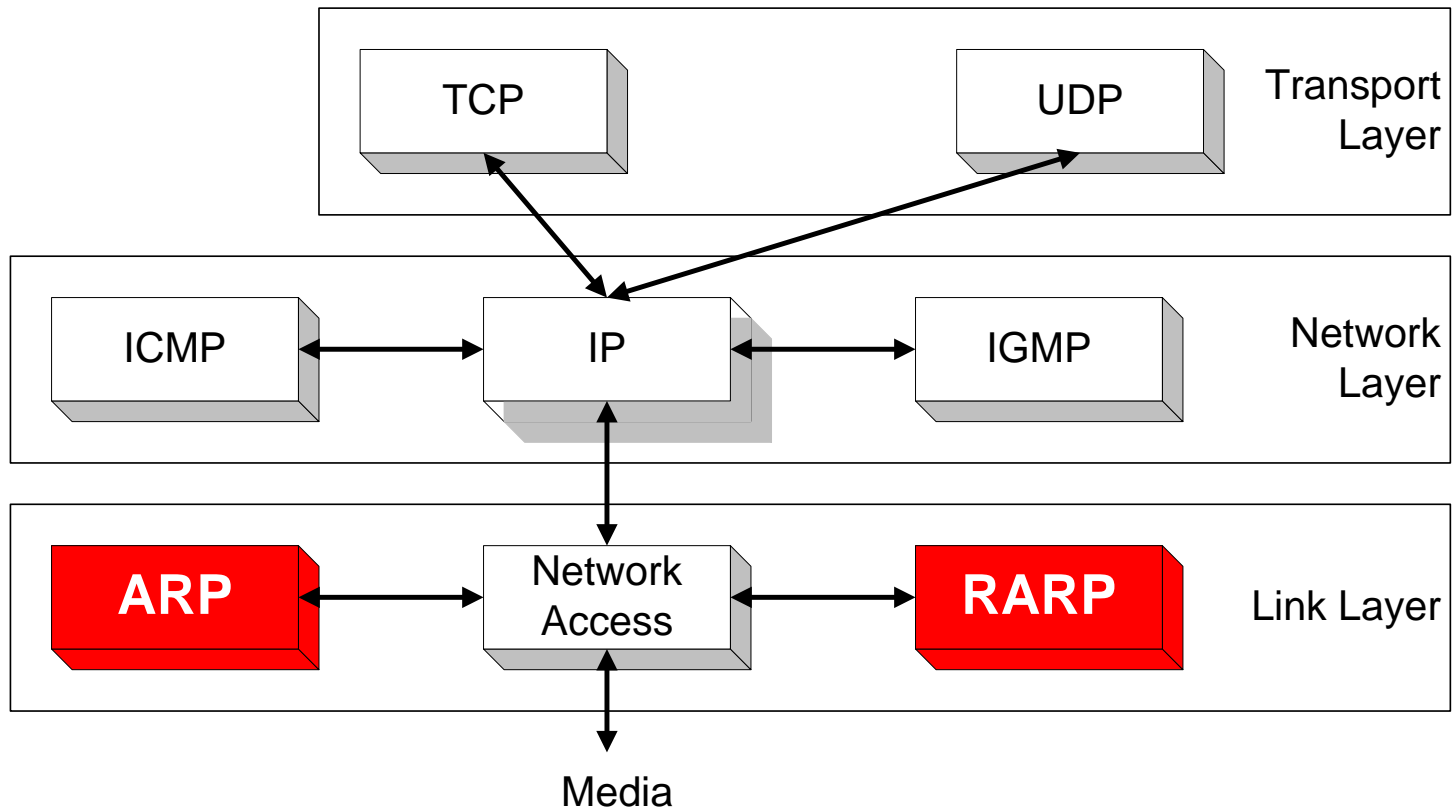


# **Address Resolution Protocol (ARP)**

---

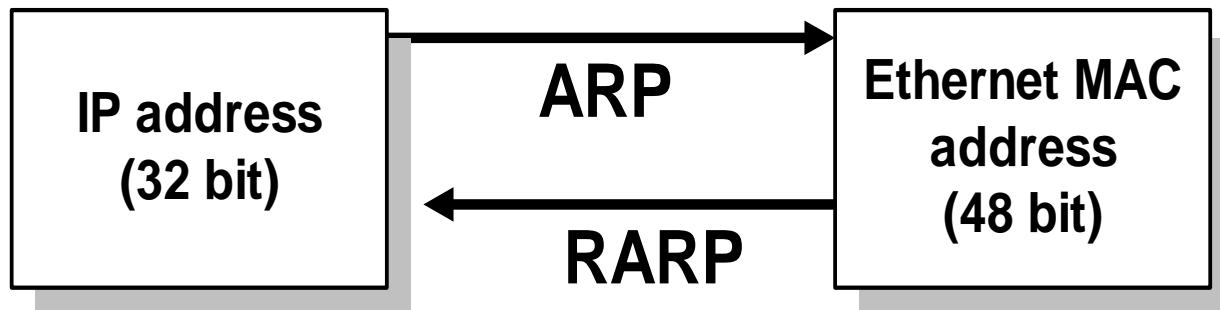
This module is about the address resolution protocol.

