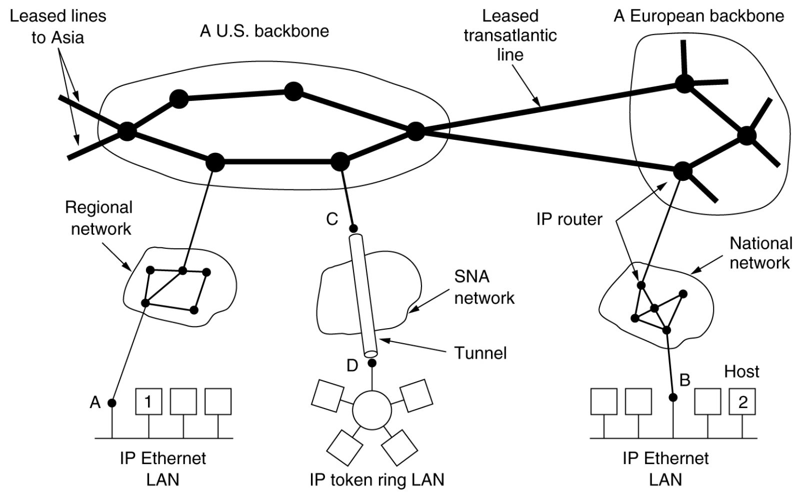
# The Network Layer

## The Network Layer in the Internet

## The Network Layer in the Internet

- The IP Protocol
- IP Addresses
- Internet Control Protocols

# Collection of Subnetworks




The Internet is an interconnected collection of many networks.

# The IP Protocol

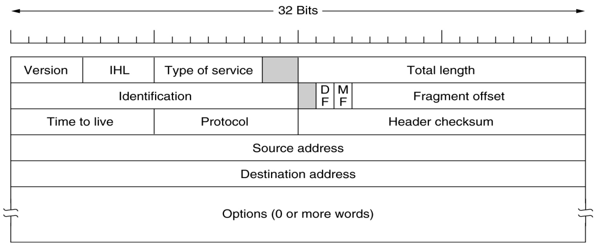
32 Bits

| | | | | | |
|---|---|---|---|---|---|
| Version | IHL | Type of service | | Total length | |
| Identification | | | | DF / MF | Fragment offset |
| Time to live | | Protocol | | Header checksum | |
| Source address | | | | | |
| Destination address | | | | | |
| Options (0 or more words) | | | | | |

The IPv4 (Internet Protocol) header.

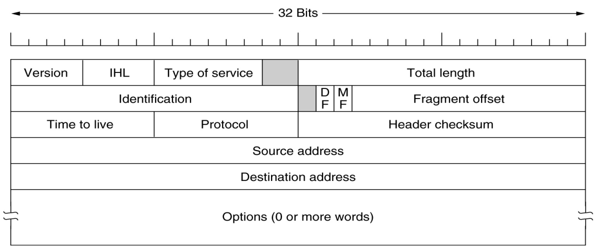
# The IP Protocol

- *Version* : keeps track of the version of the protocol used by the datagram (now IPv4 or IPv6).
- *IHL*: the header size is not constant; IHL tells the header size in 32 bits words. Minimum 5 (no options) ,Maximum 15 (60 bytes header, 40 bytes of options).
- *Type of Service*: Now used by differentiated services to tell the service class.
- *Total length*: maximum size of the datagram up to 65535 bytes.
- *Identification*: Used to identify fragments of the same datagram.
- *DF*: Don’t Fragment bit




# The IP Protocol

- *MF*: More Fragments; all fragments except the last have this bit set.
- *Fragment Offset*: tells where in the current datagram this fragment belongs (8192 fragments are possible with the 13 bits field).
- *Time to Live*: It is supposed to count seconds but in practice it counts hops (255)
- *Protocol*: TCP , UDP and many others (http://www.iana.org)
- *Header Checksum*: Verifies the header only.
- *Source and Destination addresses*: Indicate IP (Network+ host) numbers
- *Options*: Multiples of four bytes

