COMPUTER NETWORKS

6 August 2025

Dr Noor Mahammad Sk

Course Content

- Network – Requirements – Network architecture – Implementing network software – Performance
- Direct link networks – Encoding – Framing – Error detection – Reliable transmission – Ethernet – Token rings – Wireless
- Packet switching – Forwarding – Bridges – Cell switching – Internetworking – Datagram forwarding – ARP – DHCP – Routing – Multicast
- Protocols – UDP – TCP – Remote procedure call – Congestion control – Congestion avoidance – QoS
- Presentation formatting – Data compression – Cryptographic algorithms – Security mechanisms – Firewalls – Name service and other applications
- Network Management
- Practical aspect of Networking

Reference Books

- **Larry L Peterson** & B. S Davie, Computer Networks A systems Approach, 4Th Edn, Morgan Kauffman Publishers.
- **William Stallings**, Data and Computer Communications, Pearson Education Publishers
- **Tenenbaum A. S**, Computer Networks, 4Th Edn, Prentice Hall.
- **Keshav**, An Engineering Approach to Computer Networks, Addison Wesley.

Computer Networks

- Heterogeneous systems need to talk to each other:
  - Media to connect
    - Wired – twisted pair, coaxial cable, fibre
    - Wireless – radio
  - Topology of the Network
  - Protocols and Software

Network Requirements

- Connectivity
- Cost-Effective Resource Sharing
- Support for Common Services

Network Requirement

Connectivity

Connectivity

- A network must provide connectivity among a set of computers
- Sometimes it is enough to build a limited network that connects only a few select machines
  - For the reasons of privacy and security
  - Many corporate networks
- Networks are designed to grow (scale) in a way that allows them the potential to connect all the computers in the world
  - Internet

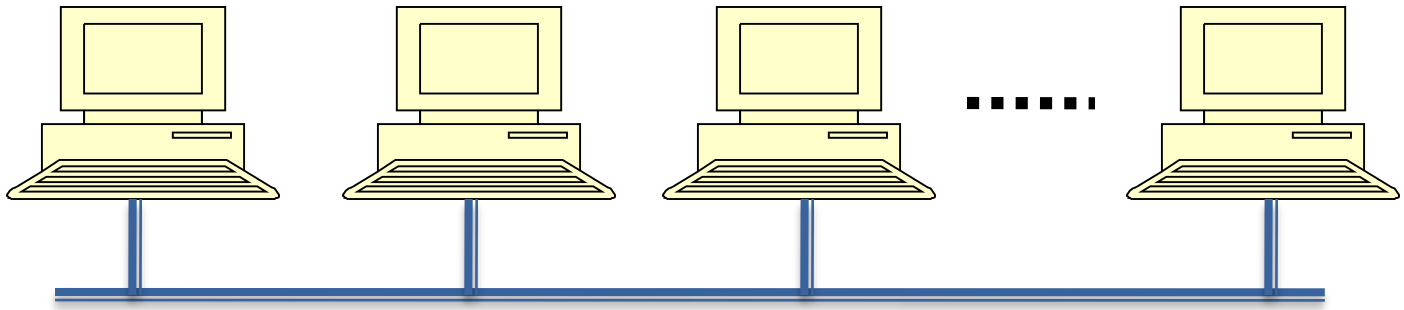
Connectivity – Links, Nodes

- At the lowest level
- A network can consist of two or more computers directly connected by some physical medium
  - A coaxial cable or an optical fiber
- We call such a physical medium a link
- Node: is a specialized piece of hardware rather than a computer
  - We often refer computer as a node

Connectivity – Direct Links

- *Point-to-point*: physical links are sometimes limited to a pair of nodes

- *Multiple Access*: more than two nodes may share a single physical link

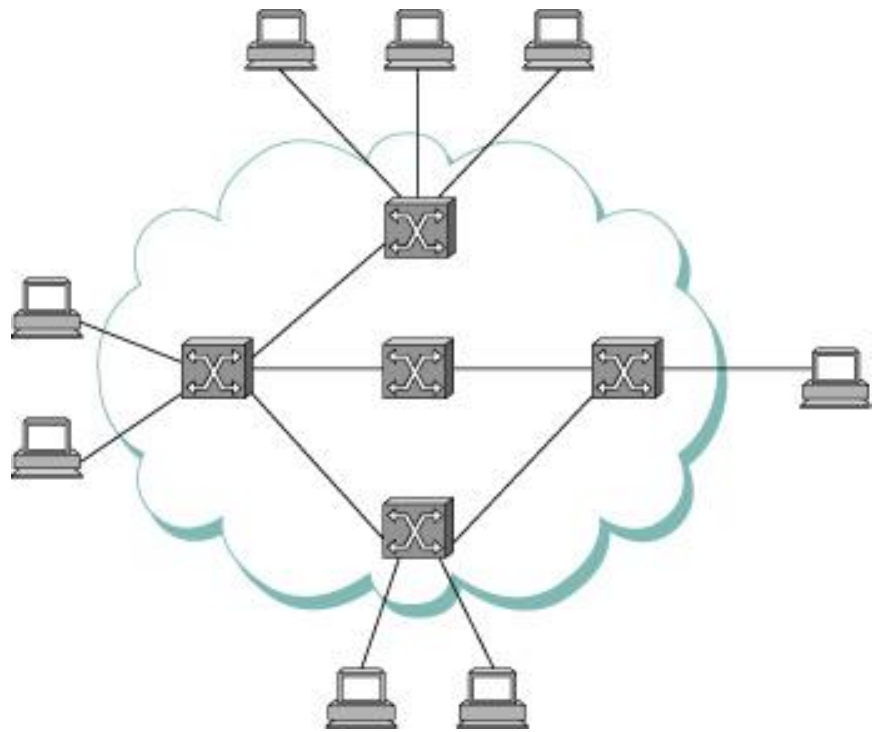


Limitations of Direct Link

- If computer networks were limited to situations in which all nodes are directly connected to each other over a common physical medium
- Then networks would either be very limited in the number of computers they could connect
- Or the number of wires coming out of the back of each node would quickly become both unmanageable and very expensive
- Fortunately, connectivity between two nodes does not necessarily imply a direct physical connection between them
- Indirect connection may be achieved among a set of cooperating nodes

Connectivity – Indirectly connected links

- Figure shows a set of nodes, each of which is attached to one or more point-to-point links
- Those nodes that are attached to at least two links run software that forwards data received on one link out of another
- If organized in a systematic way, these forwarding nodes form a switched network



Switched Networks

- Two most common types are:
  - Circuit switched
    - Employed by the Telephone Systems
  - Packet switched
    - Computer networks
- Packet Switched Network
  - The nodes in such a network send discrete blocks of data to each other
  - These blocks of data corresponding to some piece application data such as a file, a piece of email, or an image
  - We call each block of data either a packet or a message

Packet Switched Networks

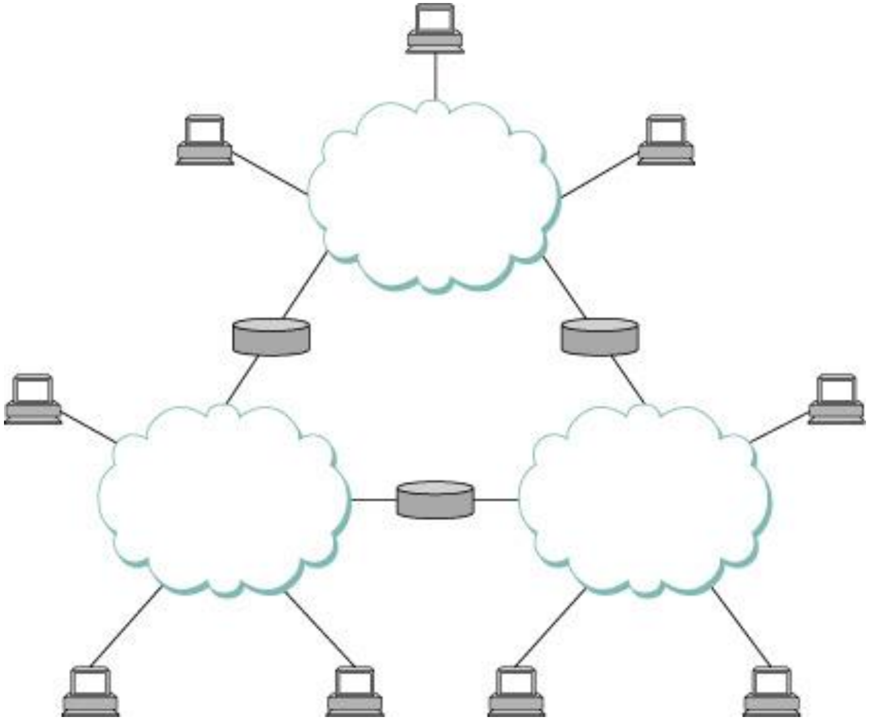
- Typically use a strategy called *store-and-forward*
- Each node in a store-and-forward network first receives a complete packet over some link
- Stores the packet in its internal memory, and then forwards the complete packet to the next node
- In contrast, a circuit switched network first establishes a dedicated circuit across a sequence of links
  - and then allows the source node to send a stream of bits across this circuit to a destination node
- The major reason for using packet switching rather than circuit switching in a computer network is efficiency

