Lecture 4.1

Network Layer
Address Mapping Protocols

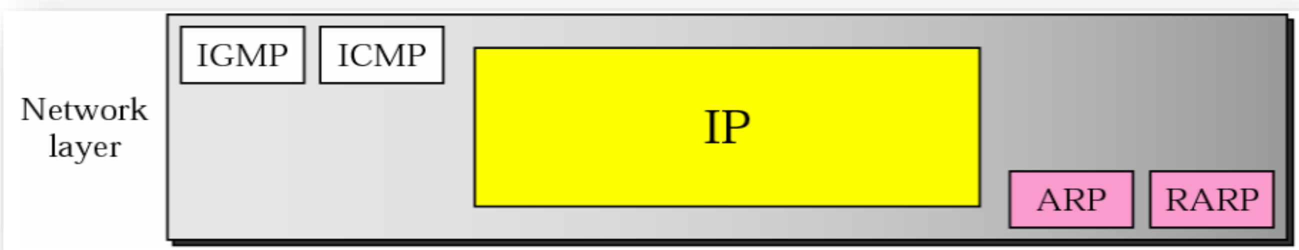
Dr. Vandana Kushwaha

Department of Computer Science
Institute of Science, BHU, Varanasi

INTRODUCTION

- **IP** (Internet Protocol) was designed as a **best-effort delivery protocol**, but it **lacks some features** such as **flow control** and **error control**.
- To make **IP** more **responsive** it takes help of other protocols.
- **Protocols** to create a **mapping a logical address to a physical address**: **ARP** (Address Resolution Protocol).
- Protocols to create a **reverse mapping** i.e. **mapping** a **physical address to a logical address**: **RARP, BOOTP**, and **DHCP**.
- Lack of **flow** and **error control** in the **Internet Protocol** has resulted in another protocol, **ICMP**.
- **ICMP reports congestion** and some types of **errors** in the network or destination host.

Address Mapping & Error Reporting Protocols



ADDRESS MAPPING

- An **internet** is made of a combination of **physical networks** connected by **internetworking devices** such as **routers.**
- A **packet** starting from a **source host** may pass through several different **physical networks** before finally reaching the **destination host.**
- The **hosts** and **routers** are **recognized** at the **network level** by their **logical (IP) addresses**, while at the **physical level**, they are **recognized** by their **physical (MAC) addresses.**
- Thus **delivery** of a **packet** to a **host** or a **router** requires **two levels of addressing**: **logical (IP)** and **physical (MAC).**
- We need to be able to **map** a **logical address** to its **corresponding** **physical address** and **vice versa**.
- These can be done by using either **static** or **dynamic mapping.**

Static mapping

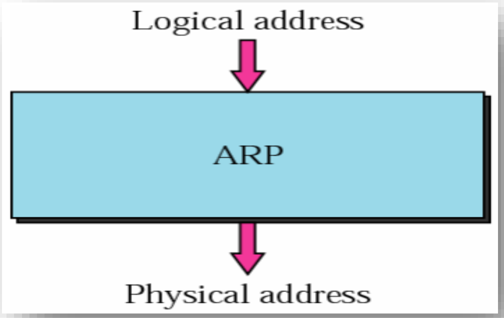
- **Static mapping** involves in the creation of a **table** that **associates** a **logical(IP) address with a physical(MAC) address.**
- This **table** is stored in **each machine** on the **network.**
- **Static mapping** has some **limitations** because **physical addresses** may **change** in the following ways:
  - A **machine** could **change** its **NIC** (Network Interface Card), resulting in a **new physical address**.
  - In some **LANs**, such as **LocalTalk,** the **physical address changes** every time the computer is **turned on.**

Dynamic mapping

In **Dynamic mapping** each time a machine knows one of the **two addresses** (**logical** or **physical**), it can use a **protocol** to find the other one.

Mapping Logical to Physical Address: ARP

- **ARP** stands for **Address Resolution Protocol** which is one of the most important protocols of the **Network layer** in the **TCP/IP model.**
- **ARP** finds the **physical address**, also known as **Media Access Control (MAC) address**, of a **host** from its **known IP address**.