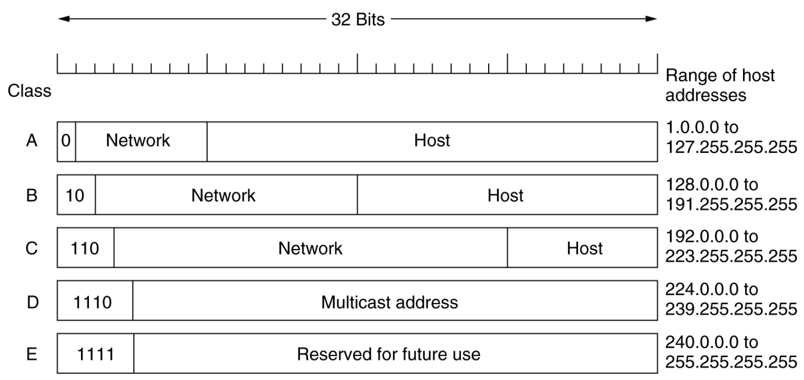


# The IP Protocol

| Option | Description |
|---|---|
| Security | Specifies how secret the datagram is |
| Strict source routing | Gives the complete path to be followed |
| Loose source routing | Gives a list of routers not to be missed |
| Record route | Makes each router append its IP address |
| Timestamp | Makes each router append its address and timestamp |

Some of the IP options.

# IP Addresses



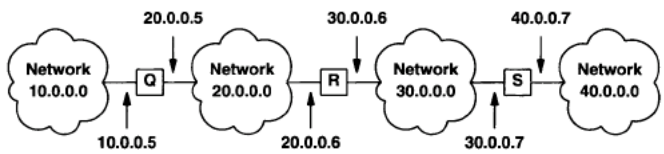

IP address formats.

# IP Addresses (2)

| | |
|---|---|
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | This host |
| 0 0 . . . 0 0 \| Host | A host on this network |
| 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | Broadcast on the local network |
| Network \| 1 1 1 1 . . . 1 1 1 1 | Broadcast on a distant network |
| 127 \| (Anything) | Loopback |

Special IP addresses.

# Routing




(a)

| TO REACH HOSTS ON NETWORK | ROUTE TO THIS ADDRESS |
|---|---|
| 20.0.0.0 | DELIVER DIRECTLY |
| 30.0.0.0 | DELIVER DIRECTLY |
| 10.0.0.0 | 20.0.0.5 |
| 40.0.0.0 | 30.0.0.7 |

(b)

**Figure 8.2** (a) An example internet with 4 networks and 3 routers, and (b) the routing table in *R*.

# Routing Algorithm

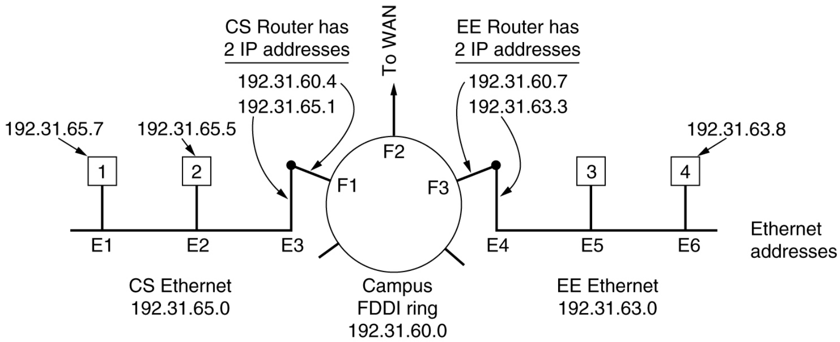
**Algorithm:**
RouteDatagram (Datagram, RoutingTable)

Extract destination IP address, D, from the datagram
    and compute the network prefix, N;
if N matches any directly connected network address
    deliver datagram to destination D over that network
    (This involves resolving D to a physical address,
    encapsulating the datagram, and sending the frame.)
else if the table contains a host-specific route for D
    send datagram to next-hop specified in table
else if the table contains a route for network N
    send datagram to next-hop specified in table
else if the table contains a default route
    send datagram to the default router specified in table
else declare a routing error;

