Internetwork Protocols

Dr Steve Gordon
ICT, SIIT

Contents

- Basic Protocol Functions
- Internetworking with the Internet Protocol
- Internet Protocol Details
- IP Addresses

Basic Protocol Functions

- There are some basic functions used by all protocols
  - Not every protocol implements every function
  - But often the functions are repeated (with different detail/purpose) at different layers in a stack
    - E.g. Data link layer has error control; Transport layer has error control
  - Functions are:
    - Encapsulation
    - Fragmentation and Reassembly
    - Connection control
    - Ordered delivery
    - Flow control
    - Error control
    - Addressing
    - Multiplexing
    - Transmission services

Basic Protocol Functions

- Encapsulation
  - Virtually all protocols transfer data in blocks which include the data and control information (header/trailer)
  - We have referred to the blocks as: messages, frames, datagrams, packets, …
  - A general name is a Protocol Data Unit (PDU)
  - Adding a header (and/or trailer) to a PDU is called Encapsulation
  - Encapsulation occurs at each layer (network layer PDU is encapsulated in a data link layer PDU)
- Fragmentation and Reassembly
  - Usually the size of a PDU used by protocol is limited
  - Fragmentation involves breaking a PDU into smaller blocks for transmission
  - Reassembly is the process at the receiver of combining the received small blocks and creating the original PDU