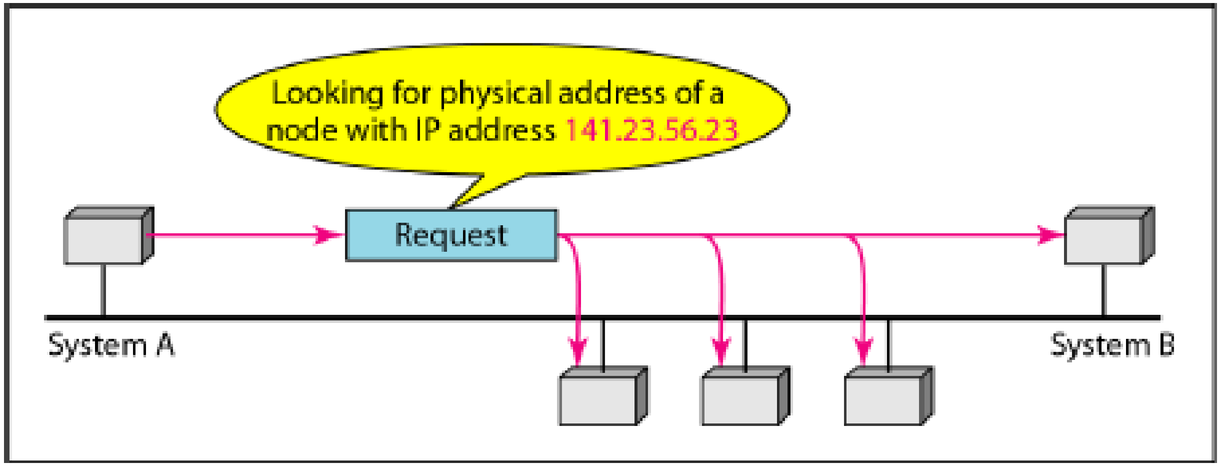


Mapping Logical to Physical Address: ARP

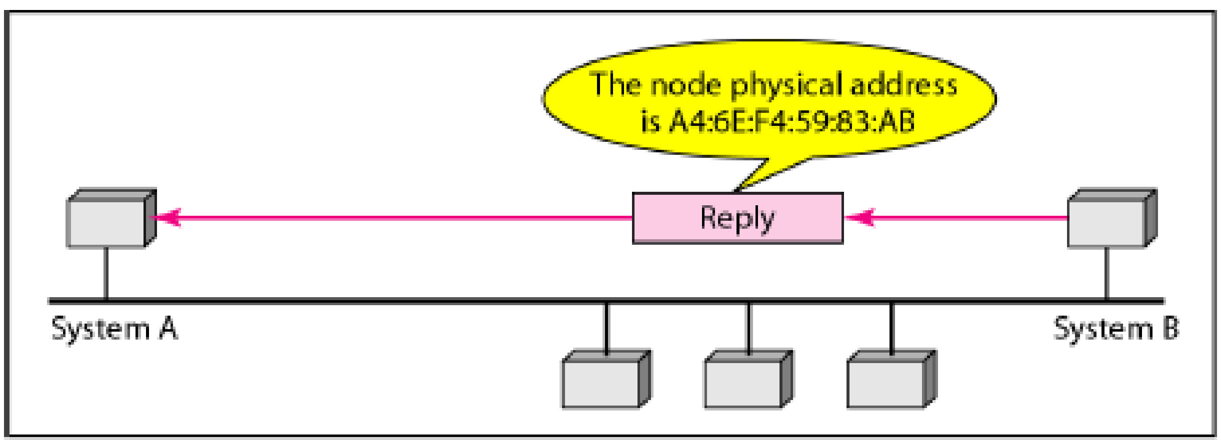
Following **steps** are involved in **logical to physical address mapping:**

- The **host** or the **router** sends an **ARP query packet**.
- The **ARP query packet** includes the **Physical** and **IP addresses** of the **sender** and the **IP address** of the **receiver.**
- As the **sender** **does not know** the **physical address** of the **receiver**, the **ARP query is broadcast** over the **network**.
- Every **host** or **router** on the **network** receives and processes the **ARP query packet,** but only the **intended recipient** **recognizes** its **IP address** and **sends back** an **ARP response packet**.
- The **ARP response packet** contains the **recipient's IP** and **physical addresses**.
- The **ARP response packet** is **unicast** **directly to the** **inquirer** (host/router) by using the **physical address** received in the **query packet.**