Connectivity

- Cloud is one of the important icons of computer networking
  - Commonly called switches, and their primary function is to store and forward packets
- In general we use a cloud to denote any type of network, whether it is a
  - Single point-to-point link
  - Multiple-access link
  - Or a switched network

Interconnection of Networks

- A set of independent networks(clouds) are interconnected to form an internetwork, or internet



Interconnection of Networks

- A set of computers can be indirectly connected
- A node that is connected to two or more networks is commonly called a router or gateway
  - It plays same role as a switch
  - It forwards messages from one network to another
- Internet itself can be viewed as another kind of networks
  - An internet built from an interconnection of internets
- Thus one can recursively build arbitrary large networks by interconnecting clouds to form larger cloud

Connectivity - Address

- A set of hosts are directly or indirectly connected to each other does not mean that we have succeeded in providing host-to-host connectivity
- The final requirement is that each node must be able to state which of the other nodes on the network it wants to communicate with
- This is done by assigning an *address* to each node
- An address is a byte string that identifies a node
- The network can use a node's address to distinguish it from the other nodes connected to the network

Connectivity – routing

- When a source node wants the network to deliver a message to a certain destination node
  - It specifies the address of the destination node
- If the sending and receiving nodes are not directly connected
  - Then the switches and routers of the network use this address to decide how to forward the message toward the destination
- The process of determining systematically how to forward messages toward the destination node based on its address is called routing

Unicast, Broadcast and Multicast

- Unicast:
  - The source node wants to send a message to a single destination node
- Broadcast:
  - The source node might want to broadcast a message to all the nodes on the network
- Multicast:
  - A source node might want to send a message to some other subset of the other nodes
  - But not all of them

Network Requirement

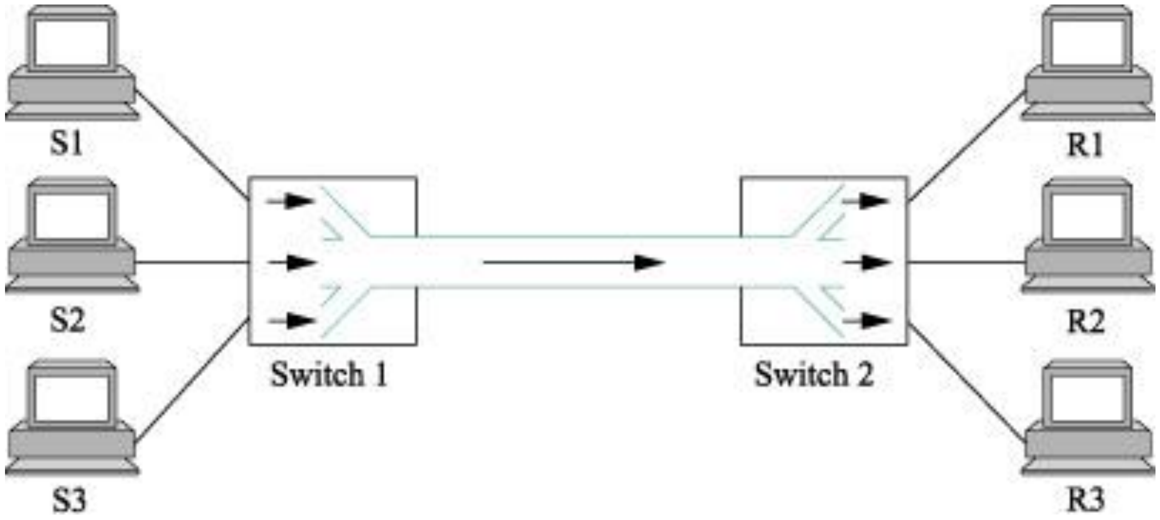
Cost-Effective Resource Sharing

How hosts share a Network

- Any one pair of hosts can exchange the messages across a sequence of links and nodes
- Will network support more than one pair of communication between the hosts
- Multiplexing:
  - A system resource is shared among multiple users
  - Analogy to a timesharing computer system
  - Data being sent by multiple users can be multiplexed over the physical links that make up a network

Multiplexing

- Three hosts on the left side of the network (senders S1 – S3) are sending data to the three hosts on the right (receivers R1 – R3) by sharing switched networks that contains only one physical link



Multiplexing

- Three flows of data – corresponding to the three pairs of hosts – are multiplexed onto a single physical link by switch1
- The demultiplexed back into separate flows by switch 2
- Multiplexing Methods
  - Synchronous time division multiplexing (STDM)
  - Frequency division multiplexing (FDM)

STDM

- Idea is to divide time into equal-sized quanta and, in a round-robin fashion, give each flow a chance to send its data over the physical link
- In other words, during
  - Time quantum 1, data from S1 to R1 is transmitted
  - Time quantum 2, data from S2 to R2 is transmitted
  - In time quantum 3, data from S3 to R3 is transmitted
  - In the next time quantum, the first flow (S1 to R1) gets go again,
  - and the process repeats

FDM

- The idea is to transmit each flow over the physical link at different frequency
- Much the same way that the signals for different TV stations are transmitted at a different frequency on a physical cable TV link

Limitations of STDM and FDM

- If one of the flows (host pairs) does not have any data to send,
  - Its share of the physical link
  - That is, its time quantum or its frequency – remain idle
  - Even if one of the other flows has data to transmit
- For computer communication, the amount of time that a link is idle can be very large
  - Example: consider the time you spend reading a web page (leave the link idle) compared to the time you spend fetching the page

Limitations of STDM and FDM

- The maximum number of flows is fixed and known ahead of time
  - It is not practical to resize the quantum or to add additional quanta in the case of STDM
  - or to add new frequencies in the case of FDM

Statistical Multiplexing

- It is like STDM in that the physical link is shared over time
  - First data from one flow is transmitted over the physical link, then data from another flow is transmitted, and so on.
- Data is transmitted from each flow on demand rather than during a predetermined time slot
  - If only one flow has data to send, it gets to transmit that data without waiting for its quantum to come around
  - Thus without having to watch the quanta assigned to the other flows go by unused
- It is avoidance of idle time that gives packet switching its efficiency

Limitations of Statistical Multiplexing

- No mechanism to ensure that all the flows eventually get their turn to transmit over the physical link
- Once flow begins sending a data, we need some way to limit the transmission, so that the other flows can have a turn
- An upper bound on the size of the block of data that each flow is permitted to transmit at a given time
- This limited size block of data is typically referred to as a packet