# Overview

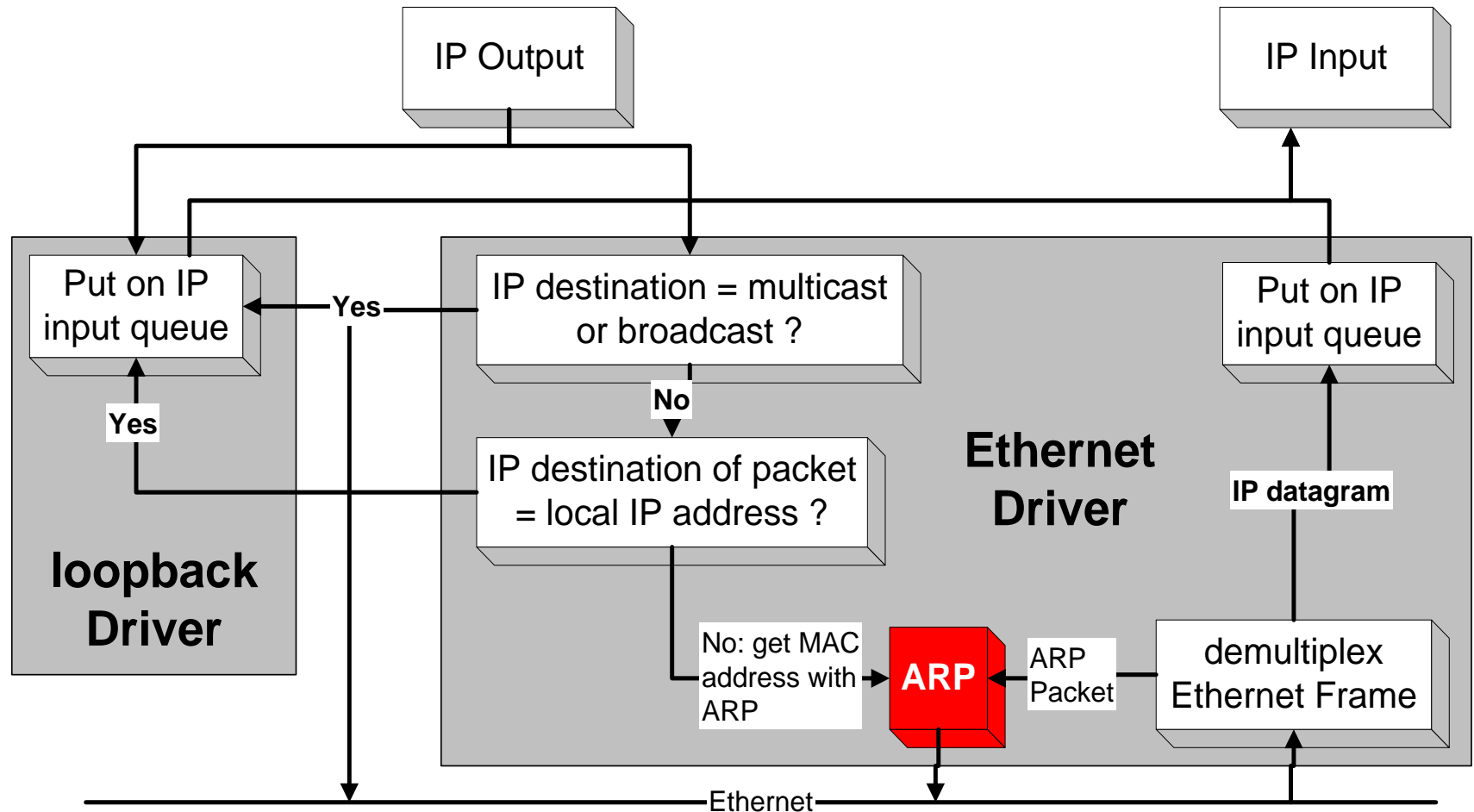


# ARP and RARP

- Note:
  - The Internet is based on IP addresses
  - Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the **translation between IP addresses and MAC layer addresses**
- We will discuss ARP for broadcast LANs, particularly Ethernet LANs



