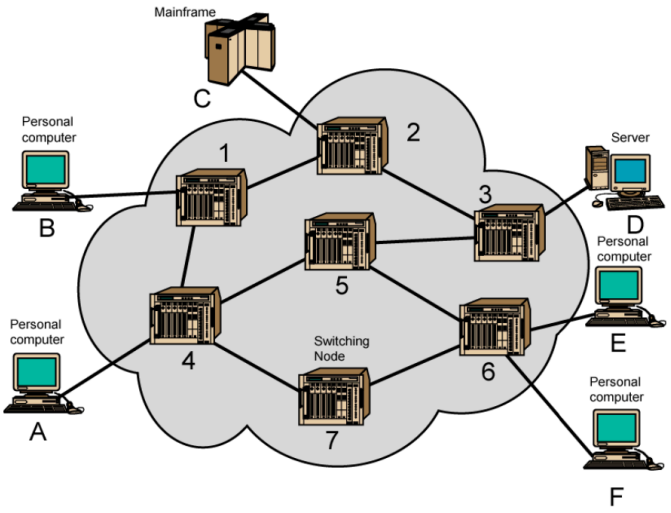
# lecture 8 Network Layer - Addressing

## Switch Networks

---

### Switch

- ISP connect a large number of users to the internet and among themselves over long distances
- Share a medium is **switching**
- The device that performs switching is called a **Switch**
- A switch network consists of a number of switches connected one another, which provide connection to external node (A-F)

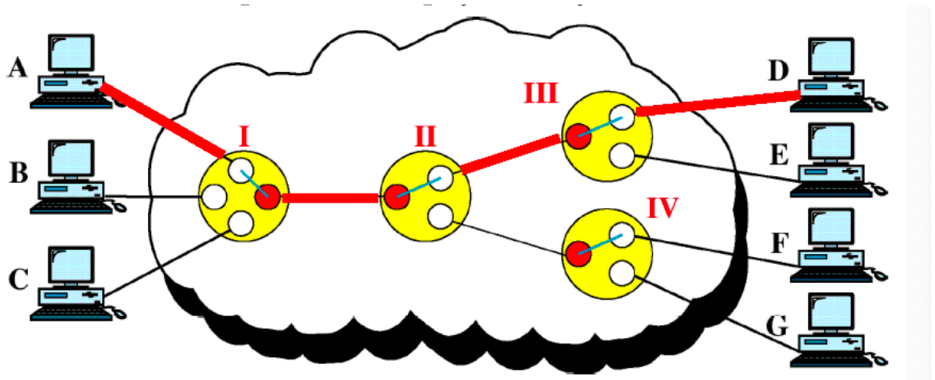





- Switch are **hardware and/or software** devices that can provide *temporary linkage* between two or more connected nodes for the duration of their required connection

#### Type of Switching

- *Circuit-switching*
- *Packet-switching*
- Circuit-switching create a **direct physical** connection between two nodes (at the physical layer)




## Circuit Switching

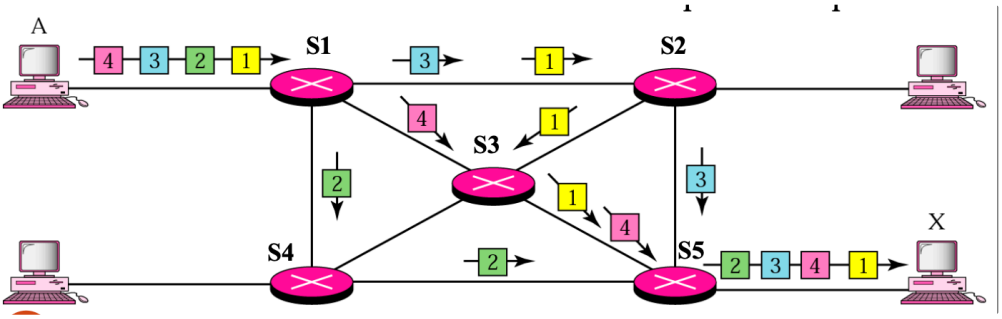
- **no other nodes share the bandwidth** during the whole session
- *substantial wastage* if the connected nodes do not communicate all the time
  - inefficient for Internet browsing, but good for *voice* or other constant bit rate data transfers
- Example of circuit switching includes the Public Switched Telephone Network (**PSTN**)

## Packet Switching

- Internet and many other computer related types of traffic are **bursty** in nature, *highly irregular in time*
- **packet switching**
  - *broken down into separate units with variable lengths*
- **control information** to header or trailer
- Two main approach to **implement packet switching**
  - **Datagram approach**
  - **Virtual circuit approach**

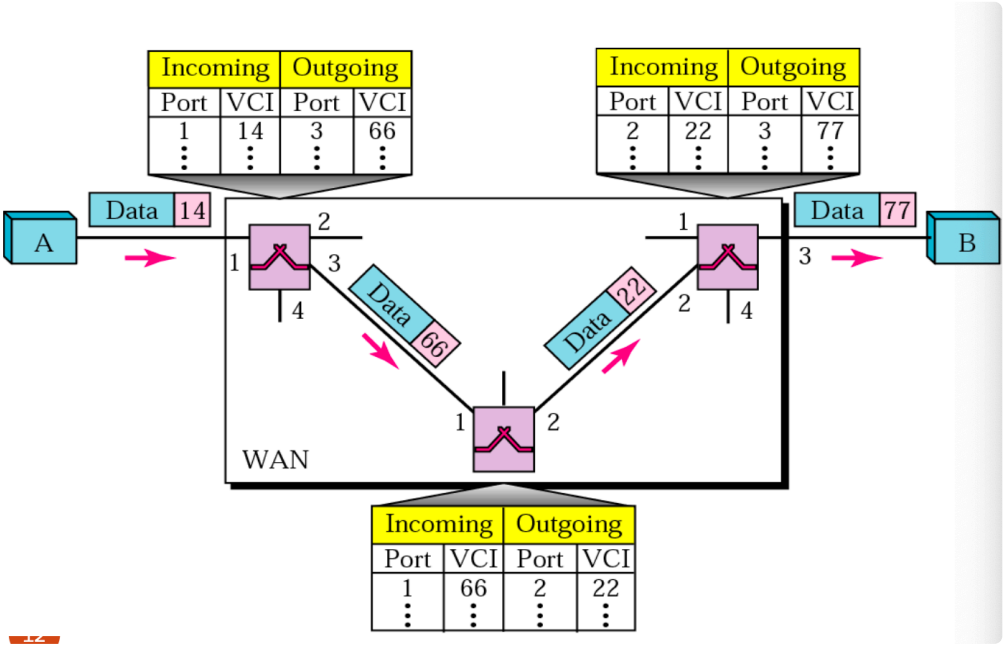
## Datagram Approach

- Datagram switching is *normally* done **at the network layer**
- each packet is treated **independently**
- a message is broken into several packets, each of them will be treated as *separate unit* across the network **go by different paths**
- **Packets under this approach is called a datagram**
- Packets may arrive *out of order*, the *transport layer* need to *re-ordering*
- The switches in datagram network are traditionally referred to as **routers**, which use a **routing table** based on the destination address to establish a route for a particular packet




## Virtual Circuit Approach

- **VC approach**
- all packets of the *same message* is transferred **via a preplanned route (the same route)**
  - Packets **arrive in order**
  - The route is *established* between sender and receiver **at the beginning of the communication**
  - **Virtual Circuit Identifier (VCI)** is used to identify the route




### Types of Virtual Circuit

- Two main methods to setting up
  - *Permanent virtual circuit (PVC)*
  - *Switch virtual circuit (SVC)*
- **Permanent virtual circuit**
  - Set up by the *network provider* and is *in place all the time*
  - *NO need to set up* the VC *before or terminate* the VC after transmission
- **Switched virtual circuit**

- *Set up every time* when a VC is needed, and *terminated after the transmission*
- The same nodes may get the *same or a different VC* every time according to network conditions
- *More flexible* but *requires set up time* before data transfer begins