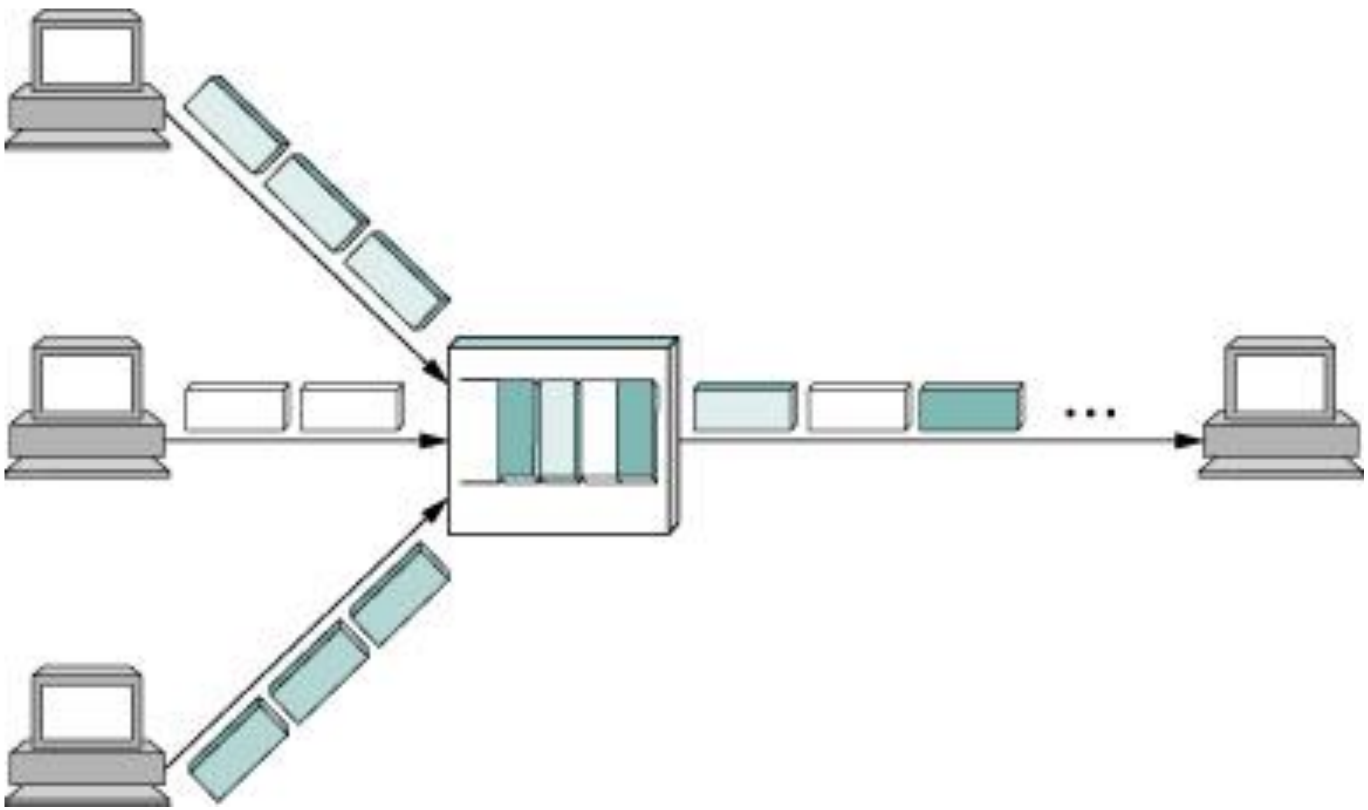
Packets

- Packet switched network limits the maximum size of packets
- A host may not be able to send a complete message in one packet
- The source may need to fragment the message into several packets, with the receiver reassembling the packets back into the original message

Single Shared Link

- Each flow sends a sequence of packets over the physical link, with a decision made on a packet-by-packet basis as to which flow's packet to send next
- If only one flow has data to send, then it can send a sequence of packets back-to-back
- If more than one of the flows have data to send, then their packets are interleaved on the link

Multiple flows on a Shared Link



A Switch multiplexing Packets from Multiple Sources onto one Shared Link

Packet Switching Decision

- Each switch in a packet switched network makes this decision independently, on a packet-by-packet basis
- One of the issues that faces a network designer is how to make decision in a fair manner
- A switch could be designed to service packets
  - On a first-in-first-out (FIFO) basis
  - Round-robin manner

QoS

- Switching might be done to ensure that certain flows receive a particular share of the link's bandwidth,
- Or that they never have their packets delayed in the switch for more than a certain length of time
- A network that attempts to allocate bandwidth to particular flows is sometimes said to support *quality of service (QoS)*

Congestion

- In the above example the switch has to multiplex three incoming packet streams onto one outgoing link
- It is possible that the switch will receive packets faster than the shared link
  - In this case this case, the switch is forced to buffer these packets in its memory
- Switch receive packets faster than it can send them for an extended period of time
- Switch will eventually run out of buffer space, and some packets will have to be dropped
- This state of switch operation is called as *congested*

Requirements

Support for Common Services

Applications Programs on Networks

- In simple the computer network is delivering packets among a collection of computers
- A network as providing the means for a set of application processes that are distributed over those computers to communicate
- The application programs running on the hosts connected to the network must be able to communicate in a meaningful way

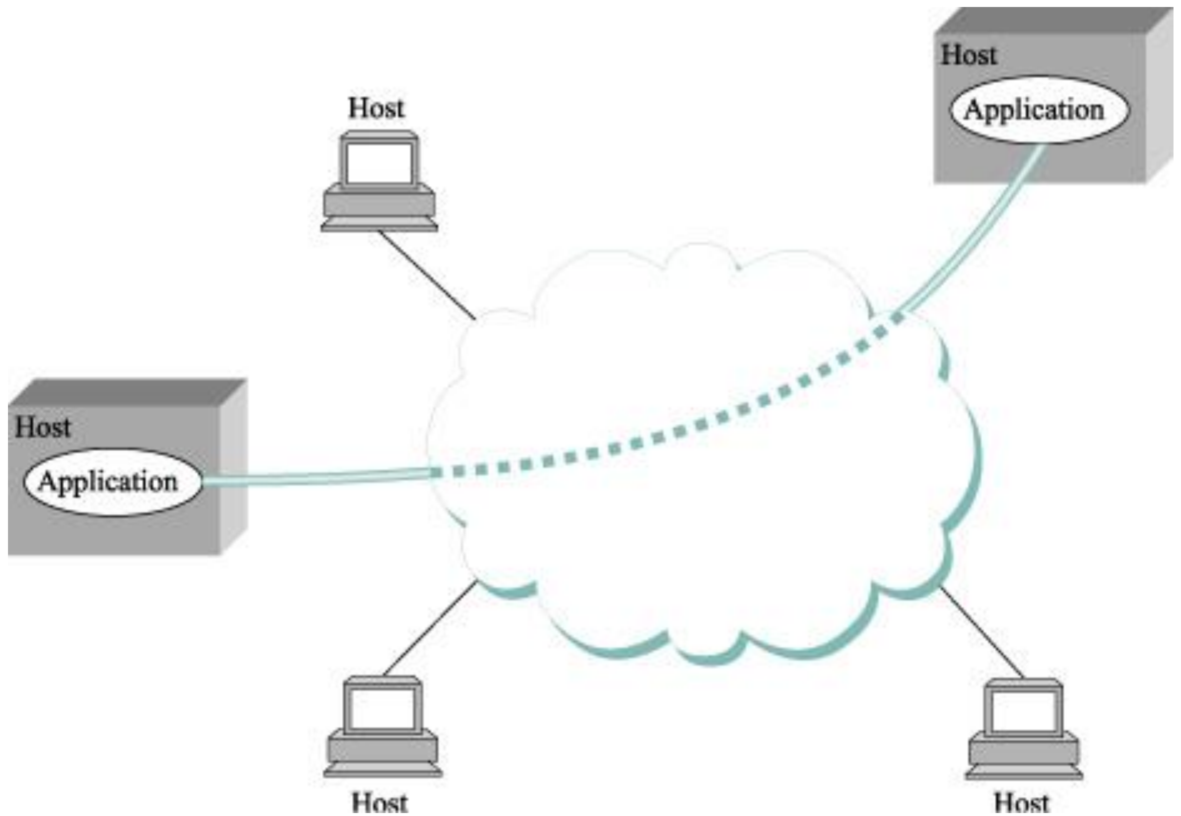
Applications Programs on Networks

- When two applications need to communicate with each other
- There are a lot of complicated things that need to happen beyond simply sending a message from one host to another
- One option would be for application designers to build all that complicated functionality into each application program
- Many applications need common services, it is much more logical to implement those common services once
  - Such that application designer build the application using those services

Network Common Services

- Network provides logical channels over which application-level processes can communicate with each other
- Each channel provides the set of services required by that application
- A cloud is abstractly represent connectivity among a set of computers
- A channel connects one process to another
- A channel is like a pipe connecting two applications
    - Sending application can put data in one end and expect that data to be delivered by the network to the application at the other end of the pipe

Channel



Processes communicating over an abstract channel

Identifying Common Communication Patterns

- Involves
  - Understanding the common needs of a representative collection of applications
  - Extracting their common communication requirements
  - Incorporating the functionality that meets these requirements in the network
- One of the earliest applications supported on any network is a file access program
  - FTP
  - NFS

Identifying Common Communication Patterns

- FTP/ NFS
  - Whole files are transferred across the network or only single blocks of the file are read/written at a given time
- The communication component of remote file access is characterized by a pair of processes
  - One that requests that a file be read or written
    - Client
  - A second process that honors this request
    - Server

Read and Write – Server & Client

- Reading a file involves
  - The client sending a small request message to a server
  - The server responding with a large message that contains the data in the file
- Writing works in opposite way
  - The client sends a large message containing the data to be written to the server
  - The server responds with a small message confirming that the write to disk has taken place

Video Applications

- Two types of applications
  - Videoconferencing
  - Video on demand
- Channels are
  - Request/reply channels
  - Message stream channels
- Request/reply channel would be used by the file transfer and digital library applications

Request/reply Channel

- It would guarantee that every message sent by one side is received by the other side
  - only one copy of each message is delivered
- The request/reply channel might also protect the privacy and integrity of the data that flows over it
  - Unauthorized parties cannot read or modify the data being exchanged between the client and server processes

Message Stream Channels

- Used by both the vide-on-demand and videoconferencing applications
- It is parameterized to support both one-way and two-way traffic and to support different delay properties
- The message stream channel might not need to guarantee that all messages are delivered
- A video application can operate adequately even if some video frame is not received
- The message stream channel might need to support multicast
  - So that multiple parties can participate in the teleconference or view the video