# ARP– The Address Resolution Protocol

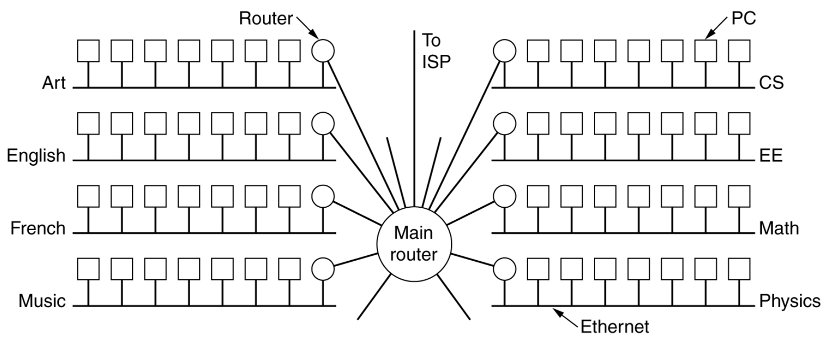
- Most part of the computers in the Internet are attached to LAN's by an interface board that only understands data link layer addresses (MAC addresses)
- How to map IP addresses into MAC addresses? ARP is the answer.
  - When a router wants to send a packet to an IP address on it's own network it must find out the corresponding MAC address.
  - Instead of managing a table (error prone and time consuming) the router broadcast a message in the network asking: "Who owns IP ###.###.### ?"
  - The corresponding host on the LAN will answer back with its MAC address.
  - Caches are used to improve the eficiency of the algorithm.
  - Almost all machines in the Interner run ARP (RFC 826)

# ARP– The Address Resolution Protocol




Three interconnected /24 networks: two Ethernets and an FDDI ring.

# Subnets

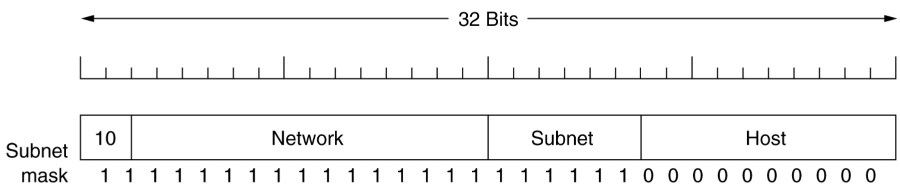



A campus network consisting of LANs for various departments.

# Subnets

- Imagine a growing University with an assigned class B address.
- When the network grows (Ethernet for example) new routers have to be introduced to separate the traffic.
- New addresses are not an option because IP addresses are scarce and we already have more than 60000 addresses...
- But the addresses belong to a single network, not to a collection of LAN’s.

# Subnets




A class B network subnetted into 64 subnets.
Subnet mask: 255.255.252.0/22

# Routing with subnets

**Algorithm:**
Route-IP-Datagram (datagram, routing-table)

```
Extract destination IP address, ID from datagram;
If prefix of ID matches address of any directly connected
        network send datagram to destination over that network
        (This involves resolving ID to a physical address,
        encapsulating the datagram, and sending the frame.)
else
        for each entry in routing table do
                Let N be the bitwise-and of ID and the subnet mask
                If N equals the network address field of the entry then
                        route the datagram to the specified next hop address
        endforloop
If no matches were found, declare a routing error;
```

# Subnets

- Example: A packet addressed to 130.50.15.6 arriving at the main router is ANDed with the subnet mask 255.255.252.0/22

  10000010.00110010.00001111.00000110

  11111111.11111111.11111100.00000000

  10000010.00110010.00001100.00000000 > 130.50.12.0
- The routing table is searched for the interface to use to get to this subnet.
- *Subnetting reduces router table space by creating a three layer hierarchy: network, subnet, host.*

## CIDR – Classless Inter-Domain Routing

- How to give Internet a little space to breath ?
  - Allocate the remaining IP addresses in variable-size blocks, without regard to classes.
  - Dropping classes makes forwarding more complicated because we can't use only the IP to look for an outgoing line as was done before.
  - The routing tables were extended with a 32 bit mask column. So we have:

    (IP address, subnet mask, Outgoing line) for each entry.
  - In order to keep the routing table as small as possible routers often combine several entries into a single *agregate entry* whenever possible.

## CIDR – Classless Inter-Domain Routing

- Consider an Example where millions of addresses are available starting at:
  - 194.24.0.0 (11000010.00011000.00000000.00000000)
  - Cambridge needs 2048 addresses ($2^{11}$ =2048) so we need 11 bits for the host, thus we have a netmask with 21 bits.