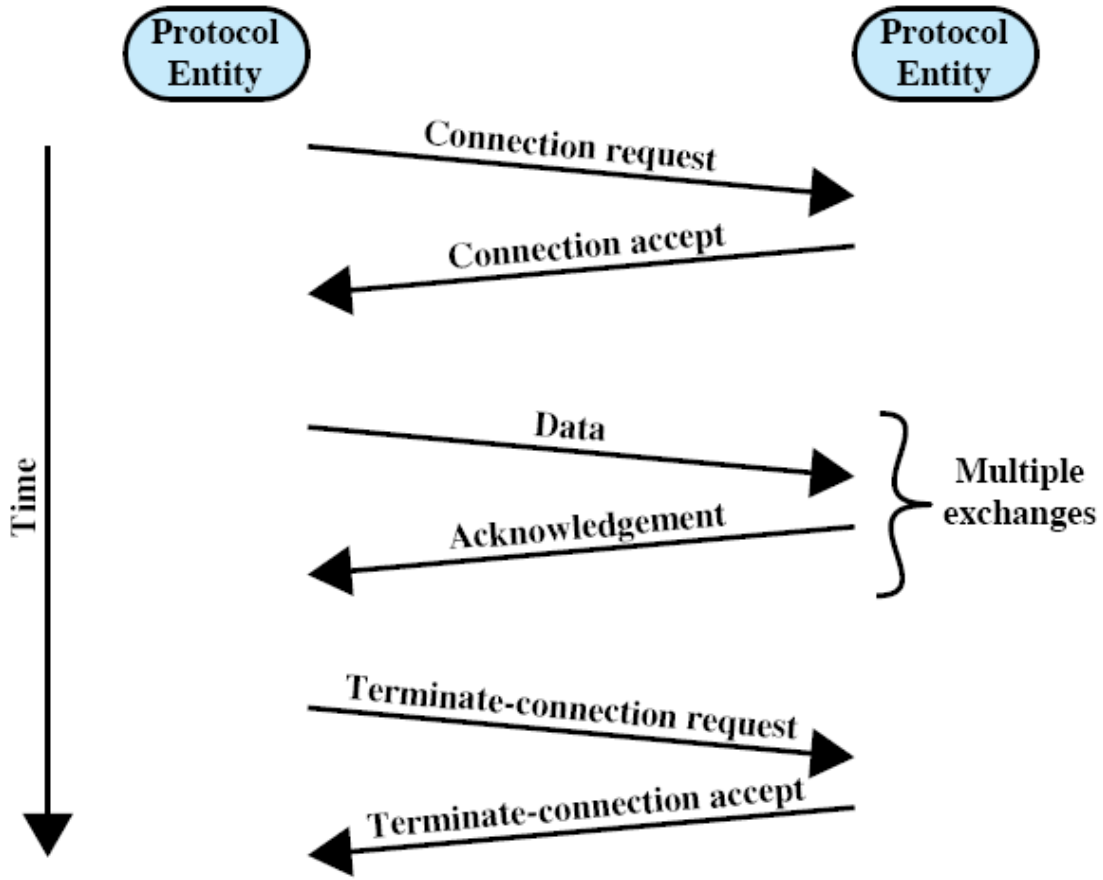
Size of PDUs

- Why break a message into smaller blocks?
  - The communication medium may only accept certain sizes
    - E.g. Ethernet has a maximum size of 1526 bytes; if you have 1MByte file to send, then must be fragmented
  - Error control may be more efficient with smaller PDUs
    - If you send 1 large PDU of 1MB and a single bit error occurs, you have to retransmit the entire 1MB
    - But if you send 1000 small PDUs of 1KB and a single bit error occurs, you only have to retransmit one 1KB PDU
  - Fair access to shared mediums (e.g. MAC protocols)
    - If a station can transmit an unlimited size PDU, the station can steal the medium from others
  - A smaller PDU may mean smaller buffers can be used at the receiver
- What is wrong with breaking a message into smaller blocks?
  - Since each block has a header added, the percentage of overhead will be greater with small PDUs
  - More time is spent processing small, numerous PDUs

Basic Protocol Functions

- Connection Control
  - A protocol may be connectionless or connection-oriented:
    - Connectionless: each PDU sent is treated independently by sender/receiver
    - Connection-oriented: for a set of PDUs, a connection (or association) is setup between sender and receiver before sending
      - Assumes the PDUs are associated with each other, e.g. a file
      - The connection has some parameters (such as maximum PDU size, encryption algorithms, error control parameters) associated with it
  - For connection-oriented, there are 3 phases:
    - Connection establishment (setup)
    - Data transfer
    - Connection termination (teardown)

Connection-Oriented Data Transfer



Basic Protocol Functions

- Ordered Delivery
  - Usually a sequence of PDUs should be delivered to the receiver in order
    - E.g. a sender application has a file that is sent as 10 PDUs; the receiving application must receive the 10 PDUs in order to create the original file
  - But networks can sometimes deliver PDUs out of order
    - E.g. in the Internet, one PDU may take one path, and the second PDU may take a different (shorter) path arriving first
  - Ordering is often achieved using sequence numbers; each PDU is given a sequence number
    - The receiver then knows which order to reassemble the PDUs
    - The sequence number is carried in the header
      - Note that there is a finite number of bits available, hence the sequence number is limited
      - Usually “wrap” the sequence number
        - » 0, 1, 2 , 3, …., 126, 127, 0, 1, 2, …

Basic Protocol Functions

- Flow Control
  - Limit the amount of data sent by transmitter (so don't overflow the receiver)
- Error Control
  - Guard against the damage or loss of data
- Addressing
  - Needed to identify communicating end-points at different levels
    - Network interfaces, software processes, users
- Multiplexing
  - Allow support of multiple connections on one system
- Transmission Services
  - Priority: given certain messages priority over others
  - Quality of service: guarantee performance for certain data/messages, e.g. minimum throughput required, maximum delay tolerated
  - Security: ensure the transmissions are secure

Internetworking with Internet Protocol

Need for Internetworking

- Need for users to access resources outside their local network
- Many different networks deployed
  - Differences in: requirements, owners, technologies
  - Not possible to merge them into a single network
- Therefore, aim to interconnect various networks so any two stations on any of the networks can communicate
- The resulting larger network is referred to as an *internet*
  - The networks are connected by routers and/or bridges
  - Some additional terminology:
    - Intranet: an internet used by a single organisation
    - Subnetwork: a constituent network of an internet
    - the **I**nternet: the name of the internet using the Internet Protocol that is used in the world today
  - Example: SIIT many LAN's interconnected to form an internet; this is the SIIT intranet; the SIIT intranet is a subnetwork of the Internet
- The main form of internetworking today is using routers to form an connectionless internet, called the Internet