Channels/pipes

- A network designer to strive for the smallest number of abstract channel types that can serve the largest number of applications
  - There is danger in trying to get away with too few channel abstractions
- With change in application the network designers will probably be inventing new types of channels and adding options to existing channels
  - for as long as application programmers are inventing new applications

Bit Pipe

- It is easiest to view host-to-host connectivity of the underlying network as simply providing a bit pipe
  - With any high-level communication semantics provided at the end hosts
- The advantage of this approach is it keeps the switches in the middle of the network as simple as possible
  - They simply forward packets
  - But it requires the end hosts to take on much of the burden of supporting semantically rich process-to-process channels
- The alternative is to push additional functionality onto the switches, thereby allowing the end hosts to be "dumb" devices

Reliability

- Reliable message delivery is one of the most important functions that a network can provide
- It is difficult determine how to provide this reliability
  - Without understanding how networks can fail
- The computer networks do not exist in a perfect world
  - Physical Problems
    - Machine crash and later are rebooted
    - Fiber are cut
    - Electrical interference corrupts bits in the data being transmitted
    - Switches run out of buffer space
  - Software
    - The software that manages the hardware sometimes forwards packets into oblivion

Reliability – Bit Level Failure

- The major requirement of a network is to recover from certain kind of failures
- Three general cases of failures in the network is
- Bit Errors
  - The packet transmitted over a physical link, bit errors may be introduced into the data
    - i.e., 1 is turned into a 0 or vice versa
- Burst error – several consecutive bits are corrupted
- Bit errors typically occurs because the outside forces
  - Such as lightening strikes, power surges and microwave ovens, interface with the transmission of data

Reliability

- The good news is that bit errors are fairly rare
  - Affecting on average only one out of every 10^6 to10^7 bits on a typical copper base cable
  - One out of every 10^{12} to 10^{14} bits on a typical optical fiber
- There are techniques that detect these bit errors with high probability
- It is sometimes possible to correct for such errors
- If damage is so bad, than it is necessary to discard the entire packet
  - In such a case, the sender may be expected to retransmit the packet

Reliability – Packet Failure

- Failure is at the packet, rather than the bit level
- i.e., a complete packet is lost by network
- One reason this can happen is that the packet contains an uncorrectable bit error and therefore has to be discarded
- A switch that is forwarding it from one link to another – is so over loaded that it has no place to store the packet, and therefore is forced to drop it (congestion)

Reliability – Packet Failure

- Software running on one of the nodes that handles the packet makes a mistake
  - It might incorrectly forward a packet out on the wrong link
  - So packet never finds its way to the ultimate destination
- One of the main difficulties in dealing with lost packets is distinguished between a packet that is indeed lost and one that is merely late in arriving at the destination

Reliability – Node and Link Failure

- Failure is at node and link level
- A physical link is cut or the computer it is connected to crashes
- This can be caused by software that crashes a power failure, or a reckless backhoe operator
- Failure due to misconfiguration of a network device are also common
- Any of these failures can eventually be corrected, they can have a dramatic effect on the network for an extended period of time

Reliability – Node and Link Failure

- In a packet switched network, it is sometimes possible to route around a failure node or link

NETWORK ARCHITECTURE

6 August 2025

Dr Noor Mahammad Sk

Network Architecture

- A computer network must provide general, cost-effective, fair, and robust connectivity among a large number of computers
- To deal with the network complexity, network designers have to develop a general blueprint – usually called network architectures
- Network architectures guide the design and implementation of network
- Two of the most widely referenced architectures –
  - The OSI architecture
  - The Internet architecture

Network Architecture

Layering and Protocols

Layering – Abstraction

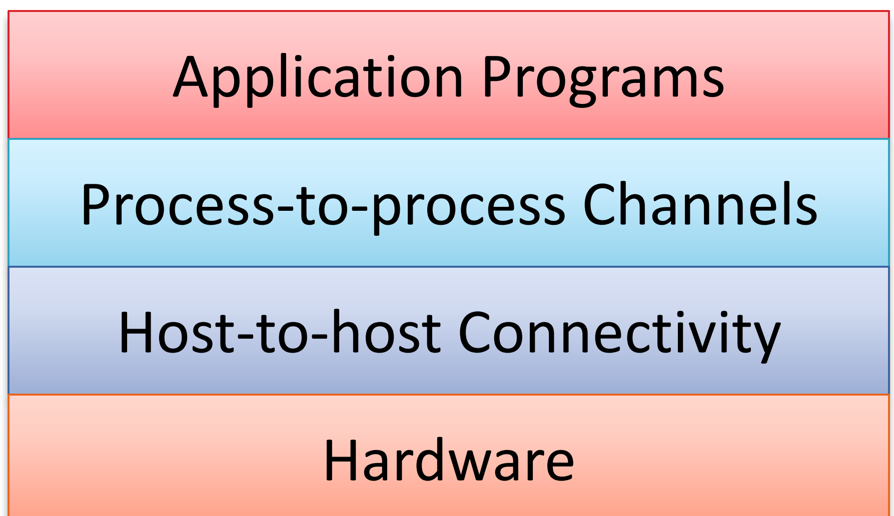
- When a system gets complex, the system designer introduces another level of abstraction
- An abstraction is to define a unifying model that can capture some important aspect of the system
- An abstraction for applications that hides the complexity of the network from application writers
- Abstraction normally leads to layering in network systems

Layering

- The general idea is to start with the services offered by the underlying hardware and
- Add a sequence of layers, each providing a higher (more abstract) level of service.
- The services provided at the high layers are implemented in terms of the services provided by the low layers

Layering – Example

- A simple network as having two layers of abstraction sandwiched between the application program and underlying hardware



Example of a layered network system

Layering – Example

- The layer immediate above the hardware in this case might provide host-to-host connectivity
  - Abstracting away the fact that there may be an arbitrarily complex network topology between any two hosts
- The next layer up builds on the available host-to-host communication service and provides support for process-to-process channels

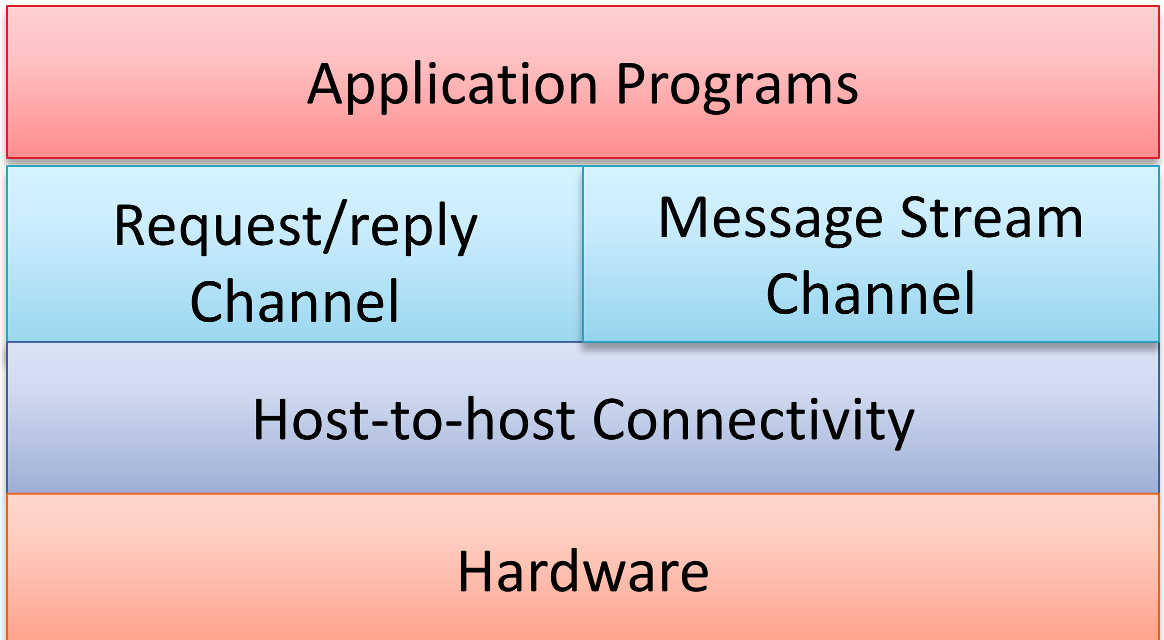
Layering

- Layering offers two features
- it decomposes the problem of building a network into more manageable components
  - One can implement several layers, each of which solves one part of the problem
- It provides a more modular design
  - If you want to add some new services, you may only need to modify the functionality at one layer
  - By reusing the functions provided at all the other layers

Layering – Multiple Abstraction

- Multiple abstractions are provided at any given level of the system
  - Each provides a different services to the higher layers but building on the same low-level abstractions
- Lets consider the two channels
  - A request/reply services
  - A message stream services
- These two channels might be alternative offerings at some level of a multilevel networking system