11000010**.**00011000.00000000.00000001 (starting address)

11000010**.**00011000.00000111.11111111 (last address = broadcast)

Decimal Notation:

194.24.0.1 (start address) 194.24.7.255 (last address = broadcast)

255.255.248.0 (netmask)

**Written as: 194.24.0.0/21**

## CIDR – Classless Inter-Domain Routing

- Next, Oxford requests 4096 addresses ($2^{12}$ =4096) so we must alocate addresses in a 12 bit boundary.
- Its not possible to start in 194.24.8.0 because there are not enough consecutive addresses available

11000010.00011000.00001000.00000000 (First available address)
11000010.00011000.00010000.00000000 (first address from Oxford)
11000010.00011000.00011111.11111111 (last address from Oxford)
Decimal Notation:
194.24.16.0 (start address) 194.24.31.255 (last address = broadcast)
255.255.240.0 (netmask)
**Written as: 194.24.16.0/20**

## CIDR – Classless Inter-Domain Routing

- Next, Edinburgh requests 1024 addresses ($2^{10}$=1024) so we must alocate addresses in a 10 bit boundary.
- There is one available starting in 194.24.8.0 just after Cambridge

11000010.00011000.00001000.00000000 (first Edinburgh address)
11000010.00011000.00001011.11111111 (last Edinburgh address)

Decimal Notation:
194.24.8.0 (start address) 194.24.11.255 (last address = broadcast)
255.255.252.0 (netmask)
**Written as: 194.24.8.0/22**

# CIDR – Classless Inter-Domain Routing

| University | First address | Last address | How many | Written as |
|---|---|---|---|---|
| Cambridge | 194.24.0.0 | 194.24.7.255 | 2048 | 194.24.0.0/21 |
| Edinburgh | 194.24.8.0 | 194.24.11.255 | 1024 | 194.24.8.0/22 |
| (Available) | 194.24.12.0 | 194.24.15.255 | 1024 | 194.24.12/22 |
| Oxford | 194.24.16.0 | 194.24.31.255 | 4096 | 194.24.16.0/20 |

A set of IP address assignments.

# CIDR – Classless Inter-Domain Routing

- Routing Tables all over the world are now updated with

| Address | Mask | Interface |
|---|---|---|
| **11000010.00011000.00000000.00000000** | **11111111.11111111.11111000.00000000** | **Cambridge** |
| **11000010.00011000.00001000.00000000** | **11111111.11111111.11111100.00000000** | **Edinburgh** |
| **11000010.00011000.00010000.00000000** | **11111111.11111111.11110000.00000000** | **Oxford** |

- ***Example:***
  Look at IP addresses assigned in the next table. What happens when a packet cames in addressed to 194.24.17.4 ?
  Consider a router in Oporto that has 2 outgoing lines, one for New York and another to Paris. What are the changes in its routing table when the entries in the next table come in ?

## CIDR – Classless Inter-Domain Routing

- 194.24.17.4 (?)

  **11000010.00011000.00010001.00000100**
  **11111111.11111111.11111000.00000000 (Cambridge Mask)**
  **11000010.00011000.00010000.00000000**
  Result does no match Cambridge base address so continue...

  **11000010.00011000.00010001.00000100**
  **11111111.11111111.11111100.00000000 (Edinburgh Mask)**
  **11000010.00011000.00010000.00000000**
  Result does no match Edinburgh base address so continue...

  **11000010.00011000.00010001.00000100**
  **11111111.11111111.11110000.00000000 (Oxford Mask)**
  **11000010.00011000.00010000.00000000**
  Result does match Oxford base address!

## CIDR – Classless Inter-Domain Routing

- ***If no longer matches*** are found farther down the table the entry is sent to the correspondent interface.
- The router in Oporto will use its outgoing line to Paris to forward traffic for all the newly assign addresses. So it will aggregate all the three entries in one:
- **194.24.0.0/19 Paris**
- If the unassigned addresses are distributed to a University in California (that will be reached using the New York interface) a new entry must be added to deal with it:
- **194.24.12.0/22 New York**