Mapping Logical to Physical Address: ARP



a. ARP request is broadcast

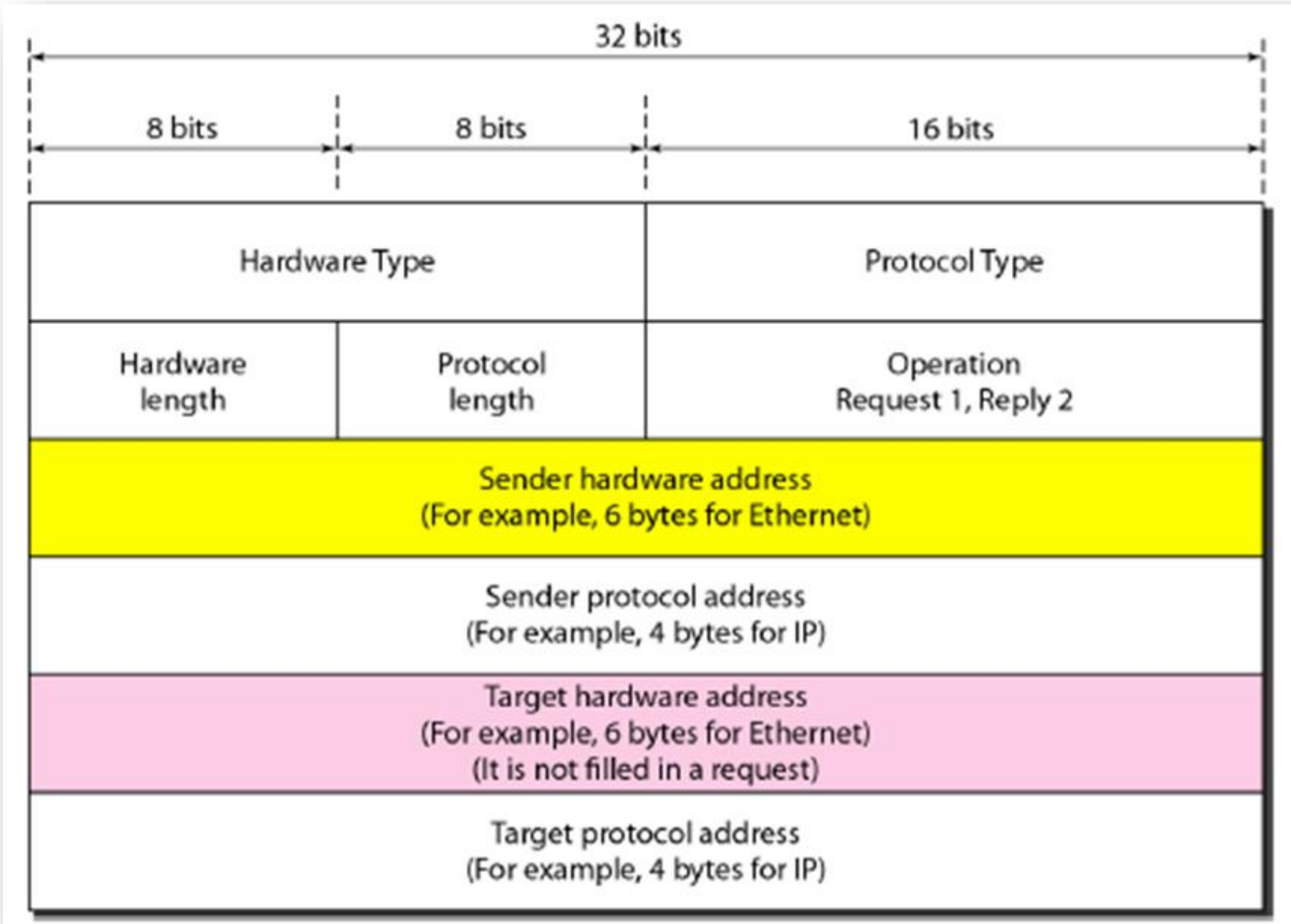


b. ARP reply is unicast

ARP Cache

- Using **ARP** is **inefficient** if **system A** needs to **broadcast** an **ARP request** for **each IP packet** it needs to send to **system B.**
- **ARP** can be useful if the **ARP reply is cached** (kept in **cache memory** for a while) because a system normally sends **several packets** to the **same destination.**
- A **system** that receives an **ARP reply** stores the **mapping** in the **cache memory** and keeps it for **20 to 30 minutes** unless the space in the **cache** is **exhausted.**
- Before sending an **ARP request**, the system **first checks its cache** to see if it can find the **mapping**.

ARP Packet Format



ARP Packet Format

a. **Hardware type.** This is a **16-bit** field **defining the type of the network** on which **ARP** is running. Each **LAN** has been assigned an **integer** based on its **type**. For example, **Ethernet is given type 1**.

b. **Protocol type.** This is a **16-bit** field defining the **protocol**. For **example**, the value of this field for the **IPv4** protocol is **080016.**

c. **Hardware length.** This is an **8-bit** field defining the **length** of the **physical address** in **bytes**. For **example,** for **Ethernet** the value is **6.**

d. **Protocol length.** This is an **8-bit** field defining the **length** of the **logical address** in **bytes**. For **example**, for the **IPv4 protocol** the value is **4.**

e. **Operation.** This is a **16-bit** field defining the **type of packet**. Two packet types are defined: **ARP request (1)** and **ARP reply (2)**.

ARP Packet Format

f. Sender hardware address. This is a **variable-length field** defining the **physical address** of the **sender**. For **example,** for **Ethernet** this field is **6 bytes** long.

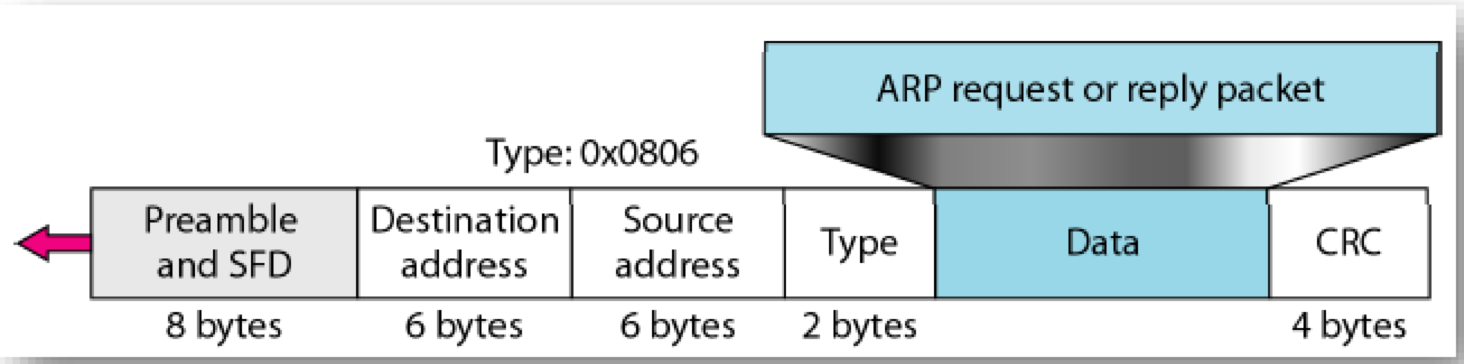
g. Sender protocol address. This is a **variable-length field** defining the **logical (for example, IP) address** of the **sender.** For the **IPv4 protocol**, this field is **4 bytes** long.

h. Target hardware address. This is a **variable-length field** defining the **physical address of the target**. For **example**, for **Ethernet** this field is **6 bytes long**. For an **ARP request** message, this field is **all 0s** because the **sender does not know** the **physical address** of the **target.**

i. Target protocol address. This is a variable-length field defining the **logical (for example, IP) address** of the **target**. For the **IPv4 protocol**, this field is **4 bytes** long.

Encapsulation

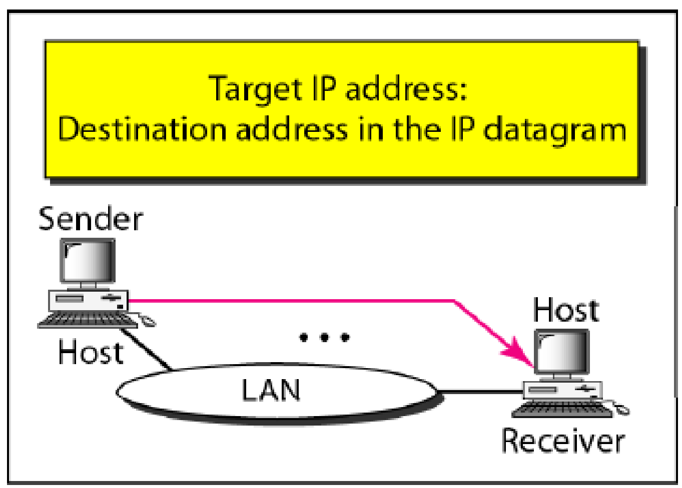
- An **ARP** packet is **encapsulated directly** into a **data link frame**.
- For **example**, in Figure an **ARP packet** is **encapsulated** in an **Ethernet frame.**
- Note that the **type field** indicates that the **data** carried by the **frame** are an **ARP packet.**



Different Cases of ARP Operation

Case 1

- *The **sender** is a **host** and wants to **send** a packet to **another host** on the **same network**.*
- In this case, the **logical address** that must be **mapped** to a **physical address** is the **destination** **IP address** in the datagram header.