# Processing of IP packets by network device drivers



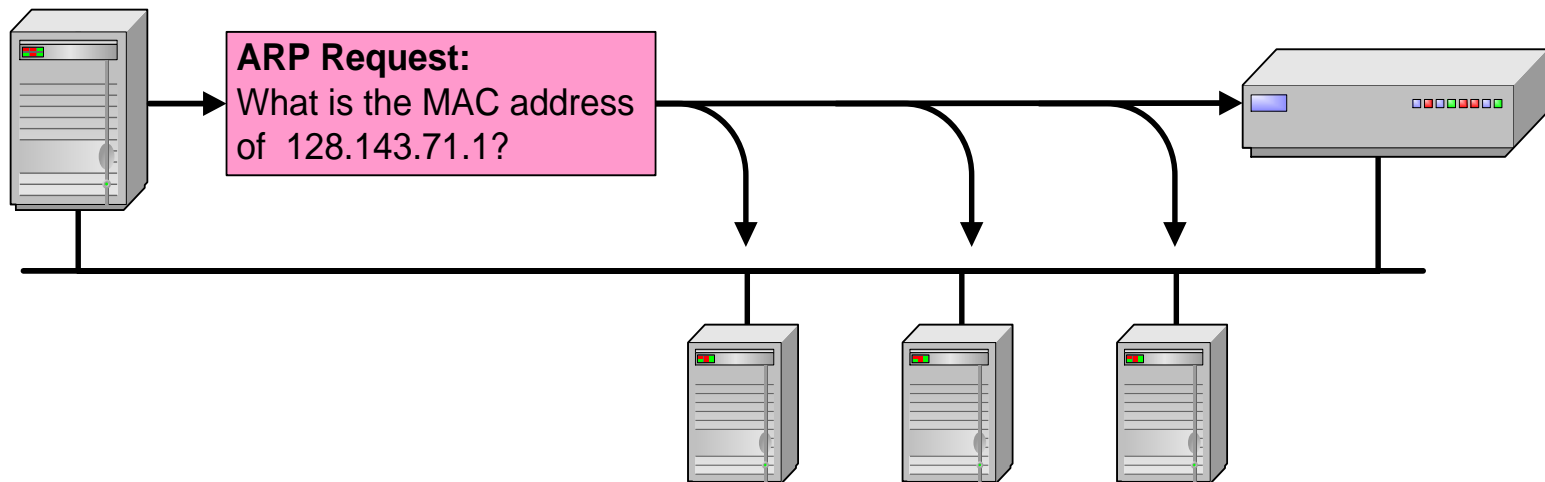
# Address Translation with ARP

## ARP Request:

Argon broadcasts an ARP request to all stations on the network: **“What is the hardware address of Router137?”**