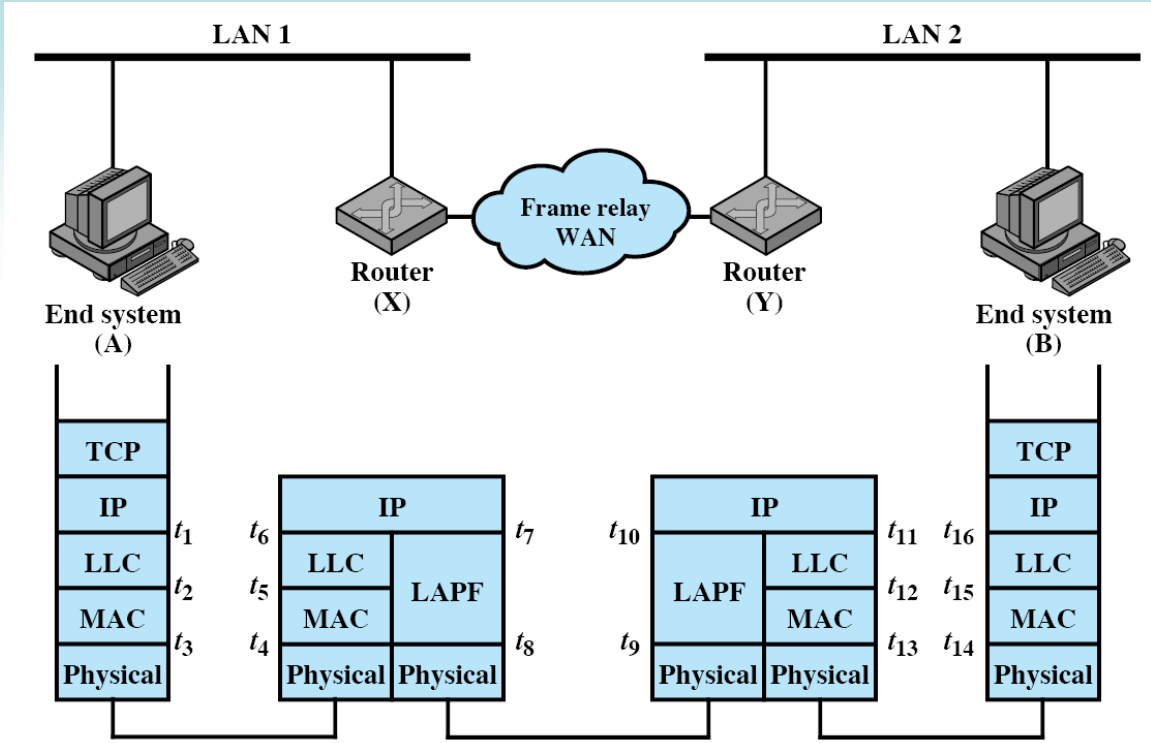
Requirements of an Internetworking Router

- Provide a link between networks
- Provide for routing and delivery of data between applications on different networks
- Provide an accounting service that keeps track of the use of the networks/routers, and maintains status information
- Provide services such that the constituent networks do not require modification. Hence must allow:
  - Different addressing schemes
  - Different maximum packet (PDU) size
  - Different network access mechanisms
  - Different timeouts
  - Error recovery
  - Status reporting
  - Routing techniques
  - User access control
  - Connection-oriented, connectionless

Internet Protocol for Internetworking

- The Internet Protocol (IP) meets some of these requirements
  - Other requirements are met by additional applications
- IP is a connectionless, network layer protocol
- IP was initially developed by US Department of Defence and published as RFC791 in 1981
  - It is now an Internet Standard produced by IETF, and various enhancements have been published and standardised

Operation of IP



$t_1, t_6, t_7, t_{10}, t_{11}, t_{16}$			IP-H	TCP-H	Data	
t_2, t_5		LLC1-H	IP-H	TCP-H	Data	
t_3, t_4	MAC1-H	LLC1-H	IP-H	TCP-H	Data	MAC1-T
t_8, t_9		FR-H	IP-H	TCP-H	Data	FR-T
t_{12}, t_{15}		LLC2-H	IP-H	TCP-H	Data	
t_{13}, t_{14}	MAC2-H	LLC2-H	IP-H	TCP-H	Data	MAC2-T

TCP-H = TCP header
IP-H = IP header
LLCi-H = LLC header
MACi-H = MAC header
MACi-T = MAC trailer
FR-H = Frame relay header
FR-T = Frame relay trailer

Operation of IP

- A has data to send to B
  - The application on A generates data and passes a PDU to the transport layer (TCP), which then passes a PDU to IP
  - IP attaches a header, which includes the global IP address of B
    - The IP address has two parts:
      - Network portion: identifies the network of B
      - Host portion: identifies node B within the network
  - IP recognises B is on another network so it sends the PDU to its local router
    - IP sends the resulting PDU (or IP datagram) to LLC/MAC (t_1) which attach header/trailer
    - The Physical layer sends the bits to the router X
  - At Router X the receiving Physical/MAC/LLC process the PDU and send it to IP (t_6)
    - IP looks at the destination address (B) and makes a routing decision
      - If B is connected directly to a network that router is attached to, then send to B
      - If B is not directly connected, send to a router (which one? Routing protocol)
      - If router does not know of address B, return an error message
    - In the example, router X sends over the WAN link to router Y (t_7)
    - Y performs the same steps as X; this time sending directly to B (t_{11})
  - IP at B receives the PDU (t_{16}) and realises it is the destination, hence passes the data to the application (via TCP)