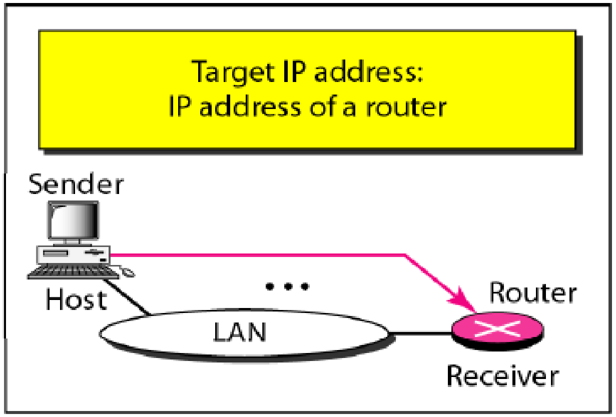
Case 1. A host has a packet to send to another host on the same network.

Different Cases of ARP Operation

Case 2

*The **sender** is a **host** and wants to **send** a **packet** to **another host** on **another network**.*

- In this case, the **host** looks at its **routing table** and finds the **IP address** of the **next hop (router)** for this **destination**.
- If it **does not** have a **routing table**, it looks for the **IP address** of the **default router**.
- The **IP address** of the **router** becomes the **logical address** that must be **mapped** to a **physical address**.



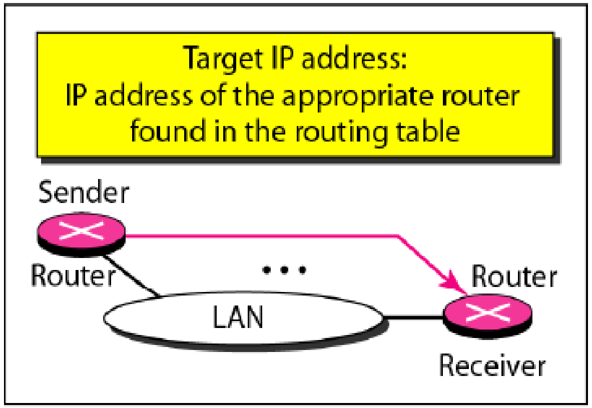
Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.

Different Cases of ARP Operation

Case 3

*The **sender** is a **router** that has **received** a **datagram** **destined** for a **host** on **another network.***

- **Router** checks its **routing table** and finds the **IP address** of the **next router**.
- The **IP address** of the **next router** becomes the **logical address** that must be **mapped** to a **physical address.**



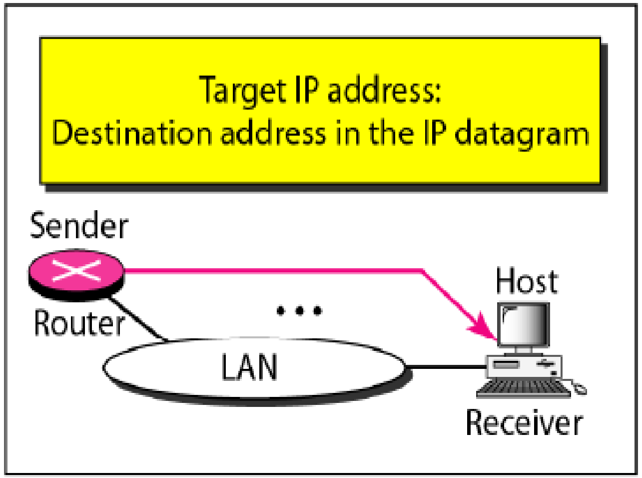
Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

Different Cases of ARP Operation

Case 4

*The **sender** is a **router** that has **received** a **datagram** destined for a **host on the same network.***

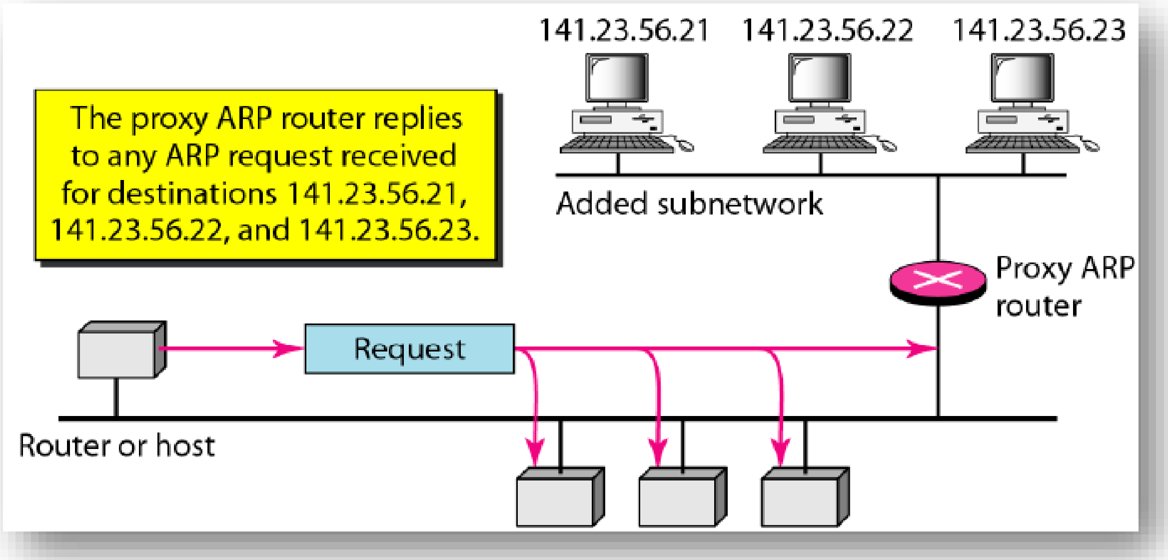
- The **destination IP address of the datagram** becomes the **logical address** that must be **mapped** to a **physical address**.



