**Chapter 08: Address Resolution Protocol**

Table of Contents

[8.1 Address Mapping 3](#_Toc86833861)

[Static and Dynamic Mapping 3](#_Toc86833862)

[Operation 3](#_Toc86833863)

[ARP Packet 4](#_Toc86833864)

[Proxy ARP 6](#_Toc86833865)

[8.4 ARP Package 7](#_Toc86833866)

The **Address Resolution Protocol** (ARP) is used to **map logical addresses** (IP addresses) to **physical addresses** (MAC addresses). The destination MAC address is required when creating **frames** in the **data link layer**.

## 8.1 Address Mapping

### Static and Dynamic Mapping

**Address mapping** can be of two types, **static** or **dynamic**. Static mapping would be having a fixed table that does not change while dynamic mapping maintains a table which can be updated over time.

The main issue with static mapping is that the addresses are not fixed, which makes static mapping unusable. There are a multitude of reasons why either the physical address or the IP address could change, perhaps due to the network the device is on changing, or a mobile device being used or even the MAC address itself being changed at the hardware level (although this is less common).

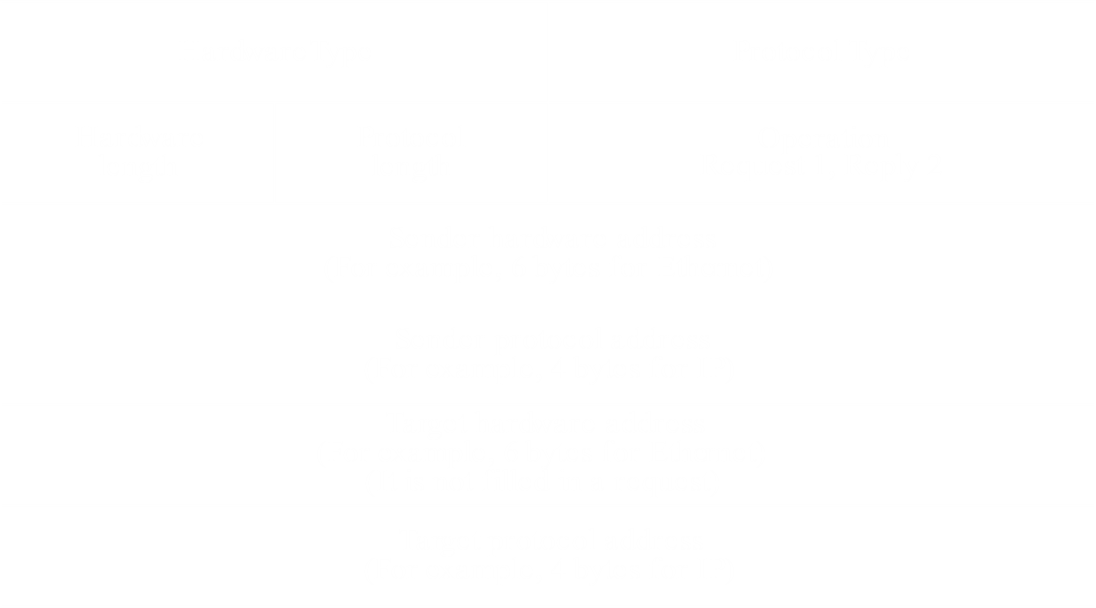
### Operation

In the **network layer**, once the IP packet is created, the **routing table** is checked. The **destination IP address** is used to determine what the IP address of the **next hop** is, and the **ARP protocol** is used to determine the **MAC address** of the next hop.

The ARP protocol performs **limited broadcasting**. It sends out a broadcast message, called the **ARP request**, with the query about what the MAC address of the specified IP address is, and the device with that IP address responds with a unicast message, called the **ARP reply**, containing its own MAC address.