Argon  
128.143.137.144  
00:a0:24:71:e4:44

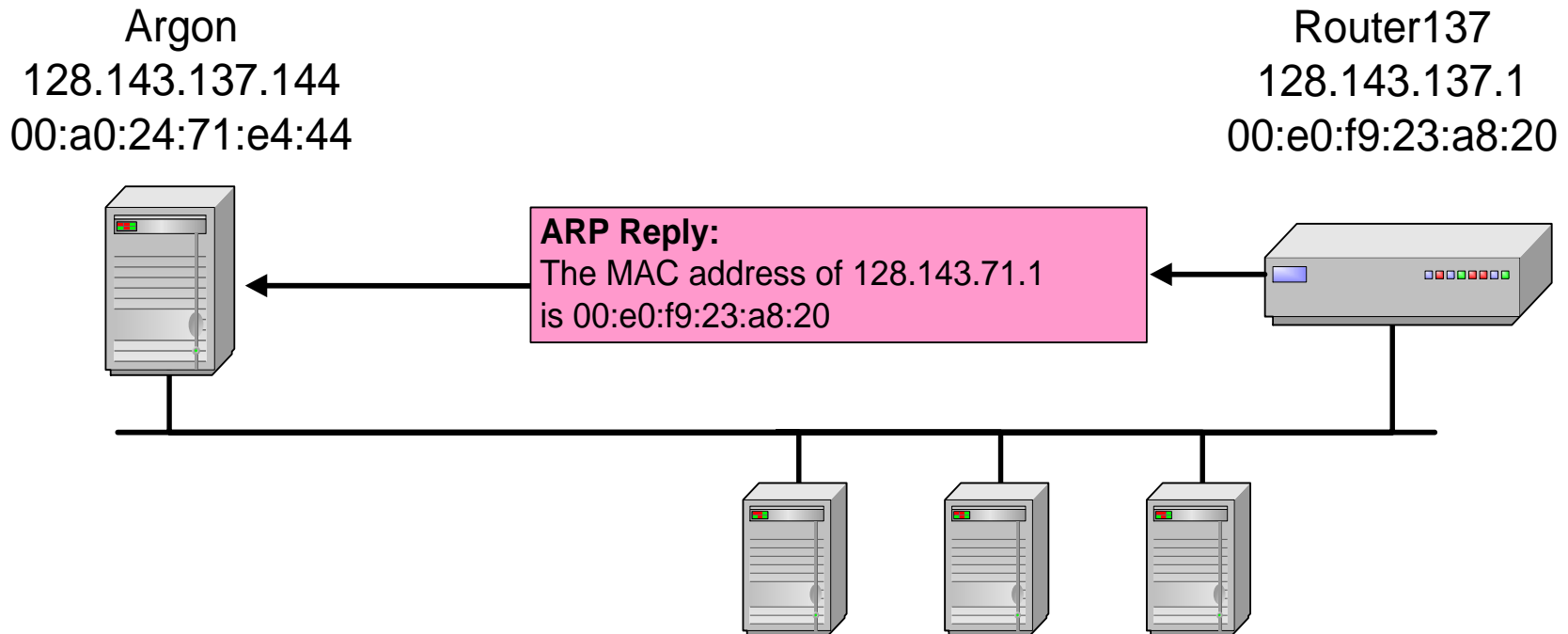
Router137  
128.143.137.1  
00:e0:f9:23:a8:20