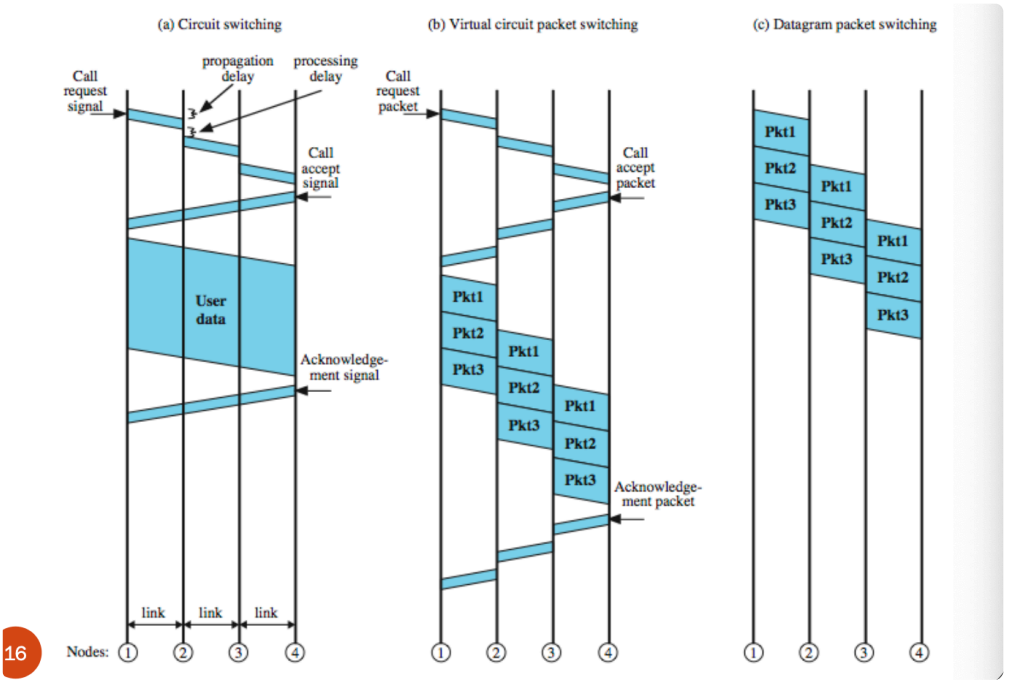
## Virtual Circuits vs. Datagram

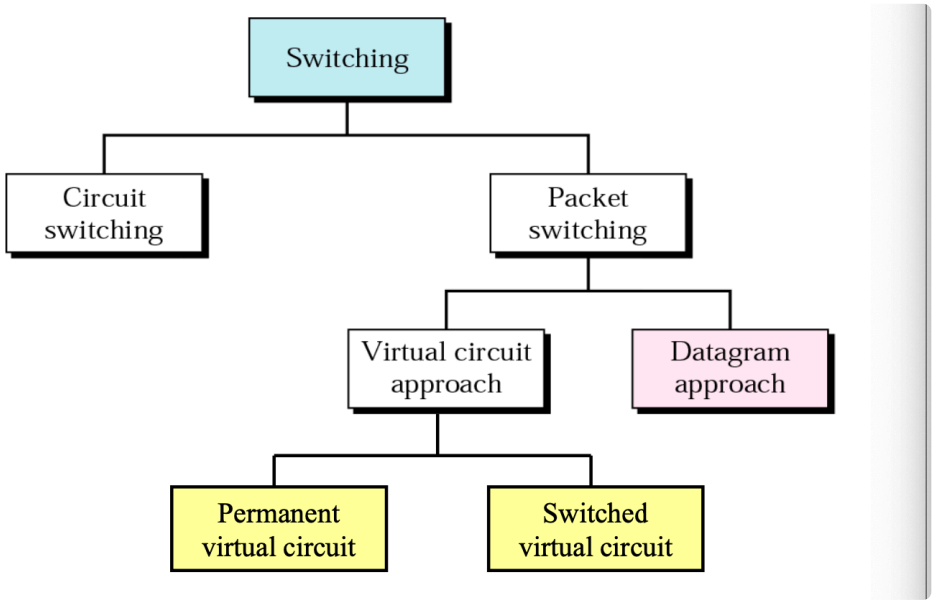
| | Comparison | Virtual Circuit | Datagram |
|---|---|---|---|
| 1 | Network Functions | Provides *sequencing and error control* | Does not provide sequencing or error control |
| 2 | Forwarding Speed | *Faster* (no routing decision needed at each node) | Slower (each packet requires independent routing) |
| 3 | Routing Decision | Made once during connection setup | Made independently for each packet |
| 4 | Reliability | *Less reliable* (failure of a node results in loss of all data) | *More reliable* (can bypass failed nodes) |
| 5 | Flexibility | Less flexible (requires pre-established connection) | More flexible (*adapts to congestion and failures*) |

## Circuit vs. Packet Switching

| | Comparison | Circuit Switching | Packet Switching |
|---|---|---|---|
| 1 | Transmission Delay | Low (once the connection is established, data flows continuously) | High (each packet may take different paths) |
| 2 | Propagation Delay | *Negligible* | *Negligible* |
| 3 | Transmission Time | Continuous once the connection is set up | Independent transmission of each packet |
| 4 | Node Delay | Low (only occurs during connection setup) | High (each packet must be *processed at nodes*) |
| 5 | Transparency | *High (provides a constant data rate)* | Low (data must be converted and processed in packets) |
| 6 | Overhead | *No additional overhead* (data flows directly) | Extra overhead (includes *destination address and control information*) |

## Summary of Switching Technologies

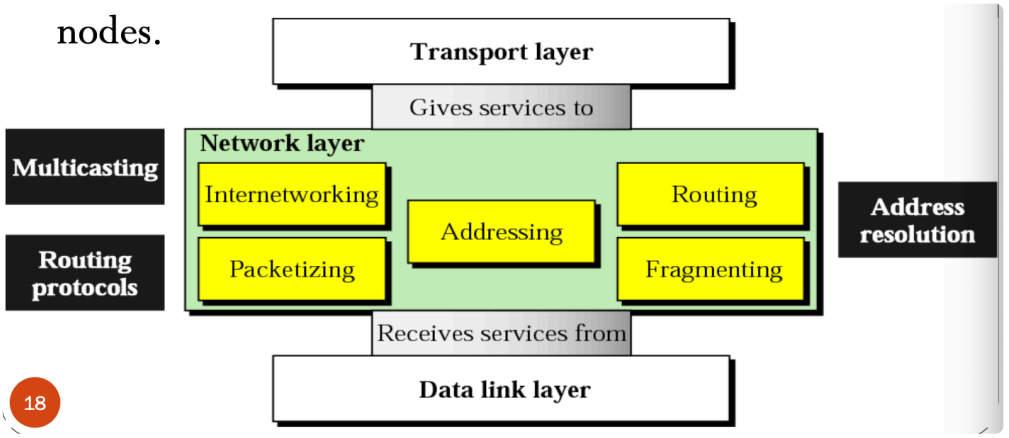



# Network Layer

## Network Layer

- *getting packets from source all the way to destinations*
- **lowest layer deal with end to end**




- The network layer is responsible for **host to host** data (packet) delivery
  - **Connect heterogeneous networks** to look like a single network (internetworking)
  - **Uniquely identify each device** to allow global communication (*addressing*)
  - **Make decision** to deliver data to the destination (*routing*)
  - **Fit data** to the size used by the lower layer protocol (*fragmentation*)

## Network-layer Services

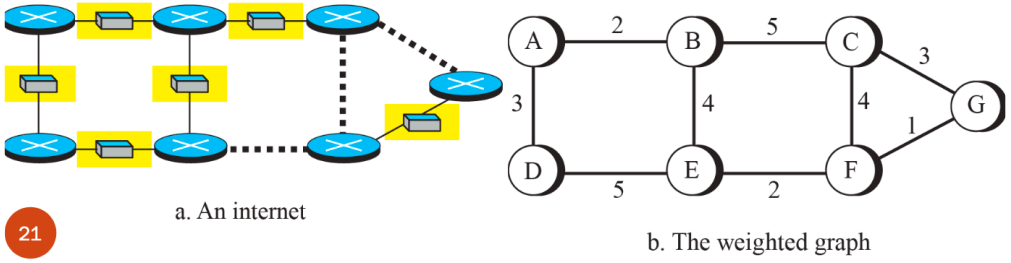
### Packetizing

- *At the source host*
  - Receives the **payload** (data form the *transport layer*), add a **header** that contains the **source and destination addresses** and some other information
  - **fragmentation**
- *At the destination*
  - *decapsulates* the packet and delivers the payload to the transport
  - *reassembled fragment* until **all fragments arrived**

### Routing

- connects LANs and WANs with routers
- **find the best one** among these routers
- **routing protocols** to derive its own *routing table* (*forwarding table*)

◆Example: From A to G, use the route A-B-E-F-G.