Layering – Multiple Abstraction



Layered System with alternative abstraction available at a given layer

Protocol

- The abstract objects that make up the layers of a network system are called protocols
- A protocol provides a communication service that higher-level objects use to exchange the messages
- Example
  - Imagine a network that supports a request/reply protocol and a message stream protocol

Protocol – Interfaces

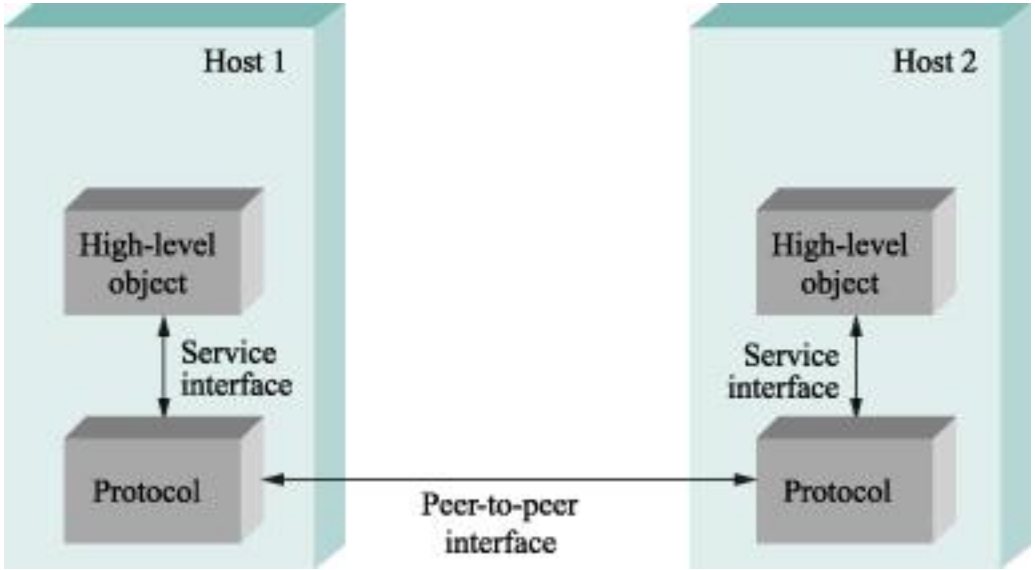
- Each protocol defines two different interfaces
- It defines a *service interface* to the other objects on the same computer that want to use its communication services
  - This service interface defines the operations that local objects can perform on the protocol
- Examples
  - A request/reply protocol would support operations by which an application can send and receive messages
  - An implementation of HTTP protocol support an operation to fetch a page of hypertext from a remote server

Protocol – Interfaces

- A protocol defines a *peer interface* to its counterpart (peer) on another machine
- This interface defines the form and meaning of messages exchanged between protocol peers to implement the communication service
  - This would determine the way in which a request/reply protocol on one machine communicates with its peer on another machine
- Example: In the case of HTTP
  - The protocol specification defines in detail how a "GET" command is formatted
  - What arguments can be used with command

Protocol

- A protocol defines a communication services that it exports locally (the service interface)
- Along with a set of rules governing the messages that the protocol exchanges with its peer(s) to implement this service (the peer interface)



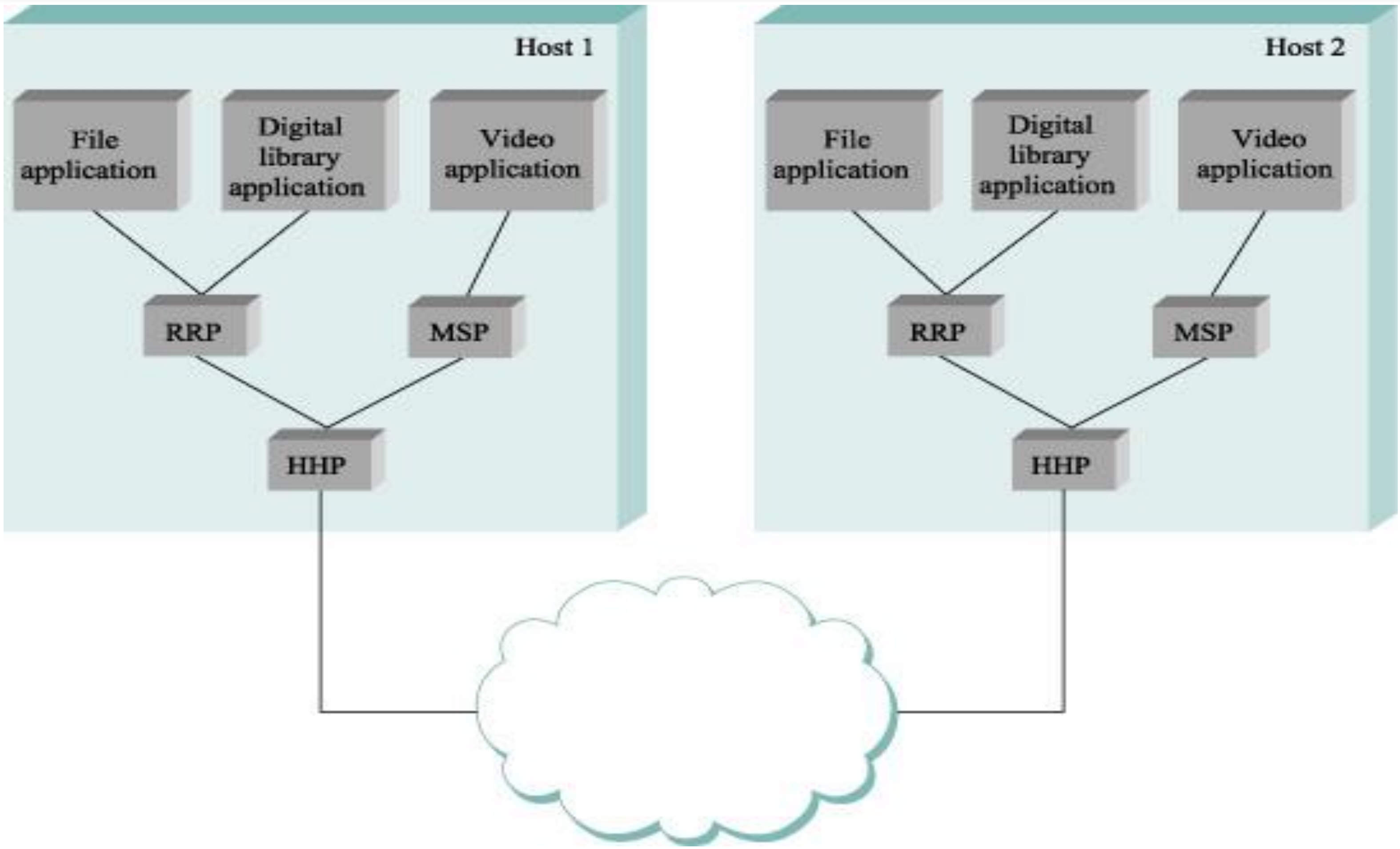
Protocol

- Peer-to-peer communication is indirect
  - Except at the hardware level where peers directly communicate with each other over a link
- Each protocol communicates with its peer by passing messages to some lower-level protocol
  - Which in turn delivers the message to its peers
- In addition, there are potentially multiple protocols at any given level, each providing a different communication service
- Therefore represent suite of protocols that make up a network system with a protocol graph

Protocol Graph

- The nodes of the graph correspond to protocols, and the edges represent a *depends on* relation
- Example
- The request/reply protocol (RRP) and message stream protocol (MSP) implement two different types of process-to-process channels
- Both depend on Host-to-host protocol (HHP), which provides a host-to-host connectivity service

Protocol Graph – Example



Protocol Graph – Example

- Host 1:
  - The file access program on host 1 wants to send a message to its peer on host 2 using communication services offered by protocol RRP
  - In this case, the file application asks RRP to send the message on its behalf
  - To communicate with its peer, RRP then invokes the services of HHP
    - which in turn transmits the message to its peer on the other machine
- Host 2:
  - Once the message has arrived at protocol HHP on host 2
  - HHP passes the message up to RRP, which in turn delivers the message to the file application

Protocol – Interface Vs Module

- Protocols is used in two different ways
  - Interfaces – the operations defined by the service interface and the form and meaning of messages exchanged between peers
  - Modules – that actually implements required interfaces
- Protocol specification distinguish the given protocol is an interface type or module type

Protocol – Specifications

- Generally expressed using combination of
  - prose, pseudocode, state transition diagrams, pictures of packet formats and other abstract notations
- A given protocol can be implemented in different ways by different programmers
- The challenge is to ensuring that two different implementations of the same specification can successfully exchange messages
- Two or more protocol modules that do accurately implement a protocol specification are said to *interoperate* with each other