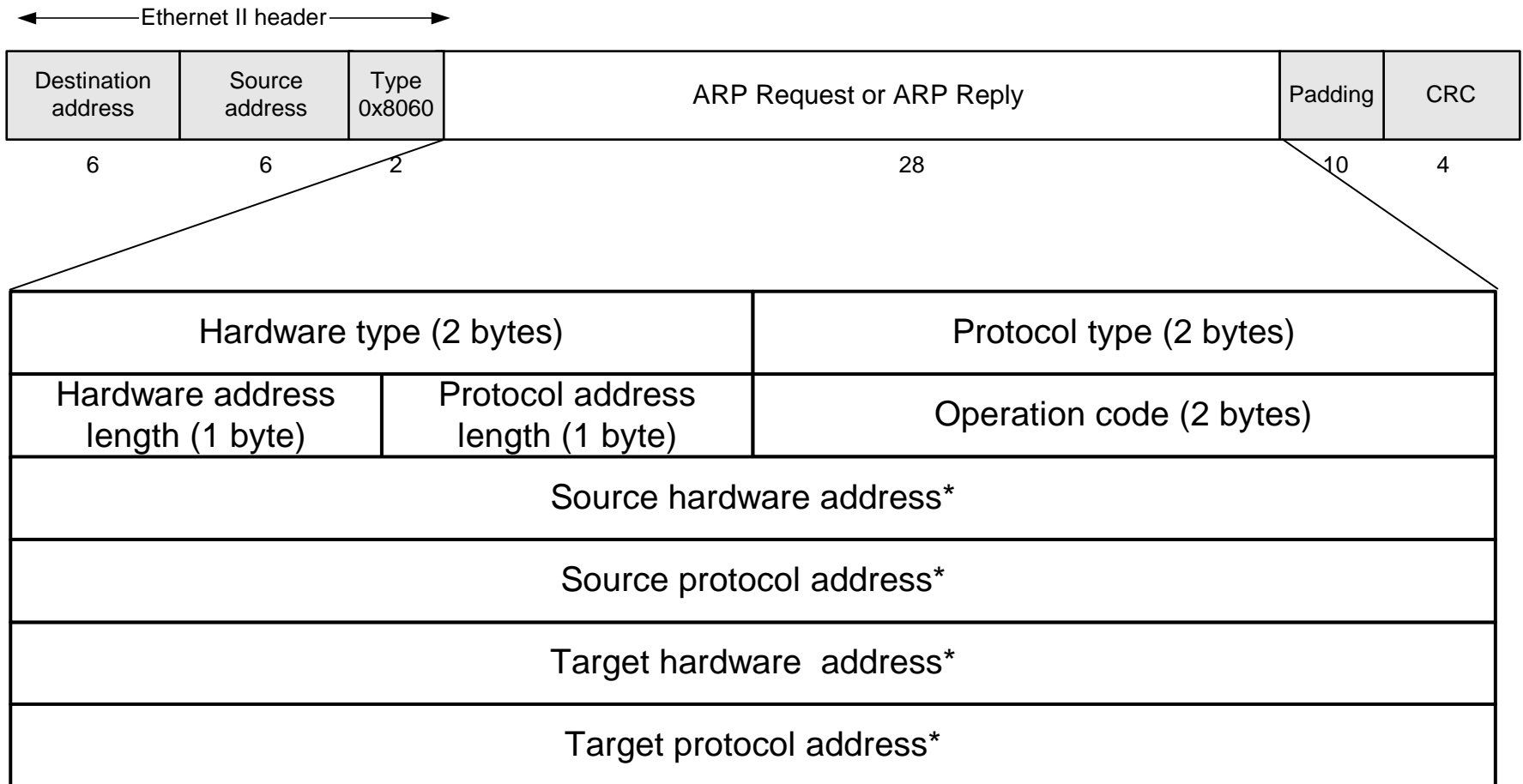
# Address Translation with ARP

## ARP Reply:

Router 137 responds with an ARP Reply which contains the hardware address



# ARP Packet Format



\* Note: The length of the address fields is determined by the corresponding address length fields

# Example

- *ARP Request from Argon:*

Source hardware address:	00:a0:24:71:e4:44
Source protocol address:	128.143.137.144
Target hardware address:	00:00:00:00:00:00
Target protocol address:	128.143.137.1

- *ARP Reply from Router137:*

Source hardware address:	00:e0:f9:23:a8:20
Source protocol address:	128.143.137.1
Target hardware address:	00:a0:24:71:e4:44
Target protocol address:	128.143.137.144