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

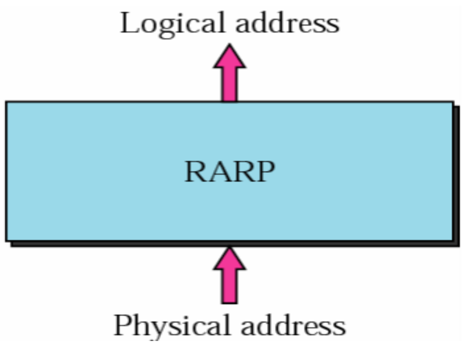
Proxy ARP

- A **proxy ARP** is an **ARP** that acts **on behalf** of a **set of hosts**.
- Whenever a **router** running a **proxy ARP** receives an **ARP request** looking for the **IP address** of one of these hosts, the **router sends** an **ARP reply** announcing **its own hardware (physical) address**.
- After the **router** receives the **actual IP packet**, it **sends** the **packet** to the **appropriate host** or **router**.



Mapping Physical Address to Logical Address

- There are **occasions** in which a **host knows** its **physical address, but needs to know its logical address**. This may happen in **two cases**:
- **Case 1:** *A **diskless station** is just **booted***. The station can find its **physical address** by checking its **interface,** but it does **not know** its **IP address**.
- **Case 2:** *An organization does **not have enough IP addresses** to assign to each **station***; it needs to **assign IP addresses on demand**. The **station** can **send** its **physical address** and ask for a **short time lease**.



Reverse Address Resolution Protocol (RARP)

- **Reverse Address Resolution Protocol** (**RARP**) finds the **logical address** for a **machine** that knows only its **physical address.**
- To **create** an **IP datagram**, a host or a router **needs** to **know** its **own IP address** .
- The **IP address** of a **machine** is usually **read from** its **configuration file stored** on a **disk file.**
- However, a **diskless machine** is usually **booted** from **ROM**, which has minimum booting information.
- The **ROM** is installed by the **manufacturer.**
- It **cannot include the IP address** because the **IP addresses** on a network are **assigned** by the **network administrator**.

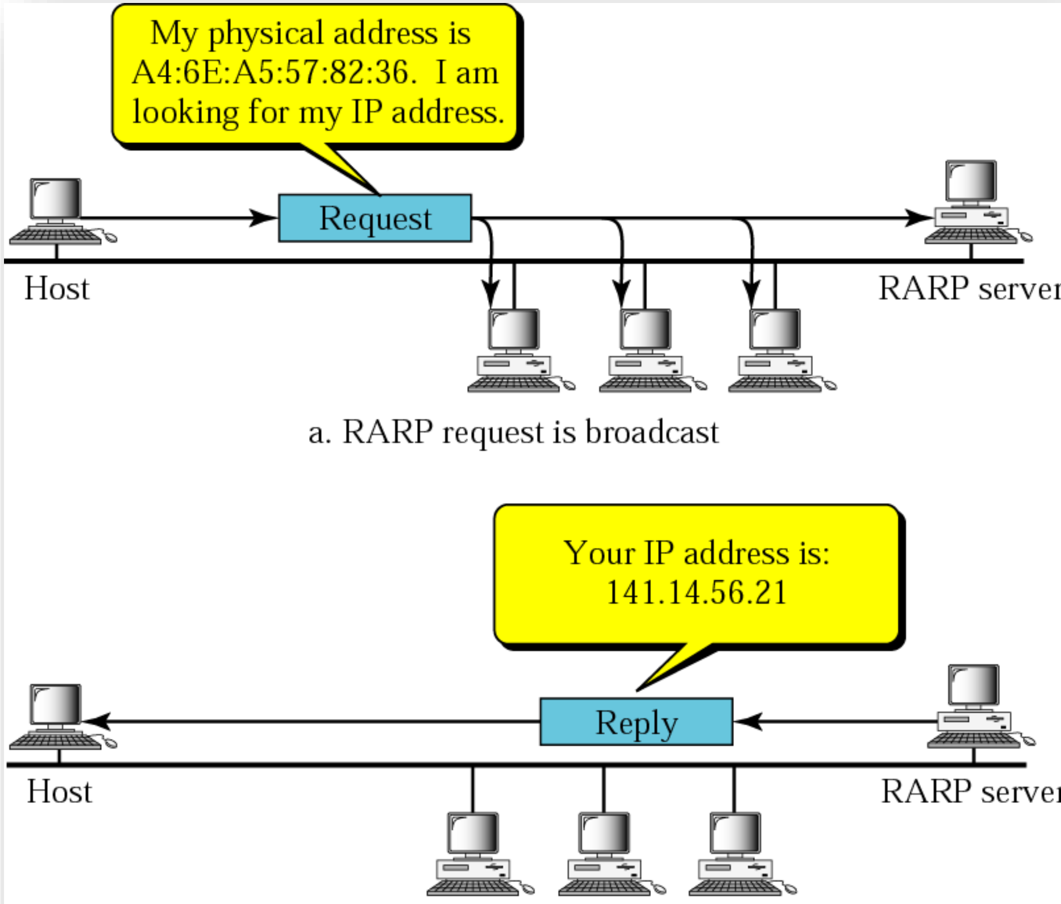
Reverse Address Resolution Protocol (RARP)

- The **machine** can get its **physical address** (by reading its **NIC**, for example), which is unique locally.
- It can then **use** the **physical address** to get the **logical address** by using the **RARP protocol**.

RARP Operation

- A **RARP request** is **created** and **broadcast** on the **local network.**
- Another **machine** on the **local network** that **knows all the IP addresses** will respond with a **RARP reply**.
- The **requesting machine** must be running a **RARP client** program; and the **responding machine** must be running a **RARP server** program.