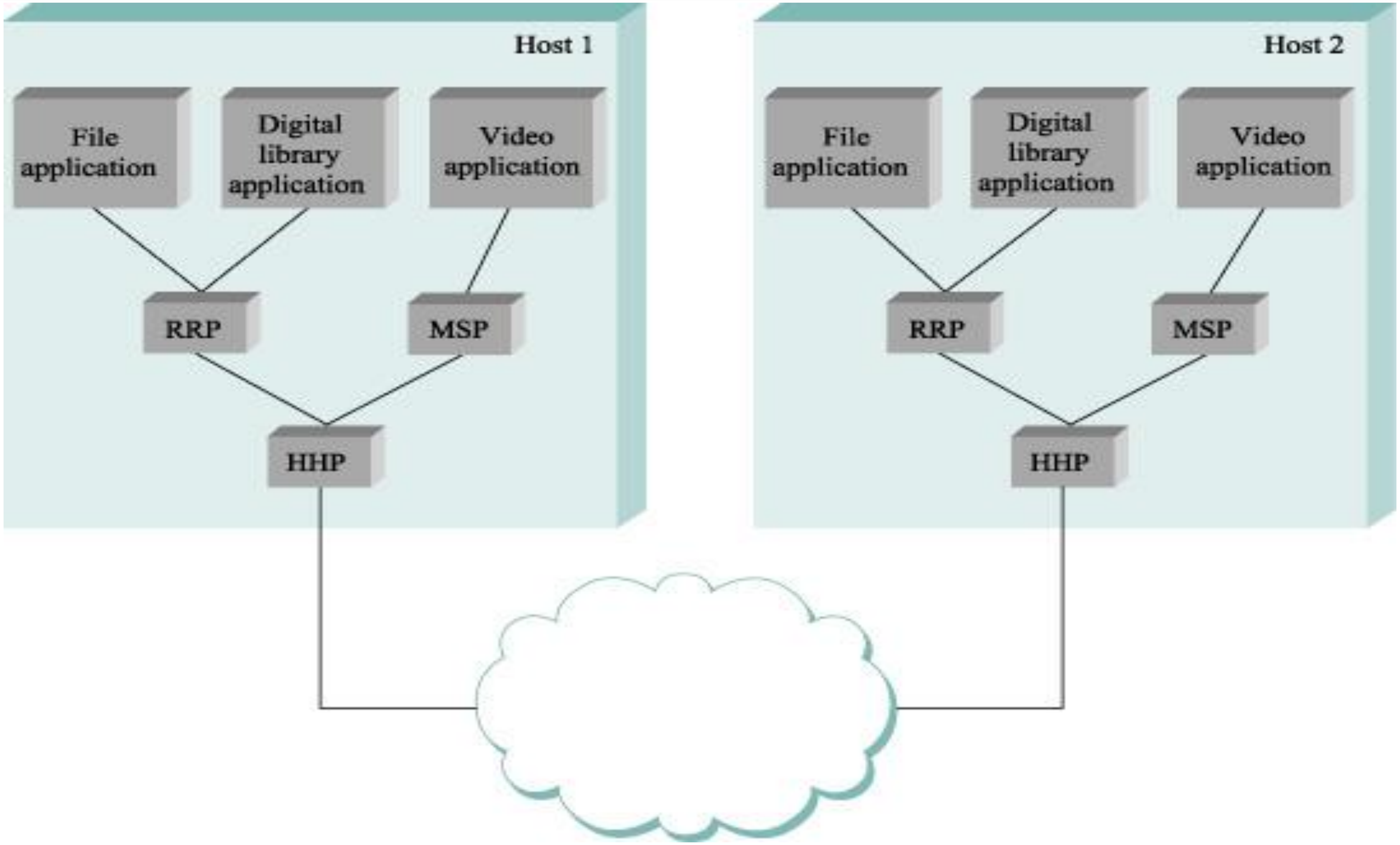
Protocols

- We can imagine many different protocols and protocol graphs that satisfy the communication requirements of a collection of applications
- Standardization:
  - International Standard Organization (ISO)
  - Internet Engineering Task Force (IETF)
- Establish/defines policies for a particular protocol graph
- Network Architecture: The set of rules governing, the form and content of a protocol graph

Network Architecture

Encapsulation

Example – A protocol Graph



Encapsulation

- When one of the application programs sends a message to its peer by passing the message to protocol RRP
- From RRP perspective, the message it is given by the application is an uninterpreted string of bytes
- RRP does not care that these bytes represent an array of integers, an email message, a digital image, or whatever
  - It simply charged with sending them to its peer
- However, RRP must communicate control information to its peer, instructing it how to handle the message when it is received

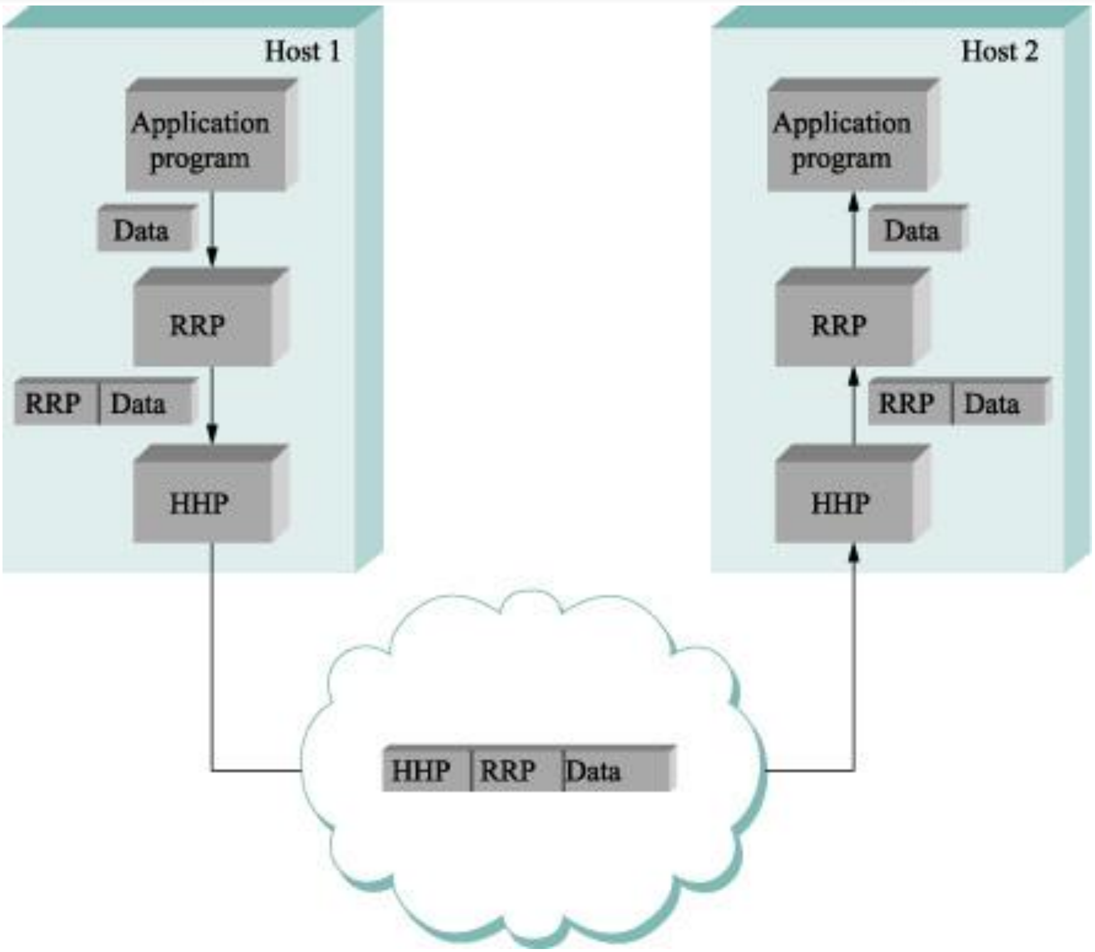
Encapsulation

- RRP does this by attaching a header to the message
- A header is a small data structure
  - From a few bytes to a few dozen bytes
  - Used among peers to communicate with each other
- Headers are usually attached to the front of a message
- In some cases, this peer-to-peer control information is sent at the end of the message
  - Called as a trailer
- The exact format for the header attached by the RRP is defined by its protocol specification

Encapsulation

- The rest of the message
  - The data being transmitted on behalf of the application is called the message's *body* or *payload*
  - It means application's data is encapsulated in the new message created by protocol RRP
- This process of encapsulation is then repeated at each level of the protocol graph

Encapsulation Example



High level messages are encapsulated inside of low-level message

Example

- HHP encapsulate RRP's message by attaching a header of its own
  - If we now assume that HHP sends the message to its peer over some network
  - Then when the message arrives at the destination host, it process it in the opposite order
- HHP first interprets the HHP header at the front of the message (i.e., takes whatever action is appropriate given the contents of the header),
- Passes the body of the message (but not the HHP header) up to the application program