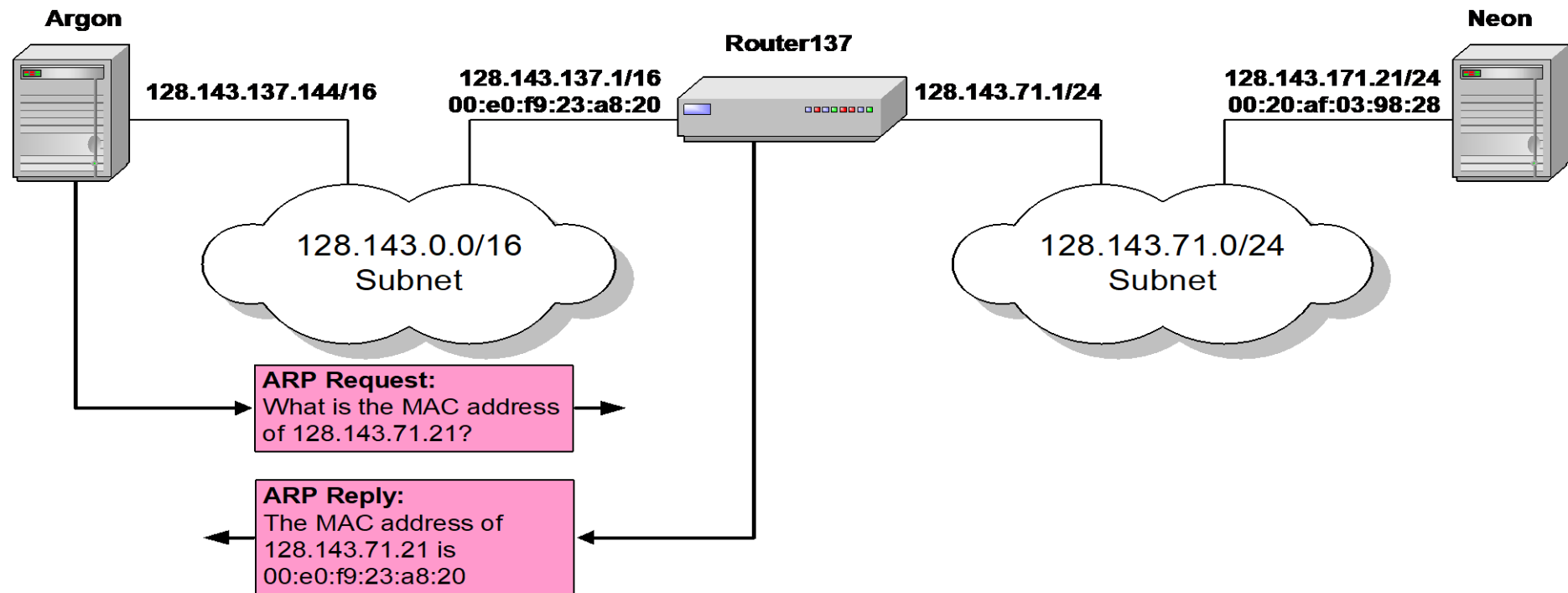


# ARP Cache

- Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a cache (ARP Cache) of current entries. The entries expire after 20 minutes.
- Contents of the ARP Cache:
  - (128.143.71.37) at 00:10:4B:C5:D1:15 [ether] on eth0
  - (128.143.71.36) at 00:B0:D0:E1:17:D5 [ether] on eth0
  - (128.143.71.35) at 00:B0:D0:DE:70:E6 [ether] on eth0
  - (128.143.136.90) at 00:05:3C:06:27:35 [ether] on eth1
  - (128.143.71.34) at 00:B0:D0:E1:17:DB [ether] on eth0
  - (128.143.71.33) at 00:B0:D0:E1:17:DF [ether] on eth0

# Proxy ARP

- **Proxy ARP:** Host or router responds to ARP Request that arrives from one of its connected networks for a host that is on another of its connected networks.



# Things to know about ARP

- What happens if an **ARP** Request is made for a non-existing host?  
Several ARP requests are made with increasing time intervals between requests. Eventually, ARP gives up.
- On some systems (including Linux) a host periodically sends ARP Requests for all addresses listed in the ARP cache. This refreshes the ARP cache content, but also **introduces traffic**.
- **Gratuitous** غير مبرر **ARP Requests**: A host sends an ARP request for **its own IP address**:
  - Useful for detecting if an IP address has already been assigned.

# Vulnerabilities of ARP

1. Since ARP does not authenticate requests or replies, **ARP Requests and Replies can be forged** مزور
2. ARP is **stateless**: ARP **Replies can be sent** without a corresponding **ARP** Request
3. According to the ARP protocol specification, **a node** receiving an **ARP packet** (Request or Reply) **must** update its local ARP cache with the information in the **source fields**, if the receiving node already has an entry for the IP address of the source in its ARP cache. (This applies for ARP Request packets and for ARP Reply packets)

## **Typical exploitation of these vulnerabilities:**

- A forged ARP Request or Reply can be used to update the ARP cache of a remote system with a forged entry (**ARP Poisoning**)
- This can be used to redirect IP traffic to other hosts