RARP Operation



a. RARP request is broadcast

b. RARP reply is unicast

RARP Packet Format & Encapsulation

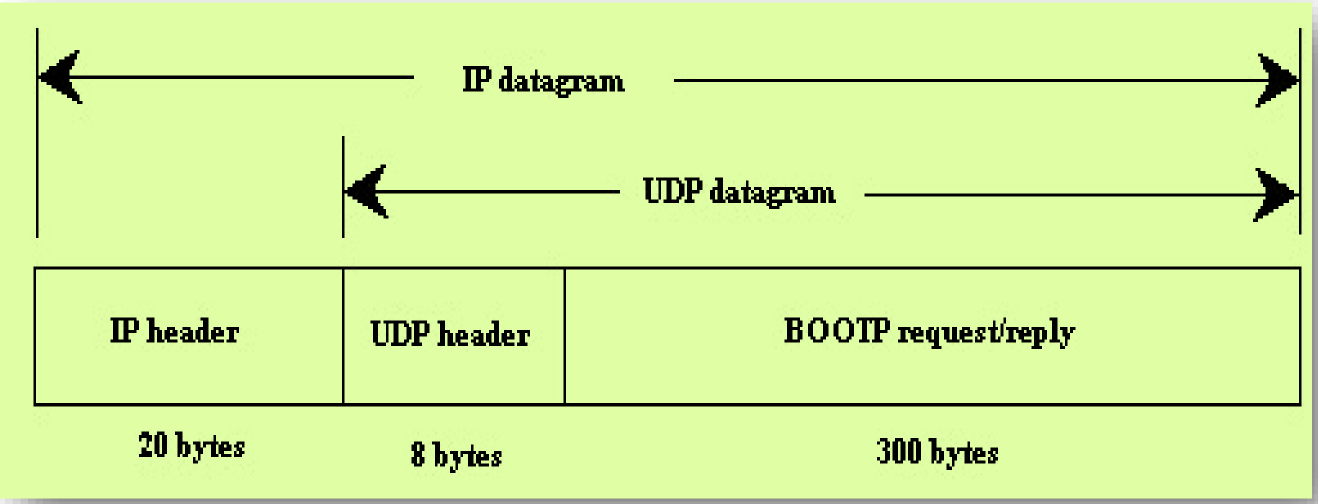
- The format of the **RARP packet** is the **same as the ARP packet** format, except that the **Operation field**.
- It's value is **3** for **RARP request** message and **4** for **RARP reply** message.
- An **RARP packet** is also **encapsulated** directly into a **data link frame** just like **ARP** packet.

Limitations of RARP

- As **broadcasting** is **done** at the **data link layer**.
- The **physical broadcast address**, all **1's** in the case of **Ethernet,** does not pass the boundaries of a network.
- This means that if an **administrator** has **several networks** or **several subnets**, it needs to **assign** a **RARP server** for each **network** or **subnet**.
- This is the **reason** that **RARP is almost obsolete**.
- Two protocols, **BOOTP** and **DHCP**, are replacing **RARP**.

Bootstrap Protocol (BOOTP)

- The **Bootstrap Protocol (BOOTP)** is a **Client/Server** based **protocol** at **Application layer**, designed to **provide physical address** to **logical address mapping**.
- The **administrator** may put the **client** and the **server** on the **same network** or on **different networks.**
- **BOOTP** messages are **encapsulated** in a **UDP packet**, and the **UDP packet** itself is encapsulated in an **IP packet.**



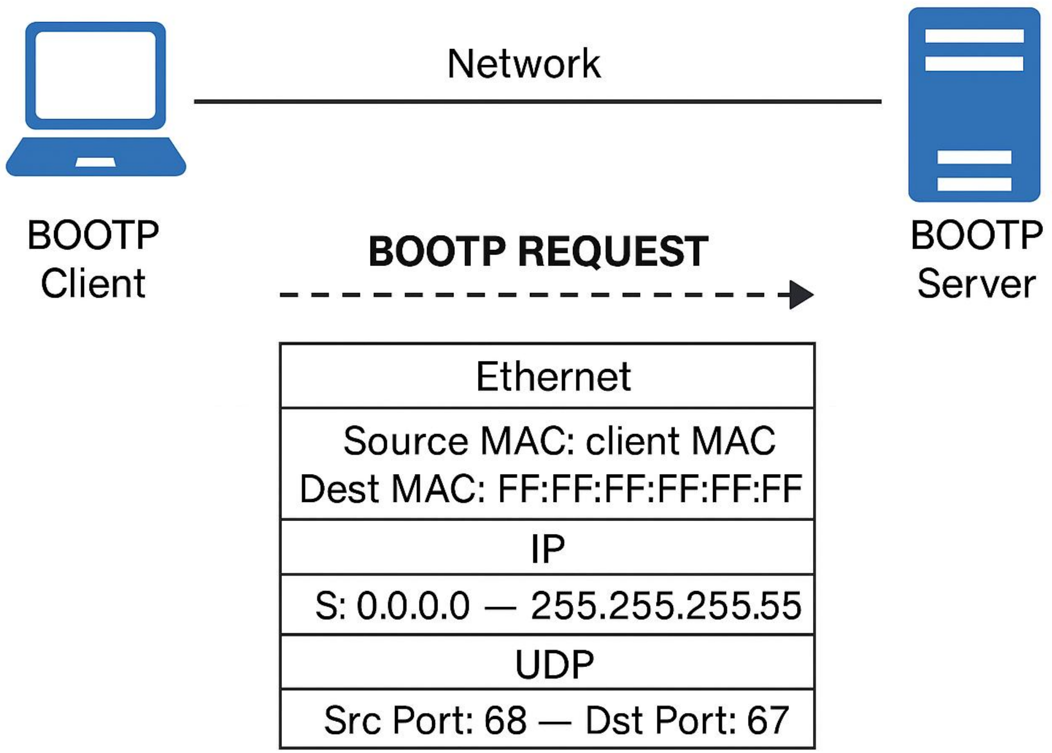
Bootstrap Protocol (BOOTP)

- ***Case 1: Client and server on same network***
- When a **BOOTP client** is started, it has **no IP address**, so it **broadcasts** a **message** containing its **MAC address** onto the network.
  - A **BOOTP client** (e.g., diskless workstation or network device) **doesn't know** its **own IP address** when it starts.
  - It sets its **source IP address** = **0.0.0.0** in the **request**, because it has no configured IP.
  - The **destination IP address** = **255.255.255.255** **(broadcast).**
- This message is called a "**BOOTP request**".