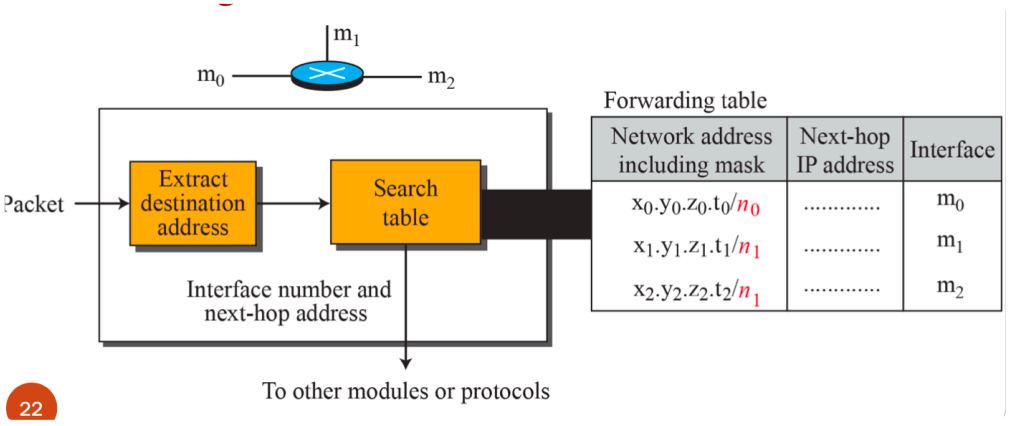

a. An internet

b. The weighted graph

21

## Forwarding

- When a router receivers a packet from one of its attached network, it is *forwarded to another* attached network.
- **Forwarding table**




22

## Error Control

- **only a checksum** in a *datagram*
- The checksum only **check the header** of a datagram

## Flow Control

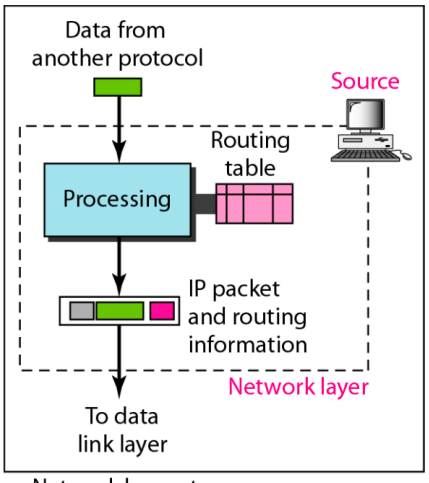
- **does not** provide any flow control
- provide by *the transport layer*

## Forwarding, Routing, and Switching

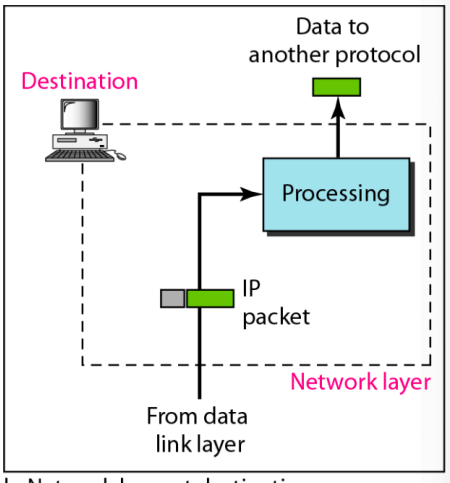
| | Concept | Definition | Key Functions | Examples |
|---|---|---|---|---|
| 1 | Routing | The process of *exchanging topological information* to *build forwarding tables.* | Uses *routing protocols* to determine *the best path* for data packets. | OSPF, BGP, RIP |

| | Concept | Definition | Key Functions | Examples |
|---|---|---|---|---|
| 2 | Forwarding | Decides the *next-hop address or output port* for a packet. | *Uses forwarding tables* created by routing to send packets to their next destination. | IP forwarding, MPLS forwarding |
| 3 | Switching | Moves a packet from an input port to an output port within a network device. | *Handles internal data transfer* within a switch or router. | Ethernet switching, Packet switching |

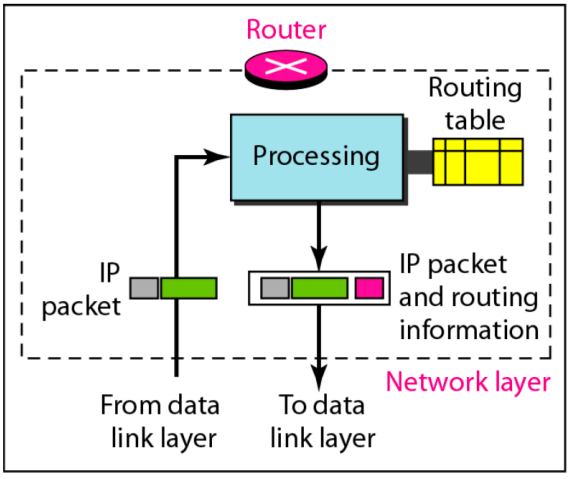
## Network Layer at the Source, Router, and Destination



a. Network layer at source



b. Network layer at destination



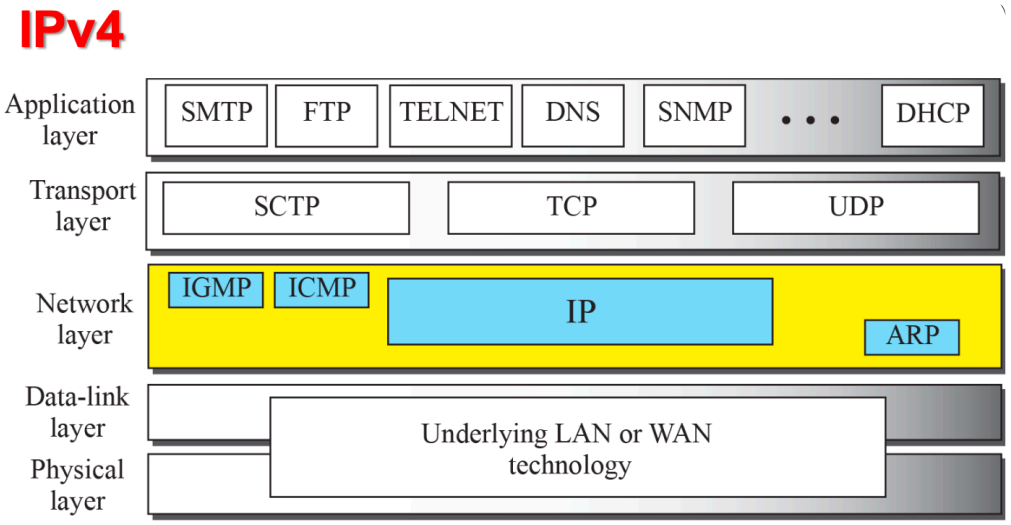
c. Network layer at a router

## Internet Protocol

- **IP** is the network layer protocol on the Internet
- IP provides a *connectionless datagram service*
- **best-effort** delivery service
  - *NOT* guarantee a datagram will arrive in *correct order*