Design Issues for IP

- Routing
  - Each node (host and router) maintains a routing table, that lists:
    - For each possible destination network, the next router the datagram should be sent to
  - Routing tables may be static, but normally they are dynamic and created/updated by routing protocols
    - We know that a routing protocol can choose routes based on many criteria (number of hops, cost, capacity, delay, security, …)
  - Datagrams between same source/destination pair may take different paths (routes)
  - There are several routing protocols in use today
    - Different protocols for different size networks

Design Issues for IP

- Datagram Lifetime
  - It is possible that a datagram could loop forever in an internet
  - Therefore, each datagram is given a lifetime; after it expires the datagram is no longer forwarded and deleted
  - Lifetime is implemented by hops
    - Datagram starts with Time To Live (TTL) of 255
    - Each router that forward the datagram (that is, each hop), the TTL is decremented by 1
    - If the TTL reaches 0, the datagram is deleted
- Fragmentation and Reassembly
  - Different constituent networks may have different maximum PDU sizes
  - IP routers fragment datagrams if necessary
  - Reassembly occurs at the destination host

Design Issues for IP

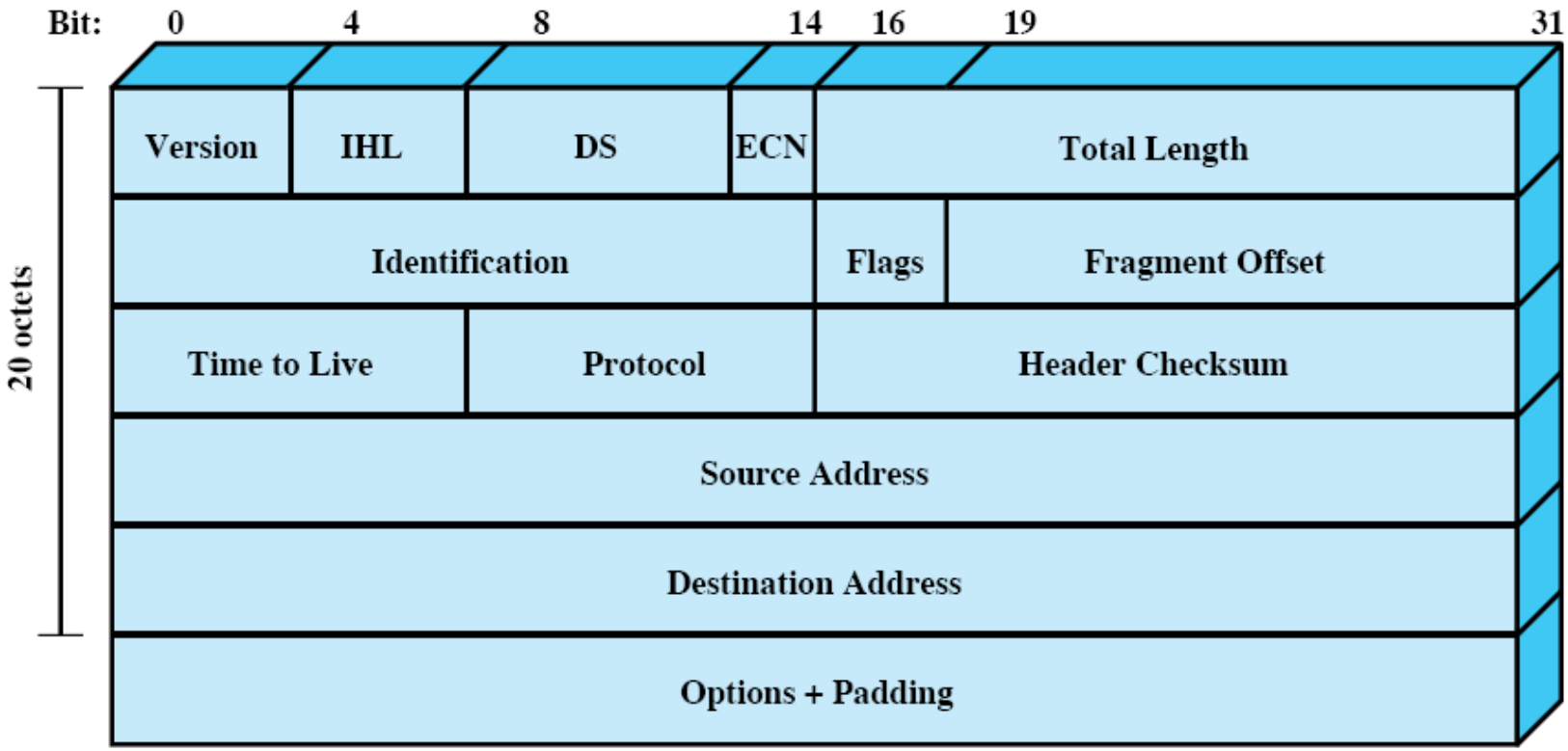
- Error Control
  - IP does not provide any guaranteed delivery
    - No retransmissions
  - If a router discards a datagram (e.g. the router has no more memory, or the datagram is in error), then the router will try to inform the source
- Flow Control
  - There are no complex flow control mechanisms
  - Only possible control is using ICMP to tell the sources to slow down (but not used much)
- Summary:
  - IP does not perform complex error control or flow control; instead leaves these functions to other layers/protocols
  - IP is relatively simple!