Bootstrap Protocol (BOOTP)

- ***Case 1: Client and server on same network***
- **Encapsulation** of **BOOTP request** at data link layer
  - Since the **client** doesn’t know the **server’s MAC**, it uses **Ethernet broadcast**.
  - **Destination MAC** = **FF:FF:FF:FF:FF:FF**
  - **Source MAC** = client’s **own MAC address** (unique and known).

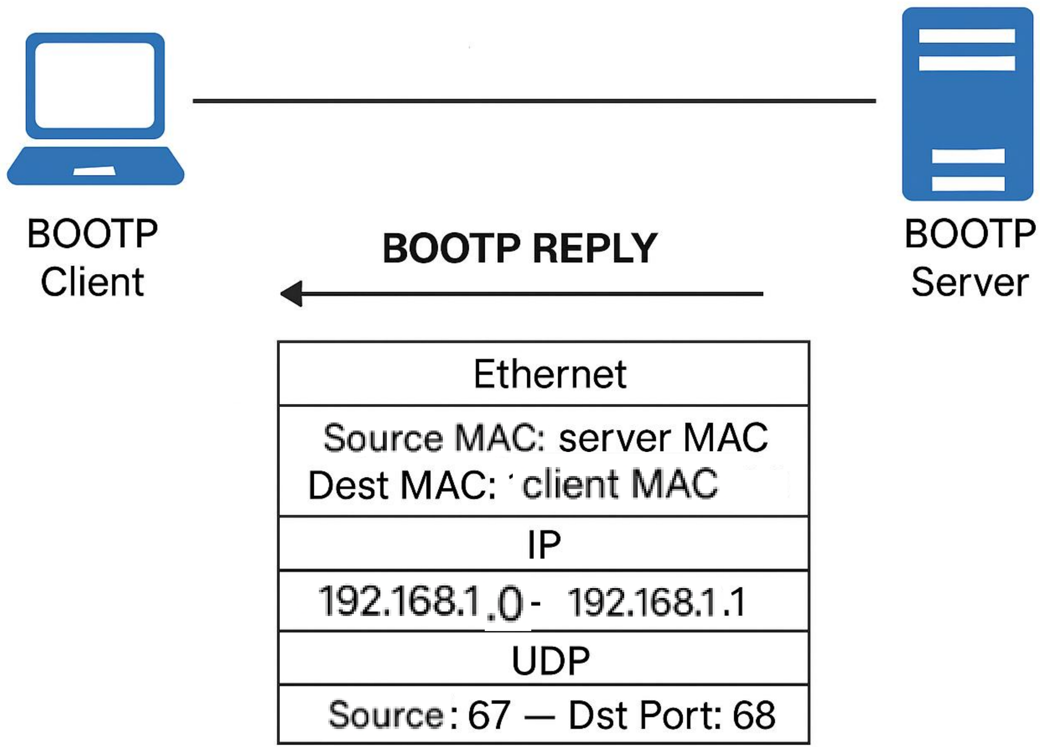
BOOTP client and server on the same network



Bootstrap Protocol (BOOTP)

- ***Case 1: Client and server on same network***
- "**BOOTP request**," is picked up by the **BOOTP server,** which **replies** to the **client** with the following information that the client needs:
  1. The **client's IP address**, **subnet mask**, and **default gateway address**.
  2. The **IP address** and **host name** of the **BOOTP server**.
- When the **client** **receives** this information from the **BOOTP server**, it **configures** and **initializes** its **TCP/IP protocol stack**, and then **connects** to the **server.**

Bootstrap Protocol (BOOTP)