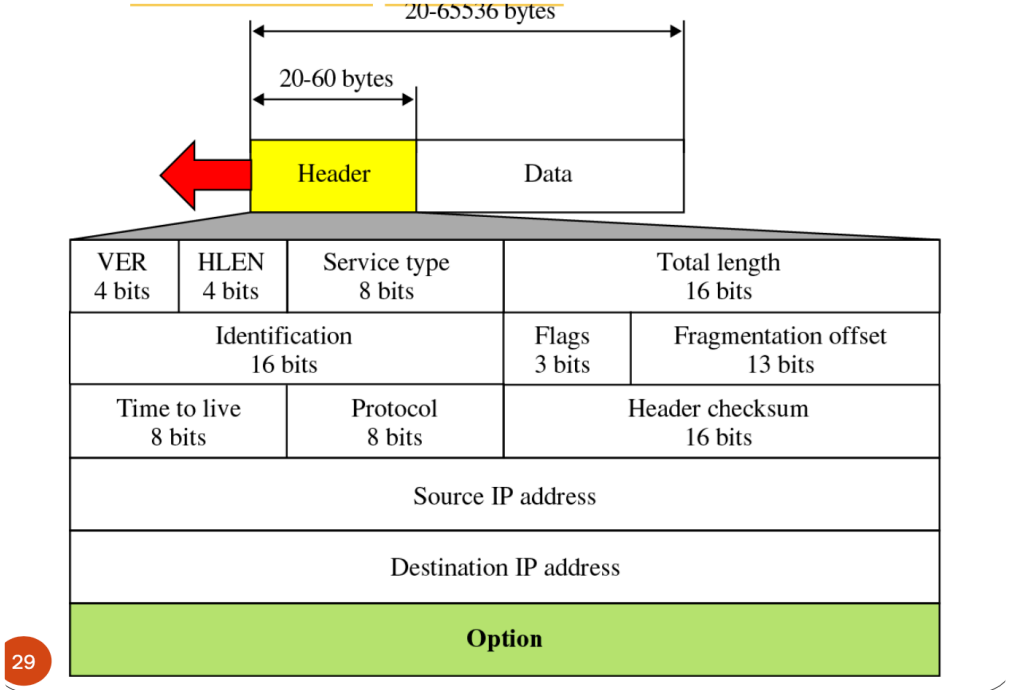
- *NOT* guarantee a datagram will arrive in *certain time*
- *NOT* guarantee the datagram will *ever arrive*

## IPv4




Internet Group Management Protocol (IGMP)
Internet Control Message Protocol (ICMP)
Address Resolution Protocol (ARP)

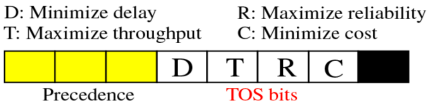
### Datagram Format






## Header Fields

| | Field | Description |
|---|---|---|
| 1 | Version | Indicates the IP version (e.g., 4 for IPv4). |

| | Field | Description |
|---|---|---|
| 2 | IP Header Length (HLEN) | Specifies the length of the header *in 32-bit words*. Example: HLEN = 6 → 24 bytes. |
| 3 | Total Length | *Length of the entire datagram in bytes* (including header and data). |
| 4 | **Type of Service (TOS)** | Specifies *how the datagram should be handled*<br><br>- *Precedence*: 3bits (priority of the datagram *Not used in version 4*)<br><br>- *Service type*: 4bits (TOS bits) **only one bit set at a time**<br><br>- *Remaining bit*: not used<br><br>  |
| 5 | Identification | A *unique sequence* number used for *reassembling* fragmented datagrams. |
| 6 | **Flags** | Controls fragmentation (e.g., "Don't Fragment" and "More Fragments" flags). *whether this is the last fragment*<br><br>  |
| 7 | **Fragmentation Offset** | Specifies the position of a fragment within the original datagram (**in units of 8 bytes**). |
| 8 | Time to Live (TTL) | *Prevents infinite loops* by decrementing at each router; if TTL = 0, the packet is dropped. |
| 9 | **Protocol** | Identifies the *upper-layer protocol* (e.g., TCP = 6, UDP = 17). |
| 10 | Header Checksum | A checksum for error detection; *recalculated at each router* due to TTL changes. |
| 11 | Source IP Address | The sender's IP address. |
| 12 | Destination IP Address | The recipient's IP address. |
| 13 | Options | Allows for optional settings (rarely used in IPv4, removed in IPv6). |
| 14 | **Padding** | Fills the header to a *multiple of 32 bits.* |

## Types of service

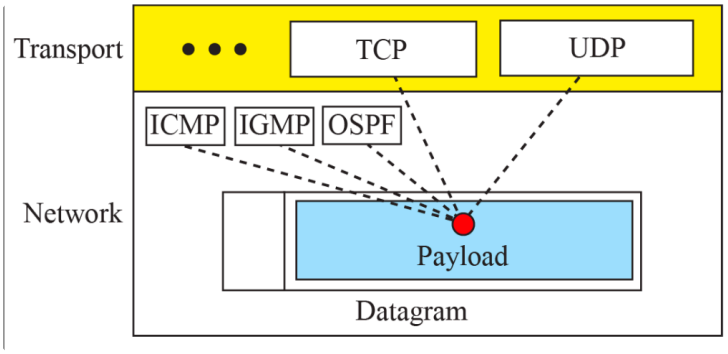
### Types of service

| *TOS Bits* | *Description* |
|---|---|
| 0000 | Normal (default) |
| 0001 | Minimize cost |
| 0010 | Maximize reliability |
| 0100 | Maximize throughput |
| 1000 | Minimize delay |

| *Protocol* | *TOS Bits* | *Description* |
|---|---|---|
| ICMP | 0000 | Normal |
| BOOTP | 0000 | Normal |
| NNTP | 0001 | Minimize cost |
| IGP | 0010 | Maximize reliability |
| SNMP | 0010 | Maximize reliability |
| TELNET | 1000 | Minimize delay |
| FTP (data) | 0100 | Maximize throughput |
| FTP (control) | 1000 | Minimize delay |
| TFTP | 1000 | Minimize delay |
| SMTP (command) | 1000 | Minimize delay |
| SMTP (data) | 0100 | Maximize throughput |
| DNS (UDP query) | 1000 | Minimize delay |
| DNS (TCP query) | 0000 | Normal |
| DNS (zone) | 0100 | Maximize throughput |

33

## Multiplexing and Demultiplexing using the value of the Protocol Field

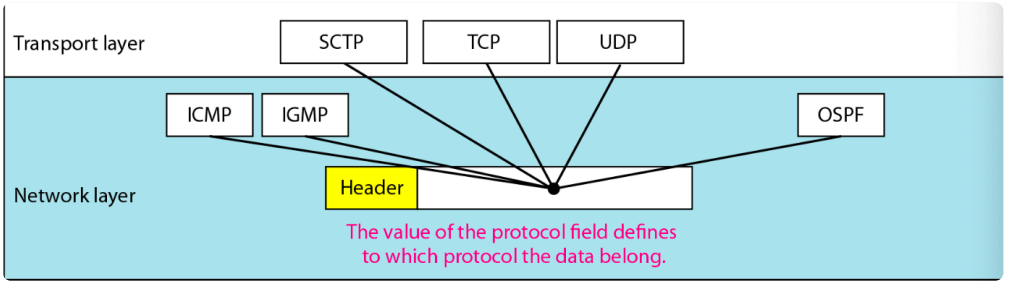



| Protocol | Decimal | Hex |
|---|---|---|
| TCP | 06 | 0×06 |
| UDP | 17 | 0×11 |
| SCTP | 132 | 0×84 |
| ICMP | 01 | 0×01 |
| IGMP | 02 | 0×02 |
| L2TP | 115 | 0×73 |
| OSPF | 89 | 0×59 |
| ESP | 50 | 0×32 |
| AH | 51 | 0×33 |