Internet Protocol Details

IP Overview

- IP is a network layer protocol
  - It is implemented on all hosts and routers in the Internet
    - For a computer to connect to the Internet, it *must* implement IP
- IP provides a simple send/receive service to the transport layer protocol
  - Send:
    - Transport layer sends data to the IP layer
    - IP adds a header, including source and destination address
    - IP sends the datagram (header + data) to Data Link layer
      - The Data Link layer destination is either the destination host or the next router in the path
  - Receive:
    - Data Link layer sends datagram to IP layer
    - IP checks
      - If destination, then IP removes header and sends data to transport layer
      - If not destination, then send datagram to next node in path
  - The protocol itself is quite simple: the main parts are the header format and addressing scheme

IP Header



Note: For our purposes, octet = byte

IP Header Fields

- **Version**: version number of IP; value is 4 (IPv4)
- **Internet Header Length** (IHL): length of header, measured in 4 byte words; minimum value is 5 (20 bytes)
- **DS/ECN** (or ToS): Used for quality of service control. Differentiated Service, Explicit Congestion Notification, Type of Service
- **Total Length**: total length of the datagram, including header, measured in bytes
- **Identification**: sequence number for datagram
- **Flags**: 2 bits are used for Fragmentation and Re-assembly, the third bit is not used
- **Fragment Offset**: Indicates where this fragment belongs in the original datagram (used for Fragmentation and Re-assembly)
- **Time To Live**: how long datagram should remain in internet
- **Protocol**: indicates the next higher layer protocol with a code
  - E.g. TCP = 6; UDP = 17; ICMP = 1

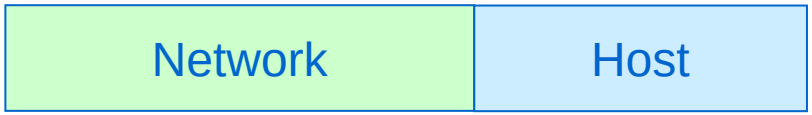
IP Header Fields

- **Header Checksum**: error-detecting code applied to header only (to check for errors in the header); recomputed at each router
- **Source Address**: IP address of source host
- **Destination Address**: IP address of destination host
- **Options**: variable length fields to include options
- **Padding**: used to ensure datagram is multiple of 4 bytes in length
- **Data**: variable length of the data. Maximum length is 65,535 bytes

IP Addresses

IP Addresses

- IP addresses (used to identify source and destination) are 32-bit addresses generally consisting of network portion identifier and host portion identifier



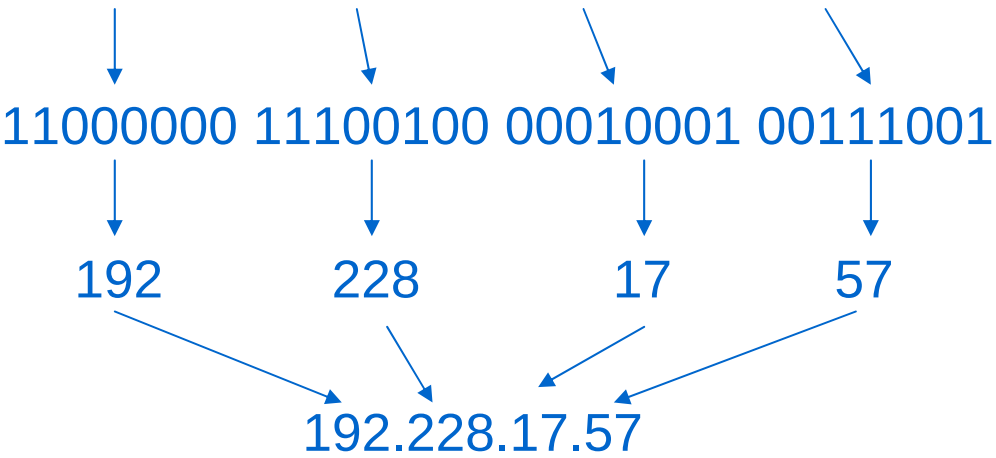
- 32-bits gives 4.2×10^9 possible values
- But IP addresses have different structures (and these have changed over time)
  - First, the set of addresses was separated into five different classes (Classful addressing)
  - Then in 1980's, for organisations to have multiple IP networks (e.g. SIIT Bangkadi is one network, SIIT Rangsit is another), subnet addressing was introduced
  - Then in 1990's, classless addressing was allowed so can fully utilise the address space