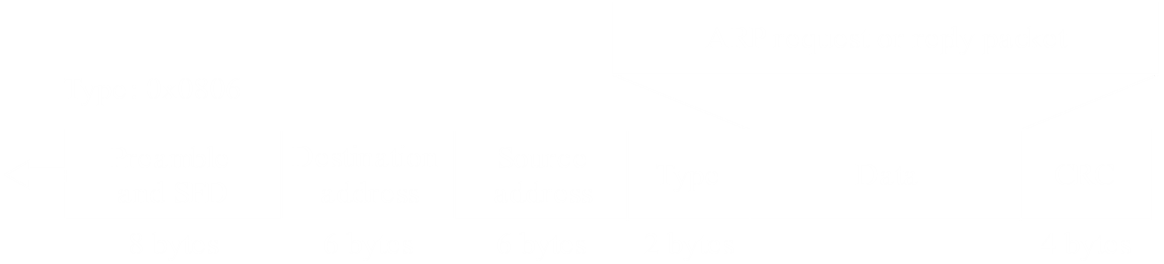
Note that the network layer gives the **complete IP packet** to the ARP protocol and the ARP protocol sends the packet along with the destination MAC address to the data-link layer.

### ARP Packet



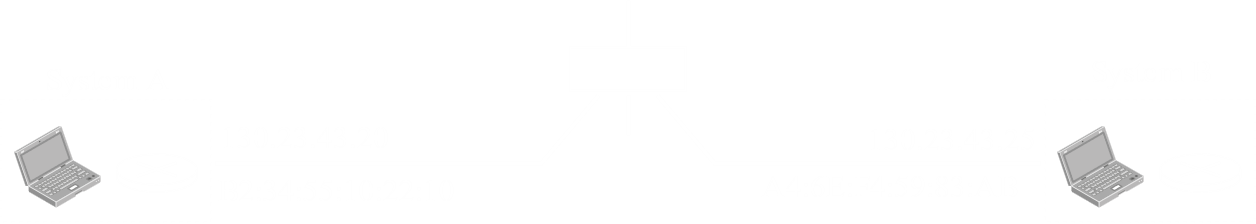
* **Hardware Type** – This is the type of the physical LAN being used.
* **Protocol Type** – This is the type of IP protocol being used.
* **Hardware Length** – The length of the physical address.
* **Protocol Length** – The length of the logical address.
* **Operation** – Denotes whether this is a request packet (value ) or a reply packet (value ).
* **Sender Hardware Address** – The MAC address of the sender.
* **Sender Protocol Address** – The IP address of the sender.
* **Target Hardware Address** – The MAC address of the receiver. This field is empty in the request packet.
* **Target Protocol Address** – The IP address of the receiver.

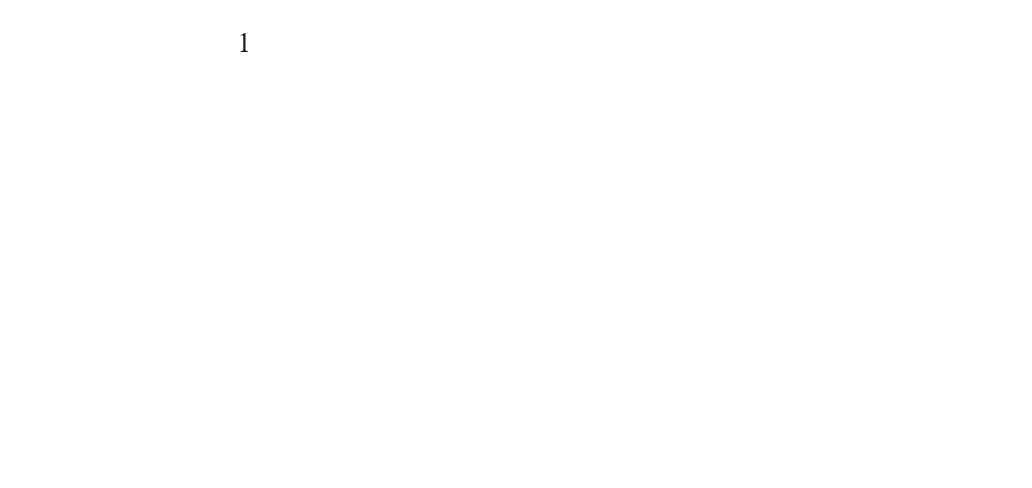
The ARP packet is encapsulated like this:

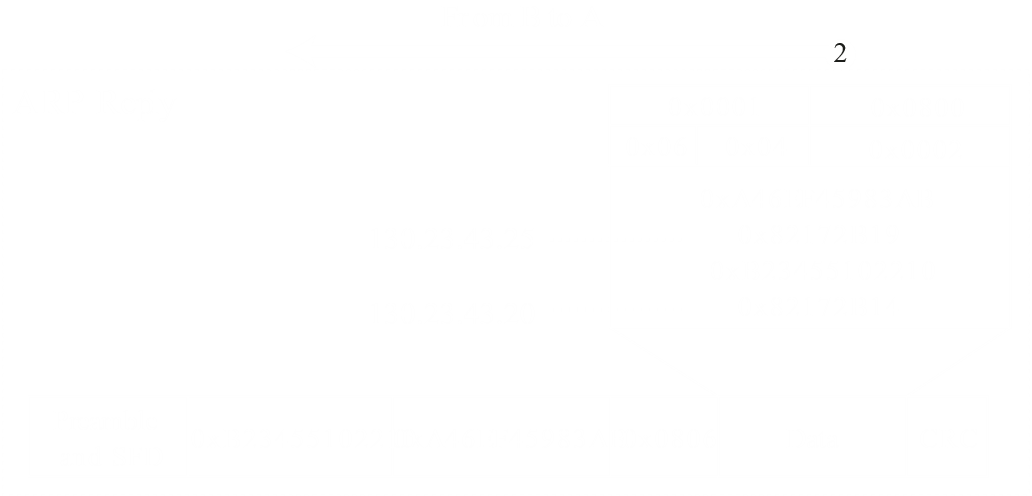


This is just an **ethernet frame**.

Example





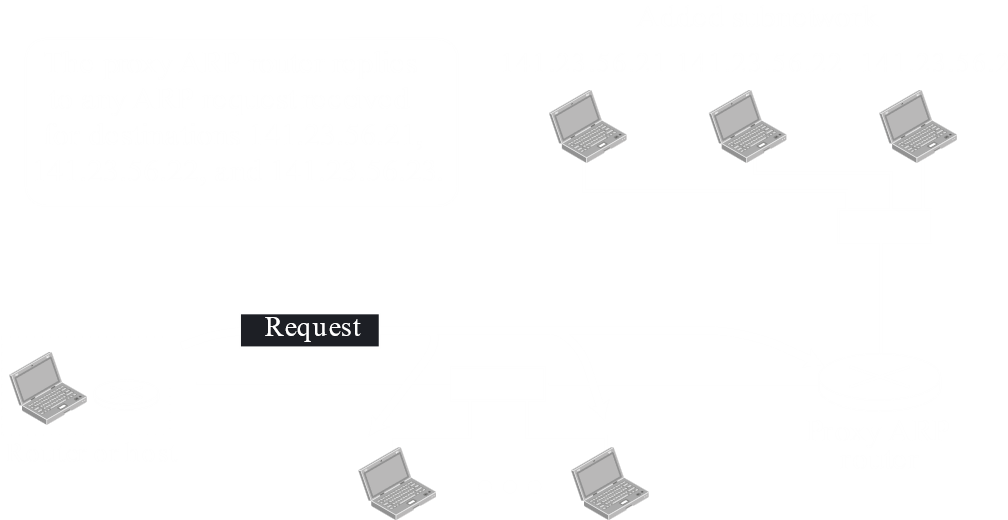


In the example above, in the **ARP reply**, notice that the target addresses become the original sender while the sender addresses become the original target. The positions of the values are **flipped**. This works because we have the **operation** field to determine how to interpret the four address fields.

In the **ethernet frame** as well, the MAC address positions are flipped. Also noticed that the destination MAC address in the ethernet frame of the ARP request is the **broadcast address**.