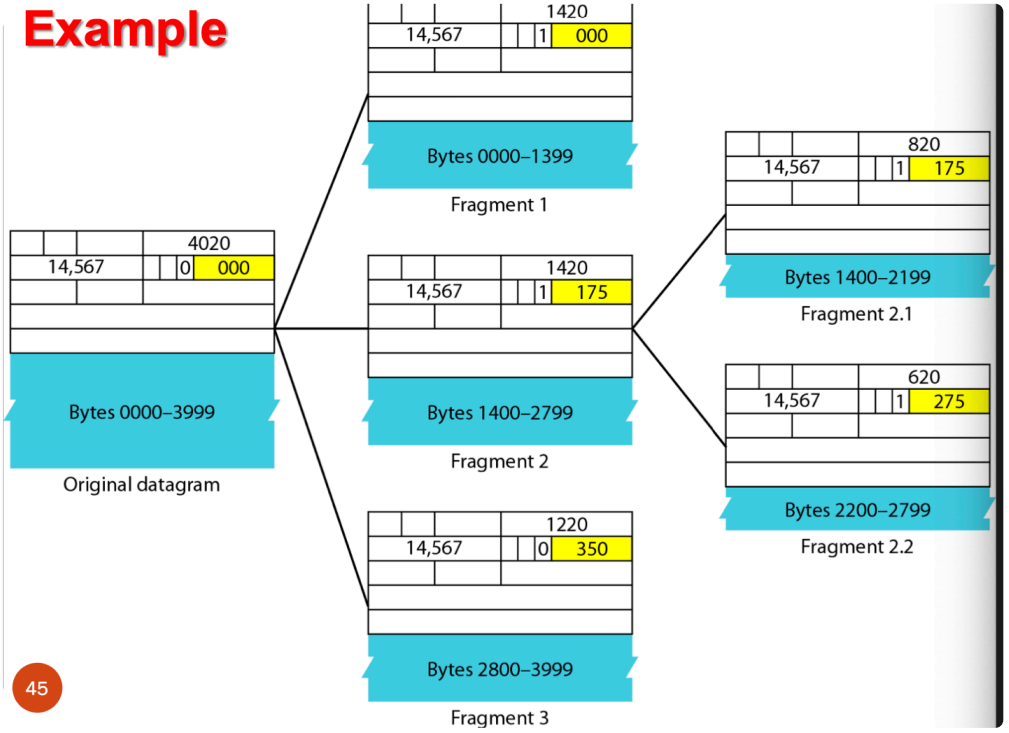
## Data Field

- Data (**payload**)
  - the transport layer segment (TCP or UDP) (application data)

- *other types of data*, such as ICMP message

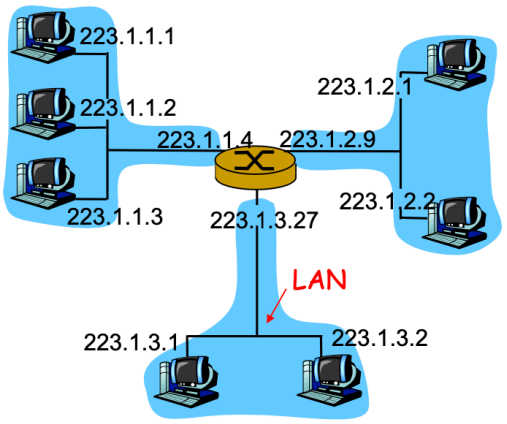



- The format and size of the received frame depend on the *protocol used by the physical network*
- **MTU (Maximum Transfer Unit)**
  - the total size of the datagram must be less than this maximum size




## IP Addressing

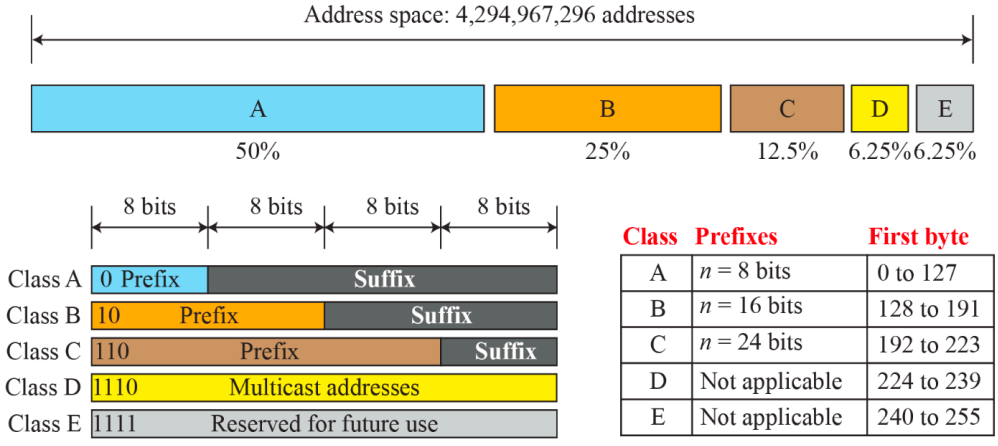
- *What's a network ?* (from IP address perspective)
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router




Network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address, for classful addressing)



## Classful Addressing




| Class | Prefixes | First byte |
|---|---|---|
| A | $n = 8$ bits | 0 to 127 |
| B | $n = 16$ bits | 128 to 191 |
| C | $n = 24$ bits | 192 to 223 |
| D | Not applicable | 224 to 239 |
| E | Not applicable | 240 to 255 |

***a.** The first bit is 0. This is a class A address.*

***b.** The first 2 bits are 1; the third bit is 0. This is a class C address.*

***c.** The first byte is 14; the class is A.*

***d.** The first byte is 252; the class is E.*

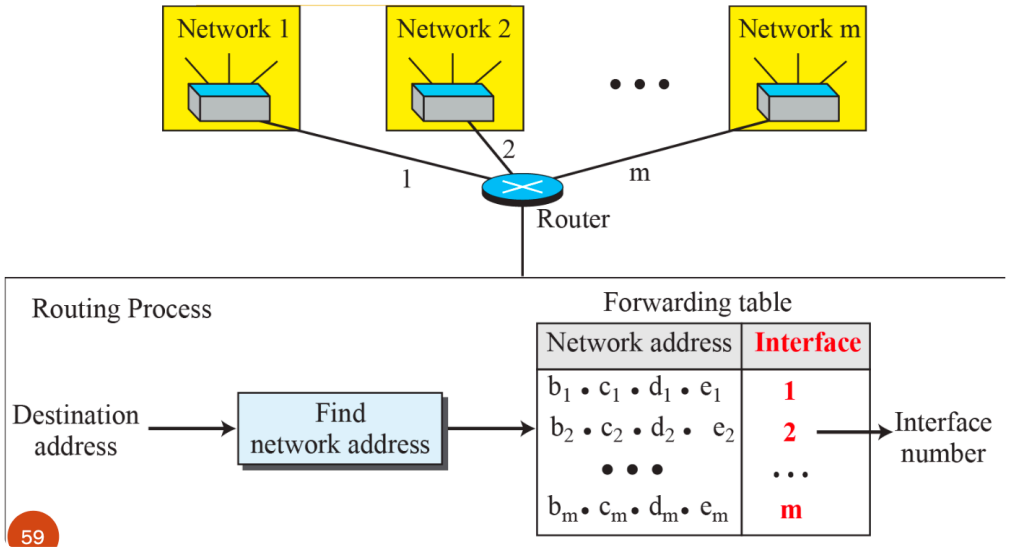
## Classless Address

- A classless address is 167.199.170.82/27
- The number of addresses in the network = $2^{32-27} = 32$
  - In general, $N = 2^{32-n}$

| | | | | |
|---|---|---|---|---|
| Address: 167.199.170.82/27 | 10100111 | 11000111 | 10101010 | 01010010 |
| First address: 167.199.170.64/27 | 10100111 | 11000111 | 10101010 | 01000000 |