Why network and host portion?

- Splitting the IP address into two parts allows for hierarchical addressing. This makes routing in the global Internet possible!
- Example:
  - A reminder: routing protocols provide information to routers about how to reach destinations
    - E.g. “if the destination is D, then send to the next router R”
  - If we did not split the IP address into two ports (that is, flat addressing) then routers must know routes to *hosts*
    - “if the destination is *host H*, then send to the next router R”
      - But on a network, there may be 100’s or 1000’s of hosts
      - Worst case: Routers must know routes to every host on Internet (100,000,000+)
  - But with hierarchical addressing, routers only need to know routes to *networks*
    - “if the destination is *network N*, then send to the next router R”
    - Routers only need to know about hosts on their own network
      - Worst case: routers must know routes to every network on Internet (100,000)

Representing IP Addresses

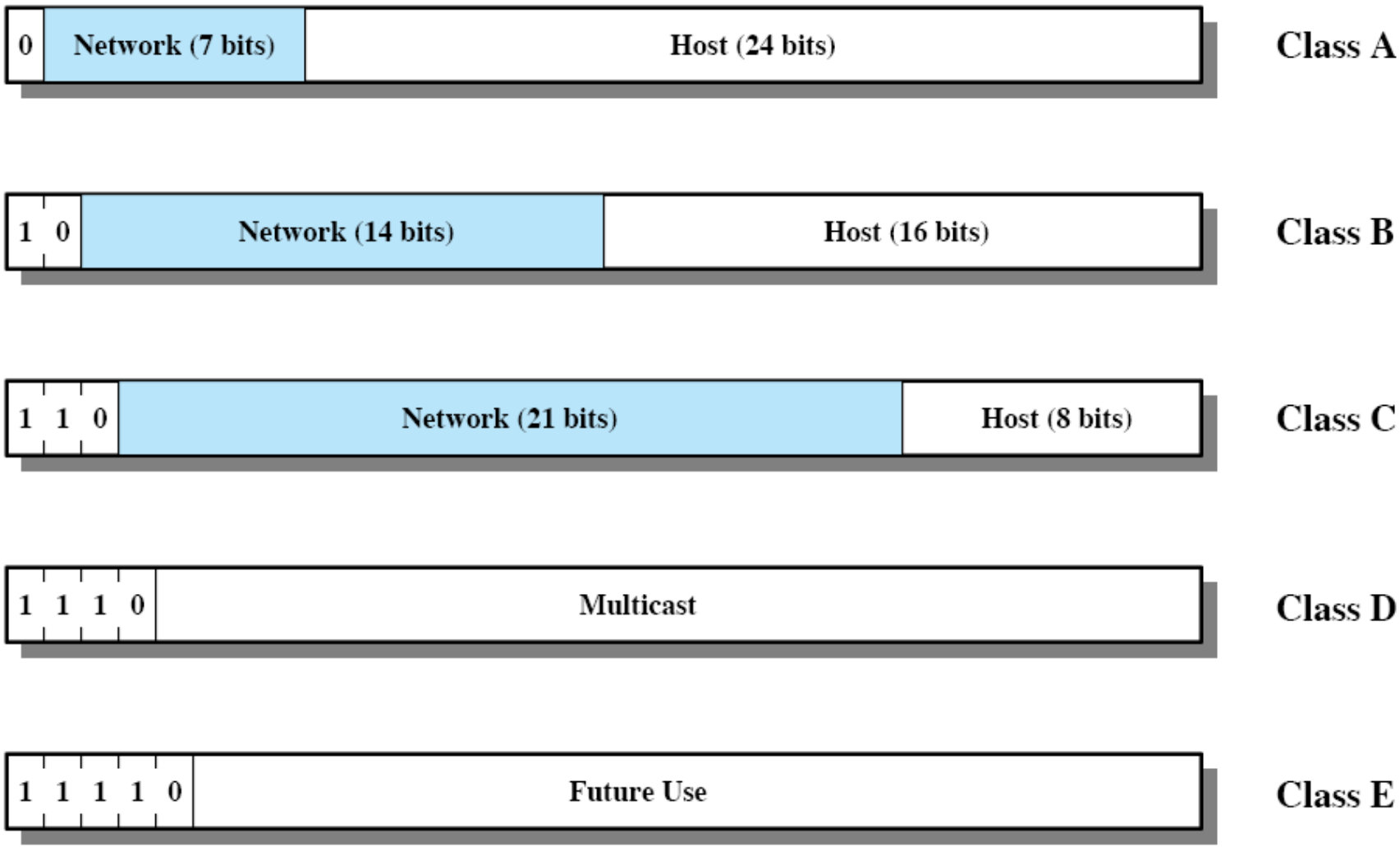
- Writing (and remembering) 32 bits is difficult!
  - 11000000111001000001000100111001
- IP addresses are usually written in dotted decimal notation
  - Decimal number represents of the bytes of the 32 bit address
  - Decimal numbers are separated by dots