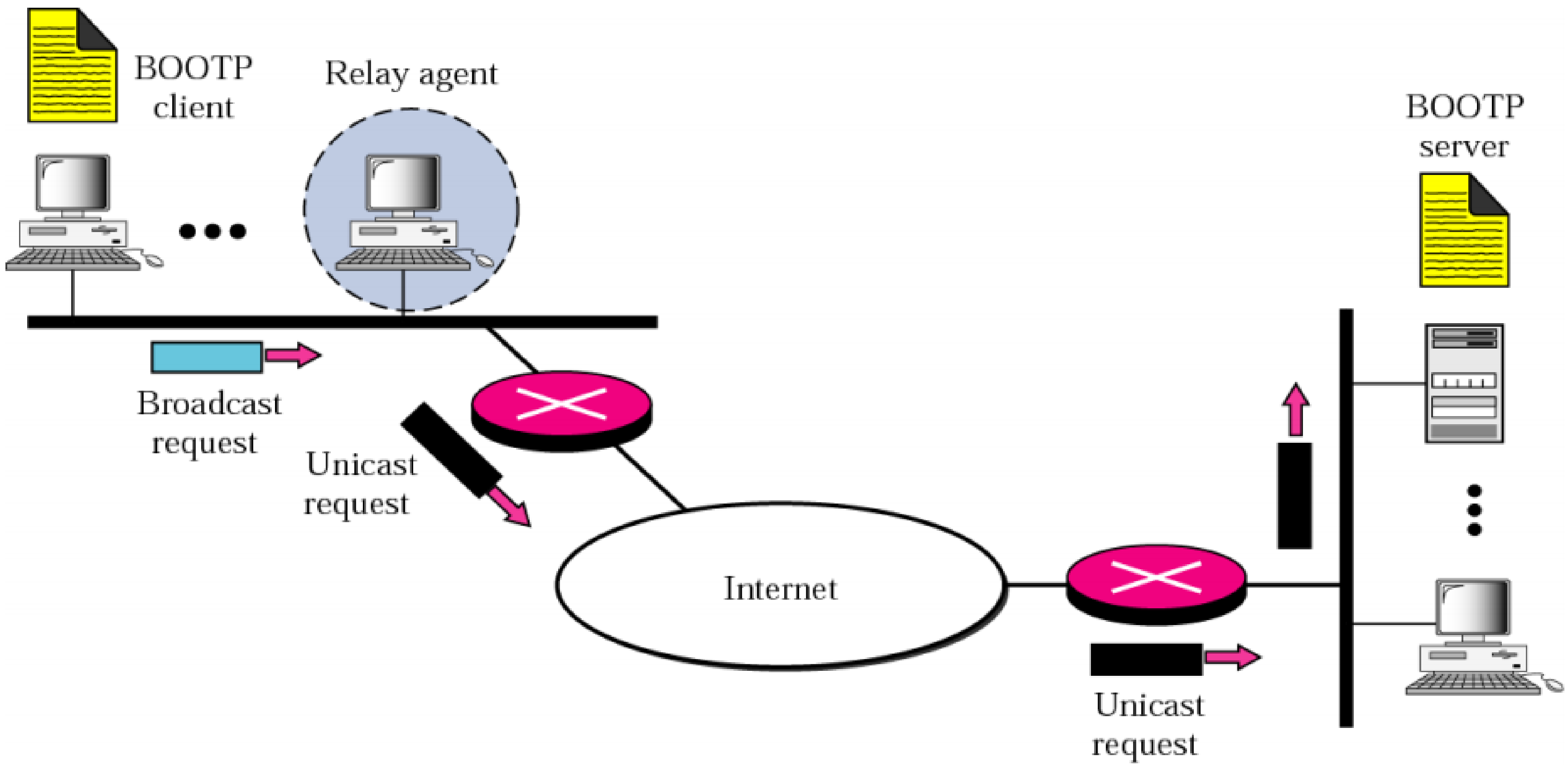
Bootstrap Protocol (BOOTP)

- ***Case 2 : Client and server on different networks***
- The **BOOTP request** is **broadcasted** because the **client** does not know the **IP address** of the **server**.
- The client simply uses **all 0's** as the **source address** and **all 1's** as the **destination address**.
- **But** a **broadcast** **IP datagram** **cannot pass** **through any router**.
- To **solve** the **problem**, there is a need for an **intermediary**.
- **One** of the **hosts** in **local network** (or a **router** that can be configured to operate at the **application layer**) can be **used** as a **relay** .
- The **host** in this case is called a **Relay Agent**.

Bootstrap Protocol (BOOTP)

- ***Case 2 : Client and server on different networks***
- The **relay agent knows the unicast address** of a **BOOTP server**.
- When it **receives** this type of packet, it **encapsulates** the message in a **unicast datagram** and **sends** the **request** to the **BOOTP server**.
- The packet, carrying a **unicast destination address**, is **routed** by **any router** and reaches the **BOOTP server**.
- The **BOOTP server** knows the message comes from a **relay agent** because one of the fields in the request message defines the **IP address** of the **relay agent.**
- **BOOTP server** sends a **BOOTP reply** message to the **relay agent**.
- The **Relay Agent**, after receiving the **reply**, **sends it** to the **BOOTP client**.

BOOTP client and server on different network



Limitations of BOOTP

- **BOOTP** is **not a Dynamic configuration** protocol.
- When a **client** requests its **IP address**, the **BOOTP server** consults a **table** that matches the **physical address** of the **client** with its **IP address.**
- This implies that the **binding** between the **physical address** and the **IP address** of the client **already exists**.
- The **binding is predetermined** i.e. **Static.**
- However, what if a **host moves** from **one physical network** to **another?**
- What if a **host wants** a **temporary IP address?**
- **BOOTP cannot handle these situations** because the **binding** between the **physical** and **IP addresses** is **static and fixed** in a **table** until changed by the **administrator.**
- **Thus BOOTP** is a **Static configuration** protocol.

Dynamic Host Configuration Protocol (DHCP)

- The **Dynamic Host Configuration Protocol (DHCP)** has been devised to provide **Static** and **Dynamic Address Allocation** that can be manual or automatic as required.

Static Address Allocation

- In this capacity **DHCP acts as BOOTP** does.
- It is **backward compatible with BOOTP,** which means a **host** running the **BOOTP client** can request a **static address** from a **DHCP server**.
- A **DHCP server** has a **static database** that **Statically binds Physical addresses to IP addresses**.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Address Allocation

- **DHCP** has a **Second database with a pool** of **available IP addresses**.
- This **second database** makes **DHCP dynamic.**
- When a **DHCP client requests** a **temporary IP address**, the **DHCP server** goes to the **pool of available (unused) IP addresses** and **assigns an IP address** for a **negotiable period** of **time.**
- When a **DHCP client** sends a **DHCP request** to a **DHCP server**, the server **first checks its Static database**.
- If an **entry** with the requested **physical address** exists in the **Static database**, the permanent **IP address** of the **client** is assigned.

Dynamic Host Configuration Protocol (DHCP)

- On the other hand**, if the entry does not exist** in the **Static database**, the **server** selects an **IP address** from the **available pool, assigns** the **address** to the **client**, and **adds** the entry to the **Dynamic database.**
- The **Dynamic** aspect of **DHCP** is **needed** when a **host moves from network to network** or is connected and disconnected from a network.
- **DHCP** provides **temporary IP addresses** for a **limited time**.
- The **DHCP server** issues a **lease for a specific time**.
- When the **lease expires**, the **client** must either **stop** using the **IP address** or **renew** the **lease.**
- The **server** has the **option** to **agree or disagree** with the **renewal.**
- If the **server disagrees**, the **client stops** using the **address.**