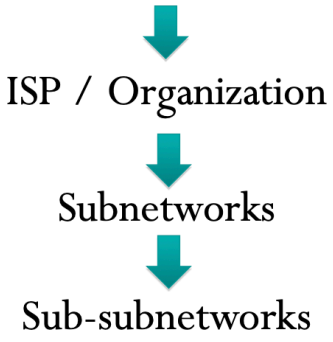
| | | | | |
|---|---|---|---|---|
| Address: 167.199.170.82/27 | 10100111 | 11000111 | 10101010 | 01010010 |
| Last address: 167.199.170.95/27 | 10100111 | 11000111 | 10101010 | 01011111 |

- **Address Mask**
  - A 32-bit number in which all prefix bits are set to 1s and the suffix bits are set to 0s.
  - 11111111.11111111.11111111.11100000 or 255.255.255.224
- The *first address* is used as the **network address**
  - Can be obtained by taking the *AND of an address with the address mask*
- Network address is **used for routing at routers**
- The **1st step** of routing is resolving the *network address* from the *destination address*



## Allocation of Address

- Internet Corporation for Assigned Names and Numbers (ICANN)




Static IP or Dynamic IP via DHCP (Dynamic Host Configuration Protocol) server

- ICANN **Basic rules** :
  - Number of allocated addresses $N = 2^{32-n}$, where n is the prefix length
  - The first address needs to be the prefix follow by (32-n) *number of 0s*

- Example:
- Suppose that an ISP requests a block of 1000 addresses from ICANN.

- Solution
  - Since 1000 is not a power of 2, 1024 addresses are granted.
  - The prefix length is calculated as $n = 32 - \log_2(1024) = 22$. An available block, 18.14.12.0/22, is granted to the ISP.
  - The first address is 18.14.12.0.

## Subnetwork

- An organization (or an ISP) can further divide its network into subnetworks (subnets) and then assign addresses
- Basic Rules
  1. Number of allocated addresses in each subnetwork is a power of 2, i.e. $N_{sub} = 2^{32-ns}$, where *ns* is the prefix length of the subnetwork
  2. The first address needs to be the prefix followed by (32-*ns*) number of 0s. This can be achieved if we first assign addresses to large subnetworks.

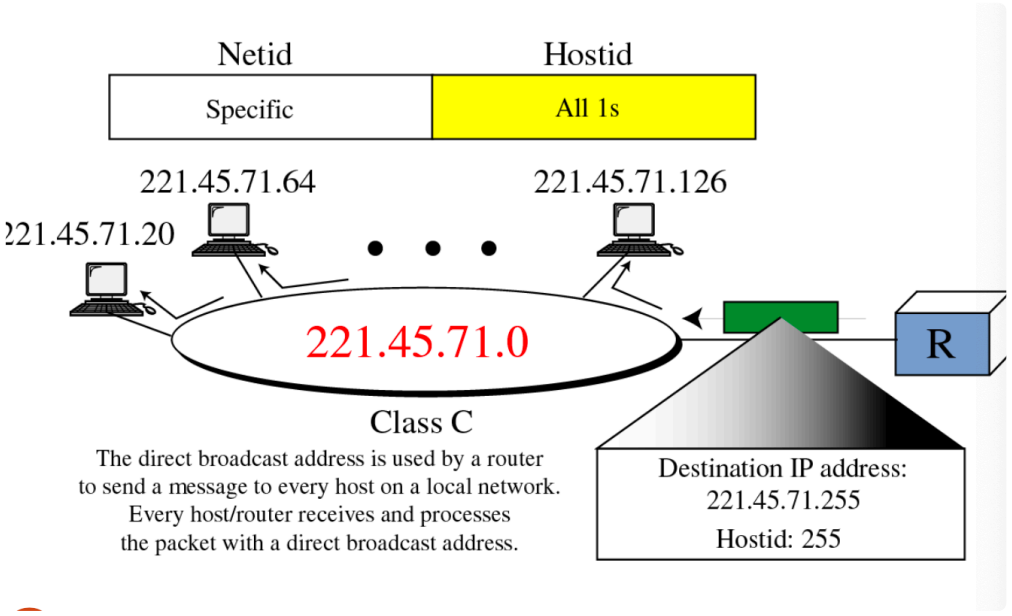
## Special Addresses

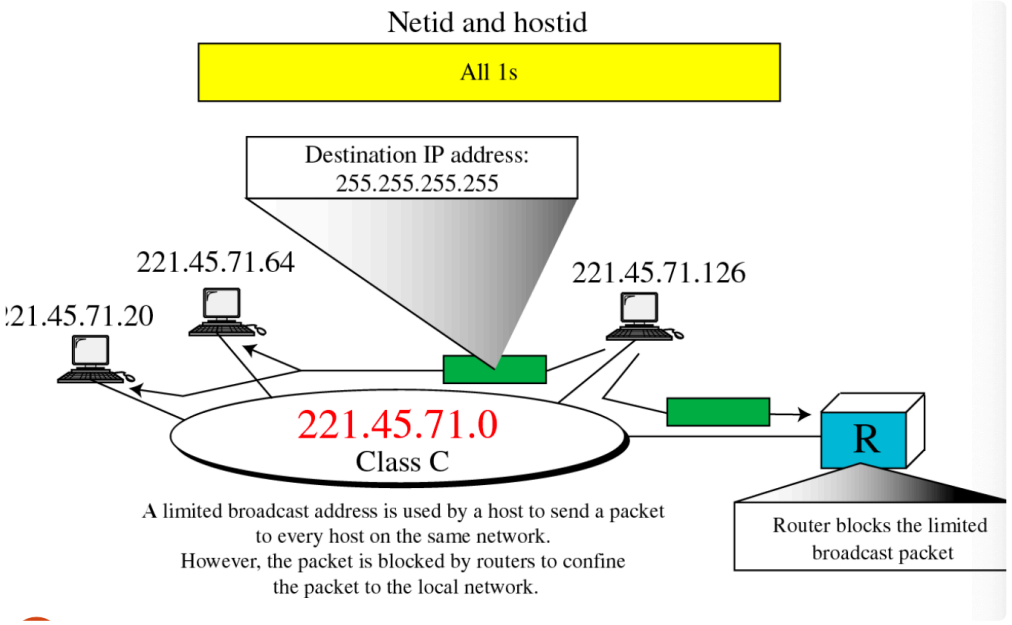
| | Address Type | Address Range | Purpose |
|---|---|---|---|
| 1 | This-host Address | `0.0.0.0/32` | Used when a host *does not yet know its own IP address* (e.g., DHCP request). |
| 2 | Limited-broadcast | `255.255.255.255/32` | Sends a datagram to *all devices in a local network* but does not travel outside the network.<br><br>When a host requests IP address from a DHCP server, it sends out a datagram with source address = `0.0.0.0` and destination address `255.255.255.255` |
| 3 | Loopback Address | `127.0.0.0/8` | Used for *local program testing*(e.g., `127.0.0.1` for localhost). |
| 4 | Private Addresses | `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`, `169.254.0.0/16` | Used within private networks; *requires NAT (Network Address Translation)* to communicate with public networks.<br><br>the num of available IP addresses much less than the num of host |
| 5 | Multicast Addresses | `224.0.0.0/4` | Used for group communication (one-to-many transmission). |