IP Address Classes

- Class A: suitable if few networks, many hosts
  - First bit is 0; network portion is 7 bits, host portion 24 bits
  - Note: network addresses with first byte 00000000 (0) or 01111111 (127) are reserved
  - Maximum of 126 networks, each with 16 million hosts
- Class B: medium number of networks, medium number of hosts
  - First 2 bits are 10; network portion 14 bits, host portion 16 bits
  - Maximum of 16384 networks, each with 65534 hosts
- Class C: many networks, each with a few hosts
  - First 3 bits are 110; network portion 21 bits, host portion 8 bits
  - Maximum of 2 million networks, each with 254 hosts
- Class D: use for multicast addressing
  - First 4 bits are 1110
- Class E: reserved for future use
  - First 5 bits are 11110

IP Address Classes



Special Cases for IP Addresses

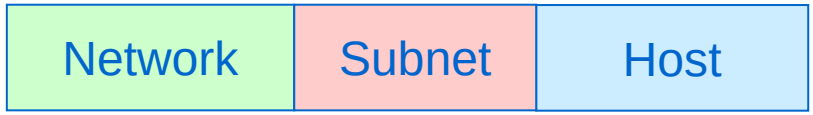
- To identify a network (not an individual host), all binary 0's (decimal 0) is used in the host portion
  - For a host with IP address 130.17.223.4 (Class B), the network address is 130.17.0.0
- To identify broadcast (all hosts on a network, all binary 1's (decimal 255) is used in the host portion
  - On the network 130.17.0.0, to send to all 65534 hosts, the address 130.17.255.255 is used
- To identify the current computer, the address 127.0.0.1 is used

Obtaining an IP Address

- The Internet Assigned Numbers Authority (IANA) manages the assignment of IP addresses
  - If your organisation wants an IP address (or set of addresses) you obtain it through a local or national registry
  - The national registry obtains IP addresses from regional centre, e.g. Asia Pacific Network Information Centre (APNIC)
  - APNIC is assigned address spaces from IANA
- Example:
  - My new company, Steve's Super Solutions, needs IP addresses for its new IP network with around 20 to 30 hosts (and 1 router)
  - Most suitable address type is Class C because not likely to have more than 256 hosts in the network
  - My company is assigned the network address 197.100.7.0, this means I can have 254 computers on my network with addresses 197.100.7.1, 197.100.7.2, 197.100.7.3, …, 197.100.7.254
    - The broadcast address (to send to all computers) is 197.100.7.255
- Problem with Classful Addresses:
  - If every organisation in the world wants a unique network address for each of its IP networks (remember some organisations have many IP networks), then not enough IP addresses!
  - Hence subnet addressing and, eventually, classless addressing were developed

Subnet Addressing

- Assume SIIT has four separate IP networks each with about 500 computers, assigned Class B addresses (maximum of 65534 hosts)
  - Bangkadi: 130.17.0.0, RS1: 130.18.0.0, RS2: 130.19.0.0, RS3: 130.20.0.0
- Using classful addressing, this can be wasteful of IP addresses: 262136 addresses, but only 2000 computers
- Subnet addressing uses an additional *subnet mask*, to separate a network into further subnetworks
  - Now three parts of address:



  - SIIT is assigned 130.17.0.0
    - SIIT network administrator decides to separate into 4 subnetworks (one for Bangkadi, three for Rangsit)
    - Applies a subnet mask to identify the subnet portion of address