Example

- The message passed up from RRP to the application on host 2 is exactly the same message as the application passed down to RRP on host 1
- The application does not see any of the headers that have been attached to it to implement the lower-level communication services
- Node in the network (e.g., switches and router) may inspect the HHP header at the front of the message

Encapsulation

- A low-level protocol does not interpret the message it is given by some high-level protocol
- It does not know how to extract any meaning from the data contained in the message
- Encryption
  - The low-level protocol applies some simple transformation to the data it is given, such as to compress or encrypt it
  - In this case, the protocol is transforming the entire body of the message, including both the original *application's data* and *all the headers* attached to that data by higher-level protocols

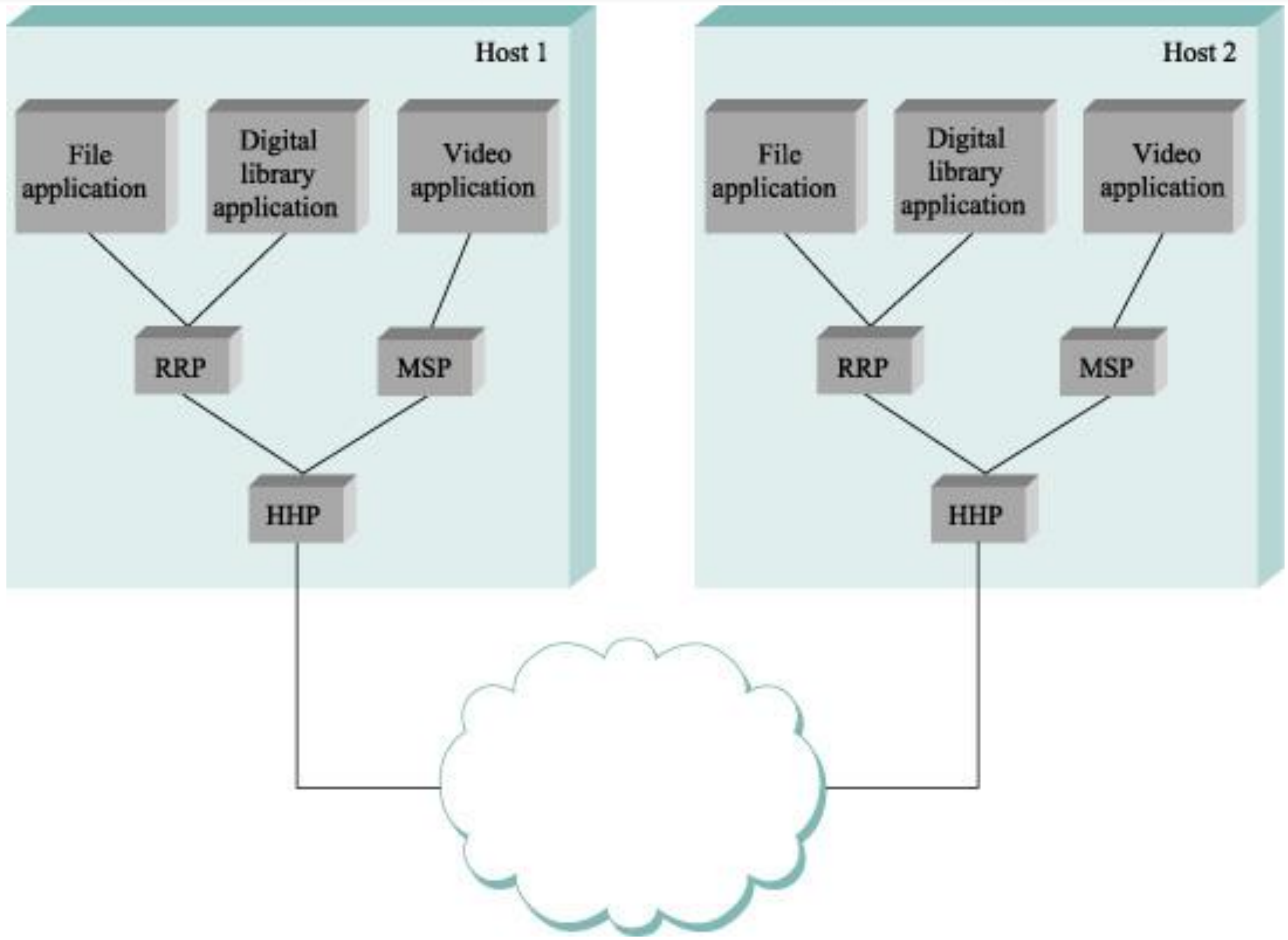
Network Architecture

Multiplexing and Demultiplexing

Multiplexing and Demultiplexing

- The fundamental idea of packet switching is to multiplex multiple flows of data over a single physical link
- The same idea applies up and down the protocol graph, not to switching nodes

Example


Host 1
File application
Digital library application
Video application
RRP
MSP
HHP
Host 2
File application
Digital library application
Video application
RRP
MSP
HHP

Multiplexing and Demultiplexing

- RRP implementing a logical communication channel, with message from two different applications multiplexed over this channel at the source host
- And then demultiplexed back to the appropriate application at the destination host
- Practically
  - A header that RRP attaches to its message contains an identifier that records the application to which the message belongs
  - This identifier is called as RRP's *demultiplexing key* or *demux key*

Multiplexing and Demultiplexing

- At the source host, RRP includes the appropriate demux key in its header
- When the message is delivered to RRP on the destination host
  - It strips its header, examines the demux key and demultiplexes the message to the correct application
- RRP is not unique in its support for multiplexing
  - Nearly every protocol implements this mechanism
- Example:
  - HHP has its own demux key to determine which messages to pass up to RRP and which to pass up to MSP

Multiplexing and Demultiplexing

- There is no uniform agreement among protocols
- Even those within a single network architecture
  - On exactly what constitutes a demux key
- Some protocols use 8-bit field (meaning they can support only 256 high-level protocols) and others use 16- or 32-bit fields
- Some protocols have a single demultiplexing field in their header
  - The same demux key is used on both sides of the communication
- Some protocols have a pair of demultiplexing fields in their header
  - Each sides uses a different key to identify the high-level protocol (or application program) to which the message is to be delivered