# IP Address

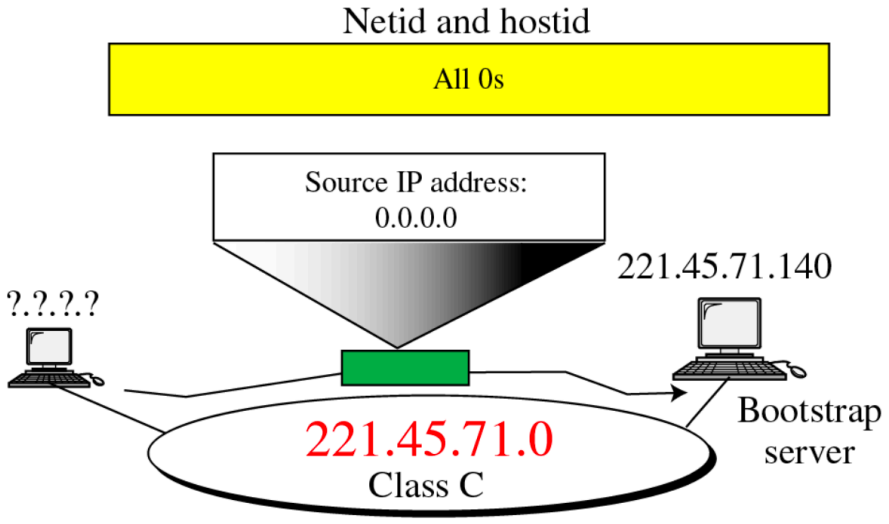
| | 类型 | 地址范围 | 作用 |
|---|---|---|---|
| 1 | Direct Broadcast | 例如 `192.168.1.255`（网络号+全1主机号） | 发送给**特定网络内的所有主机**，可以跨路由传播（如果路由器允许）。 |
| 2 | Limited Broadcast | `255.255.255.255` | 发送给**当前子网内的所有主机**，不会被路由器转发。用于本地网络发现，如 DHCP 发现服务器。 |
| 3 | Multicast | `224.0.0.0/4` | 发送给一组特定的主机，而不是整个网络。适用于视频流、在线会议等场景。 |

1. 是否跨网络传播：
   - **Direct Broadcast**：可以跨网络传播，但可能被路由器拦截。
   - **Limited Broadcast**：仅限本地子网，不会被路由器转发。
   - **Multicast**：根据组成员关系进行转发，不会像广播那样发送给所有主机。
2. 使用场景：
   - **Direct Broadcast**：网络管理员远程唤醒设备（`Wake-on-LAN`），或向整个子网发送公告信息。
   - **Limited Broadcast**：DHCP 服务器发现（`DHCP DISCOVER`），ARP 请求等。
   - **Multicast**：IPTV、在线视频会议、VR 直播等需要组播通信的应用。

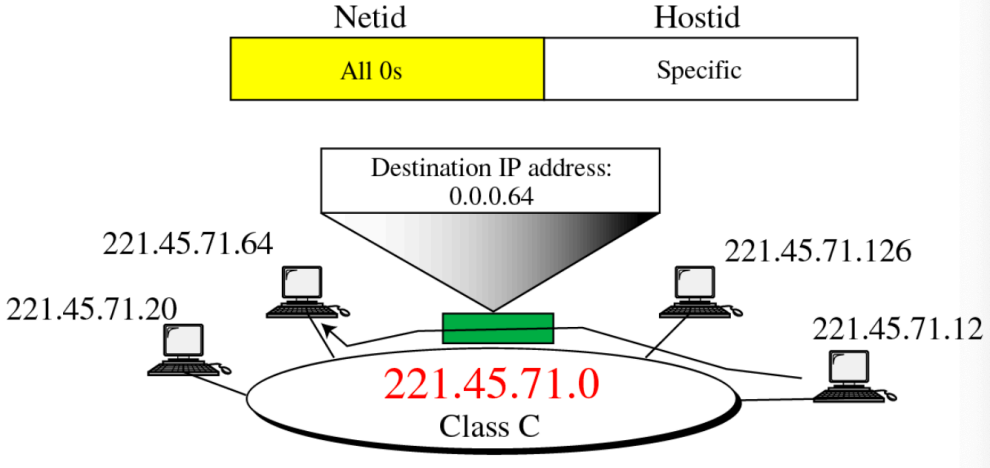




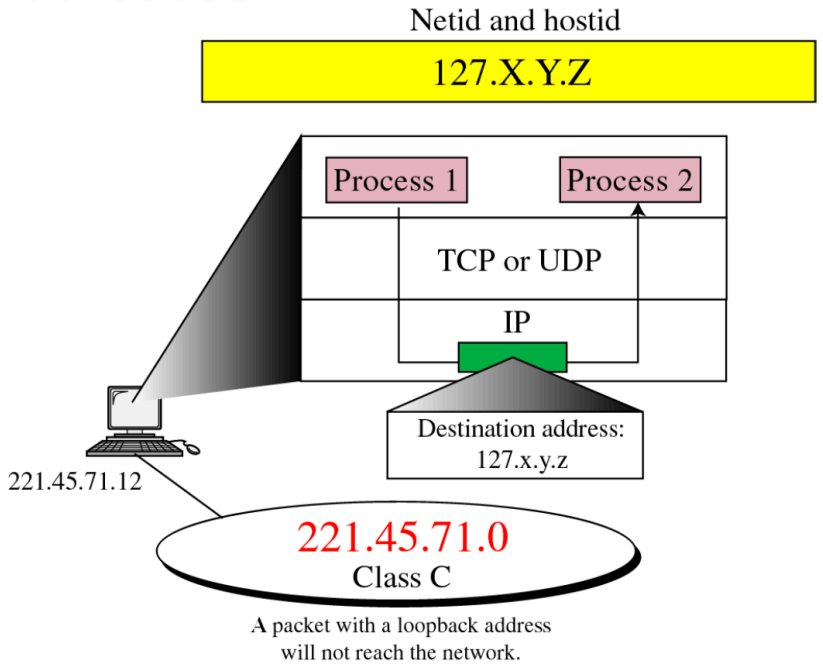

A limited broadcast address is used by a host to send a packet to every host on the same network. However, the packet is blocked by routers to confine the packet to the local network.

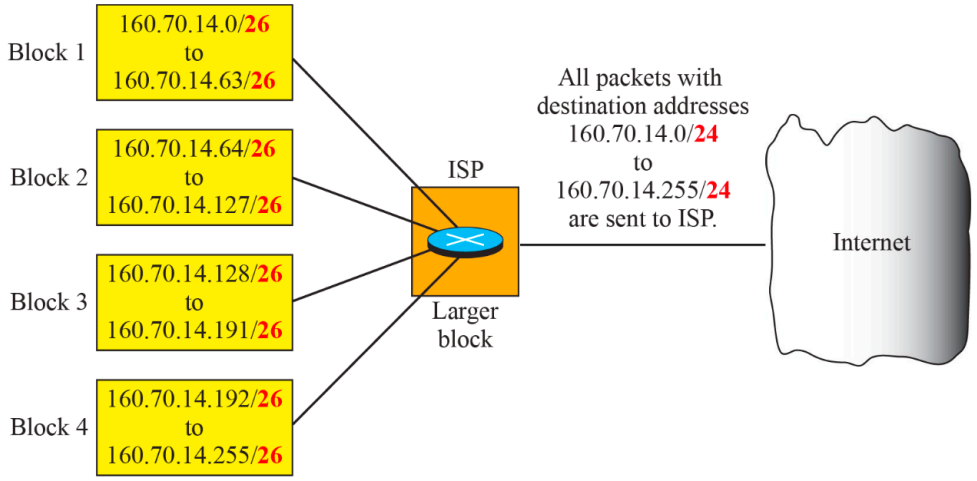



A host which does not know its IP address uses the IP address 0.0.0.0 as the source address and 255.255.255.255 as the destination address to send a message to a bootstrap server.