Network, 130.17.0.0: 10000010 00010001 00000000 00000000
Subnet mask, 255.255.224.0: 11111111 11111111 11100000 00000000

These 3 bits represent the subnetworks

Subnet Addressing

- Subnet portion of address has 3 bits, 8 values
  - Values 000 and 111 are reserved for local and broadcast addresses
  - So can represent 6 different subnets: 001, 010, 011, 100, 101, 110
    - (We only need 4 in our example)

Subnet 1	10000010 00010001 001 00000 00000000
Subnet 2	10000010 00010001 010 00000 00000000
Subnet 3	10000010 00010001 011 00000 00000000
Subnet 4	10000010 00010001 100 00000 00000000
Subnet 5	10000010 00010001 101 00000 00000000
Subnet 6	10000010 00010001 110 00000 00000000

- For example, the computers on Subnet 1 will have IP addresses:

130.17.32.1	10000010 00010001 001 00000 00000001
130.17.32.2	10000010 00010001 001 00000 00000010
130.17.32.3	10000010 00010001 001 00000 00000011
	…
130.17.63.253	10000010 00010001 001 11111 11111101
130.17.63.254	10000010 00010001 001 11111 11111110

Classless Addressing

- Classful addressing: 3 main classes of addresses, each with limited number of networks and hosts
- Subnet addressing: allow each network to be further divided into subnetworks
- Classless addressing: no classes, instead the split between network portion and host portion can be anywhere
  - Need a “subnet mask” to identify the split
  - Allows many combinations of network/host size, therefore efficient use of IP addresses

IP address, 130.17.41.129: 10000010 00010001 00101001 10000001
Subnet mask, 255.255.252.0: 11111111 11111111 11111100 00000000

Network portion | Host portion

Network, 130.17.40.0: 10000010 00010001 00101000 00000000

  - Used in the Internet today
  - Must give the IP address and subnet address
  - Subnet address can be in dotted decimal or a shorter notation:
    - 130.17.41.129/22 where 22 indicates the first 22 bits of subnet mask are 1’s

Other Network Layer Functions

Internet Control Message Protocol

- IP is used for transferring datagrams between hosts
- ICMP is used for sending feedback about problems between routers and hosts
  - Error reporting examples:
    - Destination unreachable: if a router cannot send an IP datagram to the destination host (e.g. no route because a router or link is down, or the destination doesn't exist), then it returns an ICMP Destination Unreachable message to the source host
    - Echo and Echo Reply: used to test if two computers can communicate. Source sends an ICMP Echo message to destination; if destination receives ICMP Echo, it will respond with ICMP Echo Reply
      - Used by `ping` and other network tools
- ICMP uses IP to send messages

Address Resolution Protocol

- IP (network layer) uses IP addresses to identify each interface. These are logical addresses
- Each constituent network of an internet uses its own (data link layer) addressing mechanism, for example:
  - Ethernet uses IEEE 48 bit addresses
  - Frame Relay and ATM use path/channel identifiers
  - HDLC uses 8 bit addresses

  These are physical addresses.
- A mapping must be made from an IP address to a physical address
  - The Address Resolution Protocol (ARP) is used in each constituent network to perform the mapping
  - ARP creates a table in each host in the local network that has a list of:
    - IP address and corresponding physical address

Internet Address	Physical Address
10.10.1.1	00-50-ba-be-34-cc
10.10.1.101	00-50-ba-87-61-eb
10.10.1.133	00-50-ba-be-34-d2

Other Features

- IPv6
  - The current version is IPv4, uses 32 bit addresses; estimates that there will not be enough addresses in 5 to 15 years
  - IPv6 has been designed to improve/replace IPv4
    - 128 bit addresses (enough for 10^{28} addresses per person!)
    - Still not in widespread use; no strong motivation for ISPs to change from IPv4 to IPv6
- Multicasting
  - Unicast is one-to-one communication
  - Broadcast is one-to-all communication
  - Multicast is one-to-many communication
    - Uses different parts of IP addresses and requires different routing
    - Useful for multimedia communication, e.g. video conferences, TV and audio streaming, collaborative applications
- Quality of Service Control
  - IP provides a “best effort” service: a datagram is sent with no guarantee of arriving at the destination within some time, or at some speed; also no priority is given to datagrams – everyone’s data is treated the same
  - Many multimedia applications (video, voice) work better if they have guaranteed bandwidth/delay, or at least priority over other datagrams
  - QoS includes mechanisms controlling priority of transfer, and guaranteeing certain level of service