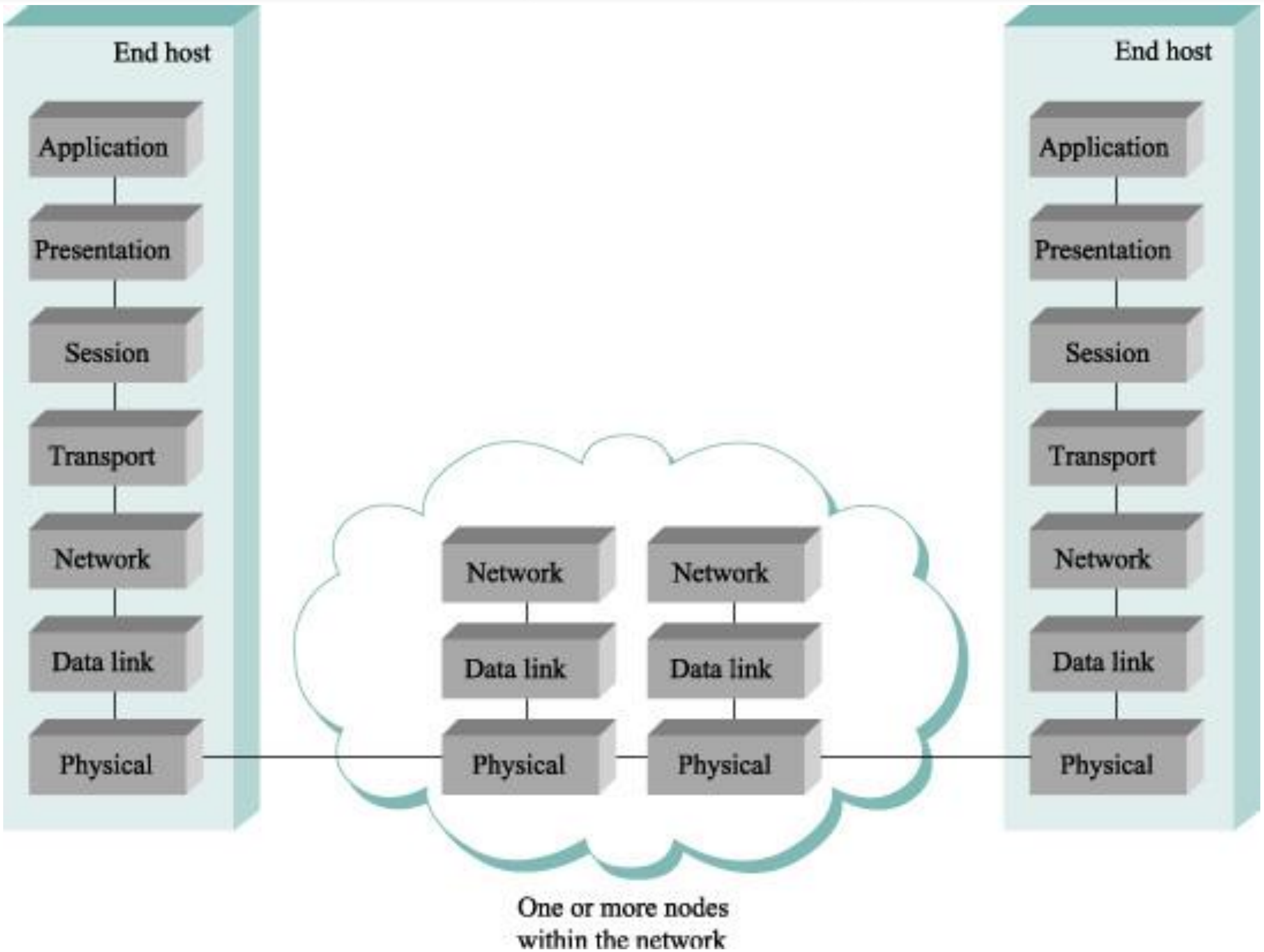
Network Architecture

OSI Architecture

OSI Architecture

- The ISO formally define a common way to connect computers
  - OSI: Open System Architecture
- OSI defines a partitioning of network functionality into seven layers
  - Where one or more protocols to implement the functionality assigned to a given layer
- ISO and ITU publishes a series of protocol specifications based on the OSI architecture
  - This series sometimes called “Xdot” series
  - X.25, X.400, X.500 and so on

OSI Network Architecture



OSI Network Architecture

- Physical layer handles the transmission of raw bits over communication link
- Data link layer then collects a stream of bits into a large aggregate called a frame
  - Network adaptors, along with device drivers running in the node's OS, typically implement the data link level
  - This means that frames, not raw bits, are actually delivered to hosts
- The network layer handles routing among nodes within a packet-switched network
  - At this layer, the unit of data exchanged among nodes is typically called a packet rather than a frame

OSI Network Architecture

- The lower three layers are implemented on all network nodes, including switches within the network and hosts connected along the exterior the network
- Transport layer implements a process-to-process channel
  - Here, the unit of data exchanged is commonly called a message rather than a packet or a frame
  - The transport layer and higher layers typically run only on the end hosts and not on the intermediate switches or routers

OSI Architecture

- Application Layer is the top (seventh) layer
  - Application layer protocols include things like the File Transfer Protocol (FTP)
  - Which defines a protocol by which file transfer application can interoperate
- Presentation Layer is concerned with the format of exchanged between peers
- Example:
  - Whether an integer is 16, 32, or 64 bits long and whether the most significant byte is *transmitted* first or last or how video stream is formatted

OSI Architecture

- Session Layer provides name space that is used to tie together the potentially different transport streams that are part of a single application
  - Example: It might manage an audio stream and a video stream that are being combined in a teleconferencing application

Network Architecture

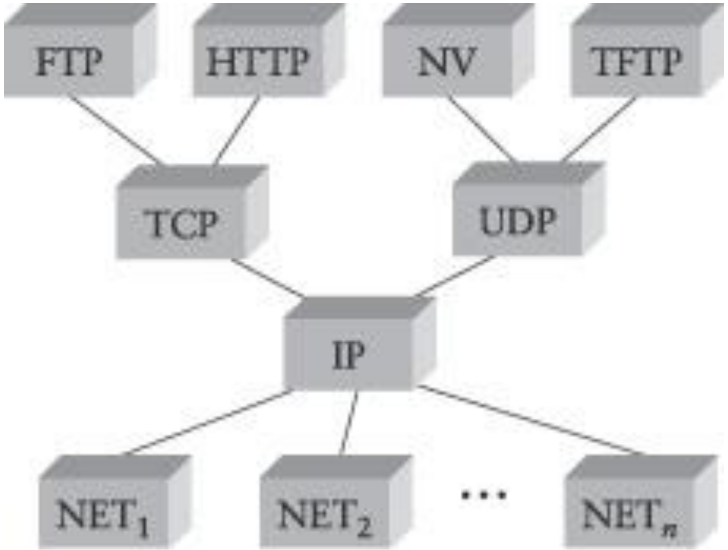
Internet Architecture

Internet Architecture

- Is also called as TCP/IP architecture
- Internet architecture evolved out of experiences with an earlier packet-switched network called the ARPANET
- Both the Internet and the ARPANET were funded by the Advanced Research Project Agency (ARPA) – US Defense
- The Internet and ARPANET were around before the OSI architecture

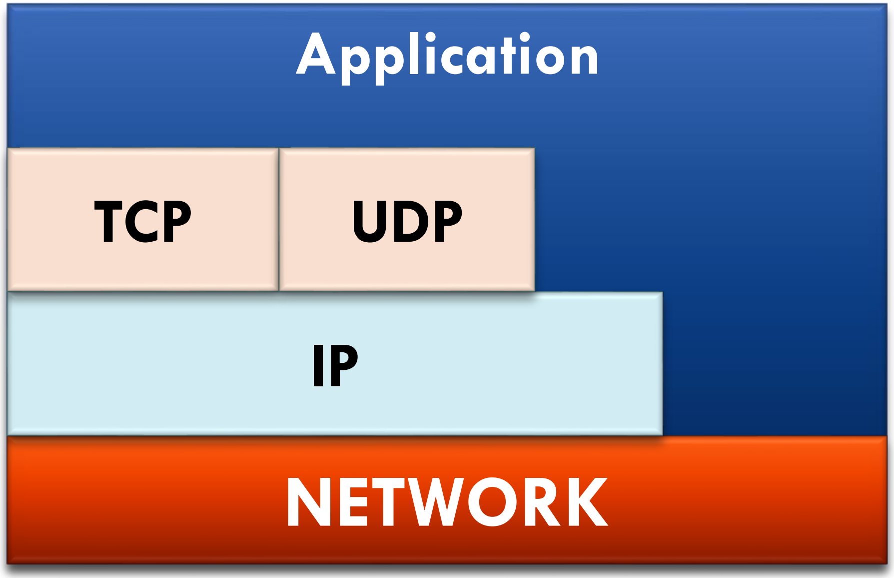
Internet Architecture

- A four layer model



Internet Protocol Graph

Internet Architecture



- An alternative view to the Internet architecture
- The "network" layer shown here is sometimes referred to as the "subnetwork" or "link" layer

Internet Architecture

- At the lowest level are a wide variety of network protocols, denoted NET_1 , NET_2 and so on
- In practice these protocols are implemented by a combination of hardware (e.g., network adaptor) and software (e.g., a network device driver)
- For example: Ethernet or Fiber Distributed Data Transfer (FDDI) protocols at this layer

Internet Architecture

- The second layer consists of a single protocol – Internet Protocol(IP)
  - This protocol supports the multiple networking technologies into a single logical internetwork
- The third layer contains two main protocols
  - The Transmission Control Protocol (TCP)
  - The User Datagram Protocol (UDP)
  - TCP and UDP provide alternative logical channels to application programs
  - TCP provides reliable byte-stream channel
  - UDP provides an unreliable datagram delivery channel
  - UDP and TCP protocols are also called as end-to-end protocols
  - We can refer this layer or protocol as Transfer protocol

Internet Architecture

- Above the transport layer/protocol is Application protocols
- Such as
  - FTP, TFTP (Trivial File Transport Protocol)
  - Telnet (remote login)
  - SMTP (Simple Mail Transfer Protocol, or electronic mail)
  - HTTP (Hyper Text Transfer Protocol)

Internet Architecture Features

- Three features
- The internet layering does not imply strict layering
  - The application is free to bypass the defined transport layers and to directly use IP or one of the underlying networks
  - Programmers are free to define new channel abstractions or applications that run on top of any of the existing protocols

Internet Architecture features

- Protocol graph looks like a hourglass shape
  - Wide at the top, narrow in the middle, and wide at the bottom
  - This shape actually reflects the central philosophy of the architecture
    - i.e., IP serves as the focal point for the architecture
    - It defines a common method for exchanging packets among a wide collection networks
  - Above IP can be arbitrarily many transport protocols, each offering different channel abstraction to application programs
  - Below IP, the architecture allows for arbitrarily many different network technologies ranging from Ethernet to wireless to single point-to-point links

Internet Architecture features

- A new protocol is officially (IETF) included in the architecture
  - There needs to be both a protocol specification and at least one (and preferably two) representative implementations of the specification
  - This IETF cultural assumption of the design community helps to ensure that the architecture's protocols can be efficiently implemented

Reference

- ***Larry L Peterson & B S Davie***, **Computer Networks: A Systems Approach**, 4 Edn, Morgan Kauffman Publishers, 2007

THANK YOU!!

6 August 2025

Noor Mahammad Sk