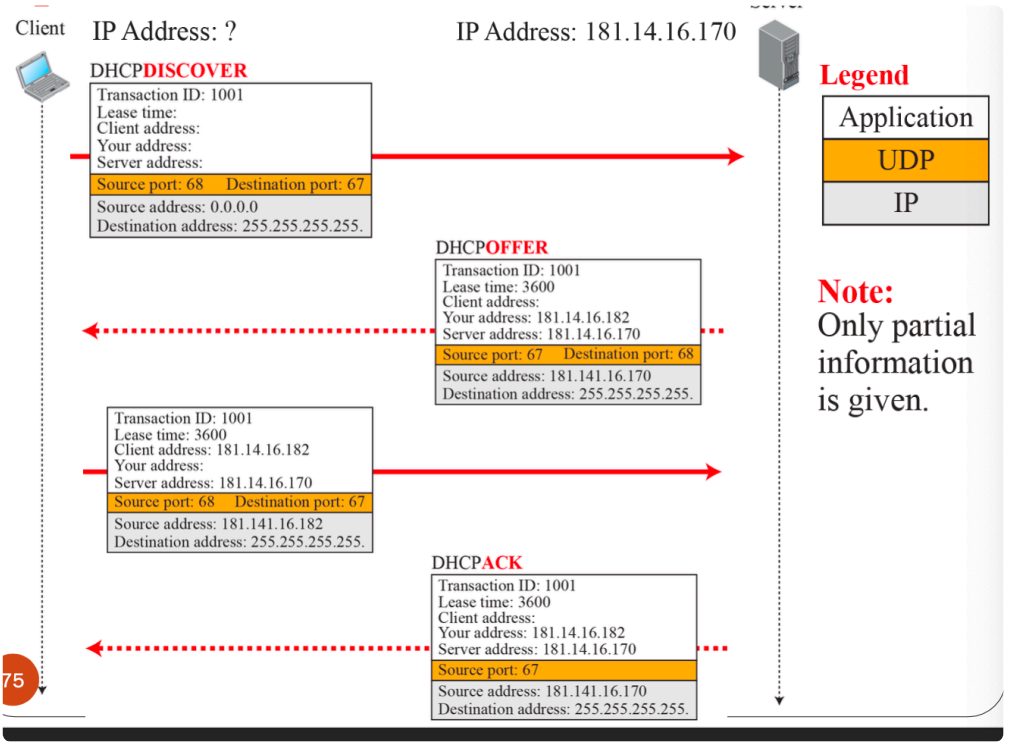



This address is used by a router or host to send a message to a specific host on the same network.



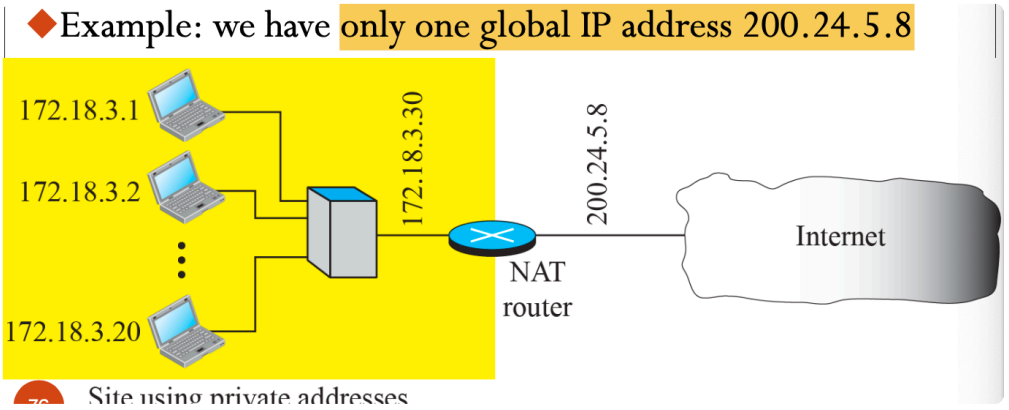




## Dynamic Host configuration Protocol (DHCP)

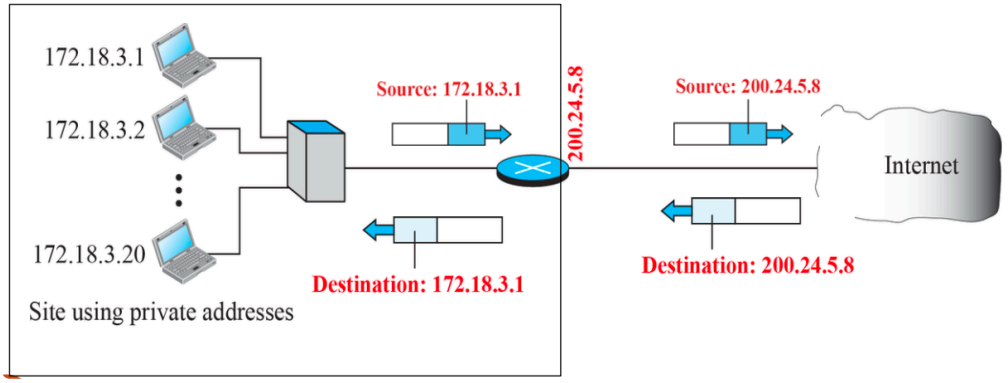



## NAT (Network Address Translation)

- In an organization, we *cannot assign* address to a new host if the num of host *exceeds its available addresses*
  - **Create a private network** and connect to the Internet via a **NAT router**

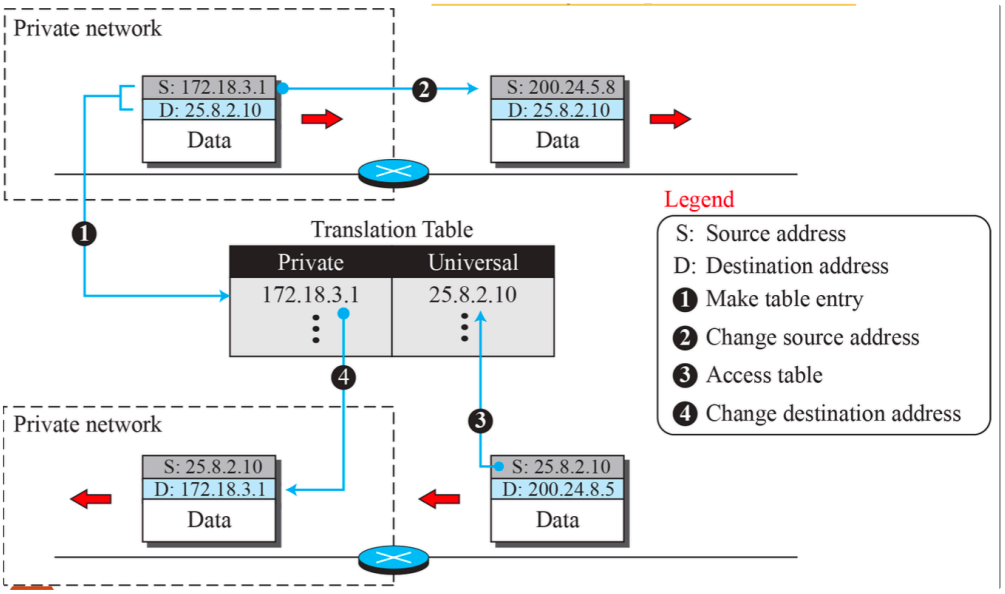



- NAT router **replace the source address** of all the outgoing packets by the *global NAT address*




We need a **translation table**

- Case I **One global address**
  - The translation table record down the *destination address of all outgoing packets*
  - When the response cone back from the destination, the router can find the private address of the packet from the source address of the packet
  - must be **initiated by the private network**




- Case II **A Pool of IP Address**
  - The number of hosts which can communicate with the *same external host at the same time* is the number of *Global IP addresses assigned to the network*
  - **Limitation**
    - Each host can only access one external server program at the same time.
    - *Solution*: Record both IP Addresses and Port Addresses

| Public IP | Port | Private IP | Port |
|---|---|---|---|
| 200.24.5.8 | 7777 | 192.168.1.10 | 700 |
| 200.24.5.8 | 8888 | 192.168.1.8 | 555 |
| … | | … | |



# Network Address Port Translation (NAPT)