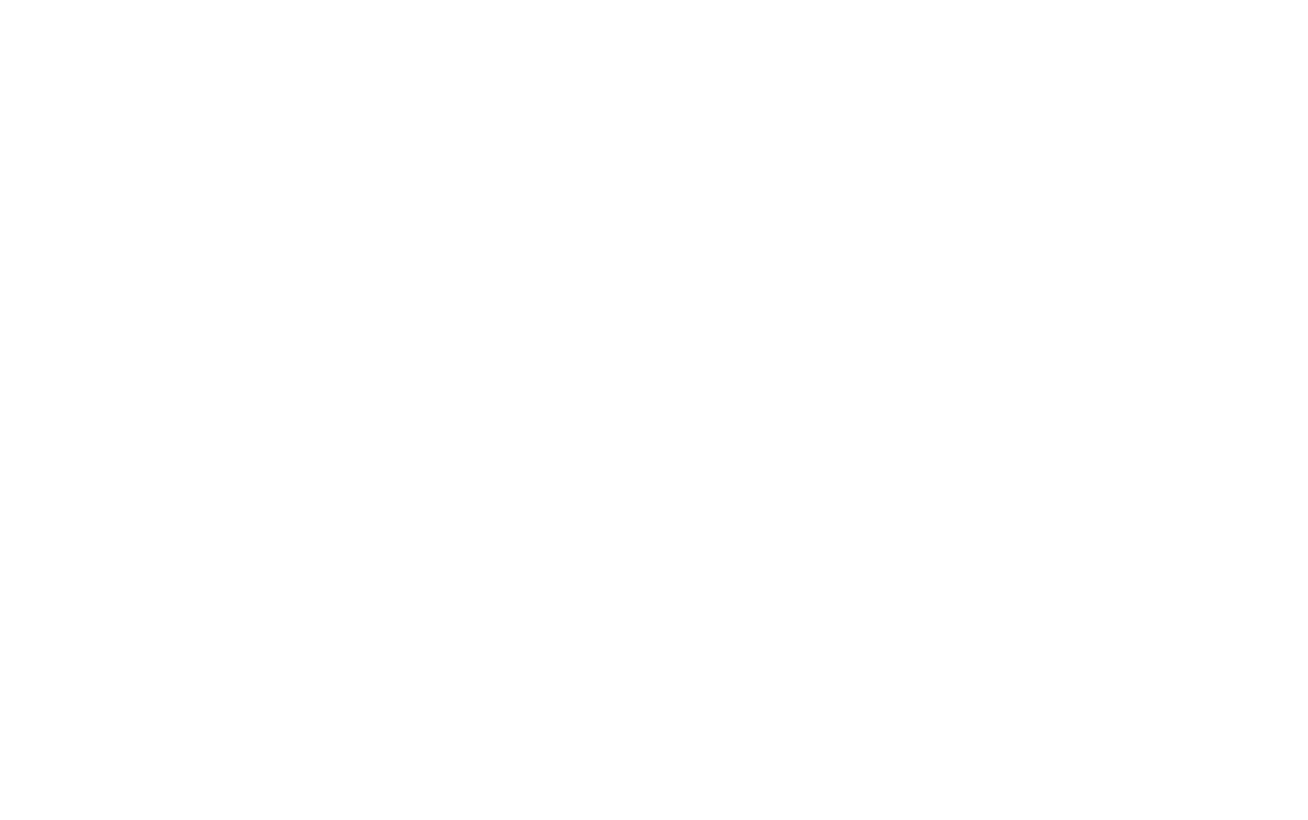
### Proxy ARP

The process of ARP can be used to create a **proxy router** that essentially results in a **subnetting effect**.



## 8.4 ARP Package

The entire process of ARP consists of a few modules working together.



The process of using an IP address to find a MAC address only happens **once**. When we have the MAC address, it is stored in a **cache table** for future use.

1. When an IP packet first comes from the upper layers, it is received by the **output module**. The first thing it does is look in the **cache table**. If the required address is not in the cache table, it needs to send out an **ARP request packet**. At this point, the IP packet is stored in a **queue**. There is a separate queue for each destination address.
2. The cache table must be updated from time to time since a particular IP address and MAC address combination is not permanent. To deal with this, the cache table has three **states** for each entry, free, resolved and pending. The **free** state indicates that the entry has expired and can be replaced by a different entry. The **resolved** state means that the entry has not expired. The **pending** state means that an ARP request has been made but a response has not yet been received.
3. After a specific interval of time, the cache table is checked again. This is being done by the **cache-control module**.

For **pending entries**, another ARP request is sent (in case the reply was not received by error). There is however, a **limit** to the number of attempts that will be made to do this. After the limit is crossed, the entry is set to free and the corresponding queue is destroyed.

For **resolved entries**, the **time-out** value is decremented. If the time-out becomes , the entry is set to free and the corresponding queue is destroyed.

1. The **input module** deals with incoming ARP packets. If the incoming packet is an ARP reply, then it updates the cache table and starts dequeuing the corresponding queue to send them to the data-link layer. If the incoming packet is an ARP request, then it sends back an ARP reply.