計算機網路概論
(Computer Networks)

張正尚

台達館 932 室

Tel: 5742579

E-mail: cschang@ee.nthu.edu.tw

課程內容與要求

- 本課程主要在介紹目前的計算機網路，包括**有線網路**及**無線網路**。介紹相關的**網路架構**，**網路通訊協定**，**網路介接**，**網路流量控制**，及**相關的應用**。學生可以藉由此課程更加了解目前的計算機網路。
- 教科書：Computer Networks: A System Approach, 5th Ed.
  – by Larry L. Peterson and Bruce S. Davie
  – Morgan Kaufmann Publishers, 2011
- 講義位置：

課程內容與要求

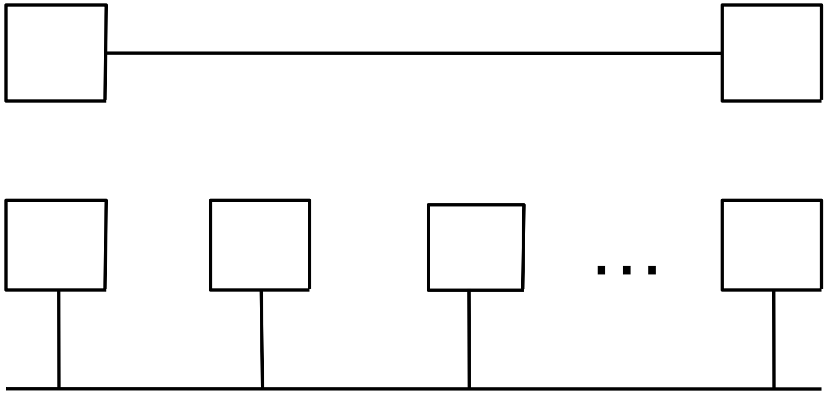
- 參考書目
  - Computer Networks by Andrew S. Tanenbaum, 4th Ed., Prentice-Hall PTR, 2003.
  - An Introduction to Computer Networking by Kenneth C. Mansfield Jr. and James L. Antonakos, Prentice-Hall PTR, 2002.
  - Computer Communications and Networking Technologies by Michael A. Gallo and William M. Hancock, Thomson Learning Inc., 2002.

課程內容與要求

- 課程內容
  - Direct-Link Network
  - Packet Switching
  - Internetworking
  - End-to-End Protocols
  - Congestion Control and Resource Allocation
  - End-to-End Data
  - Network Security
  - Applications
- 課程要求
  - **Midterm Exam (25 %)**
  - **Final Exam (30 %)**
  - **Quiz (15 %)**
  - **Homework (25 %)**
  - **Class participation (5%)**

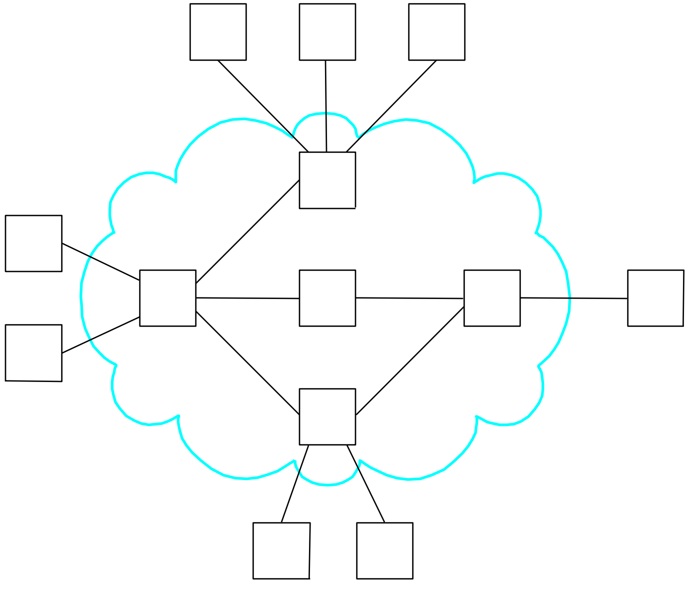
Building Blocks

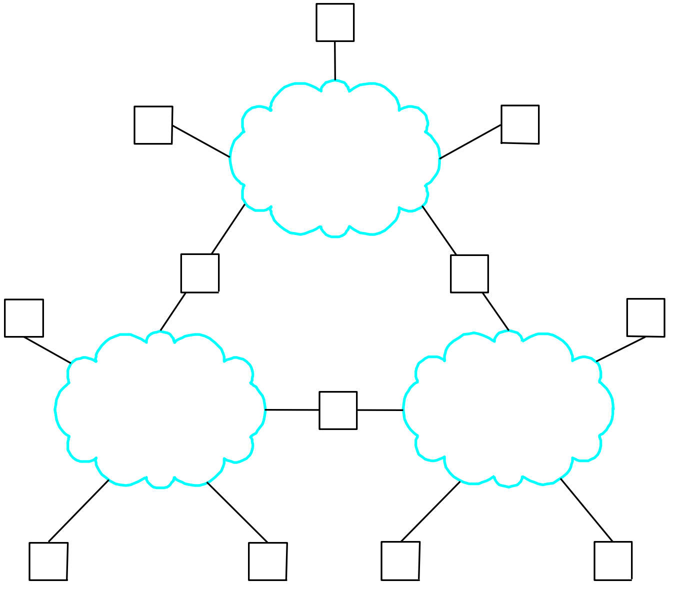
- Nodes: PC, special-purpose hardware…
  - hosts
  - switches
- Links: coax cable, optical fiber…
  - point-to-point
  - multiple access



Switched Networks

- A network can be defined recursively as...
  - two or more nodes connected by a link, or
  - two or more networks connected by two or more nodes





Strategies

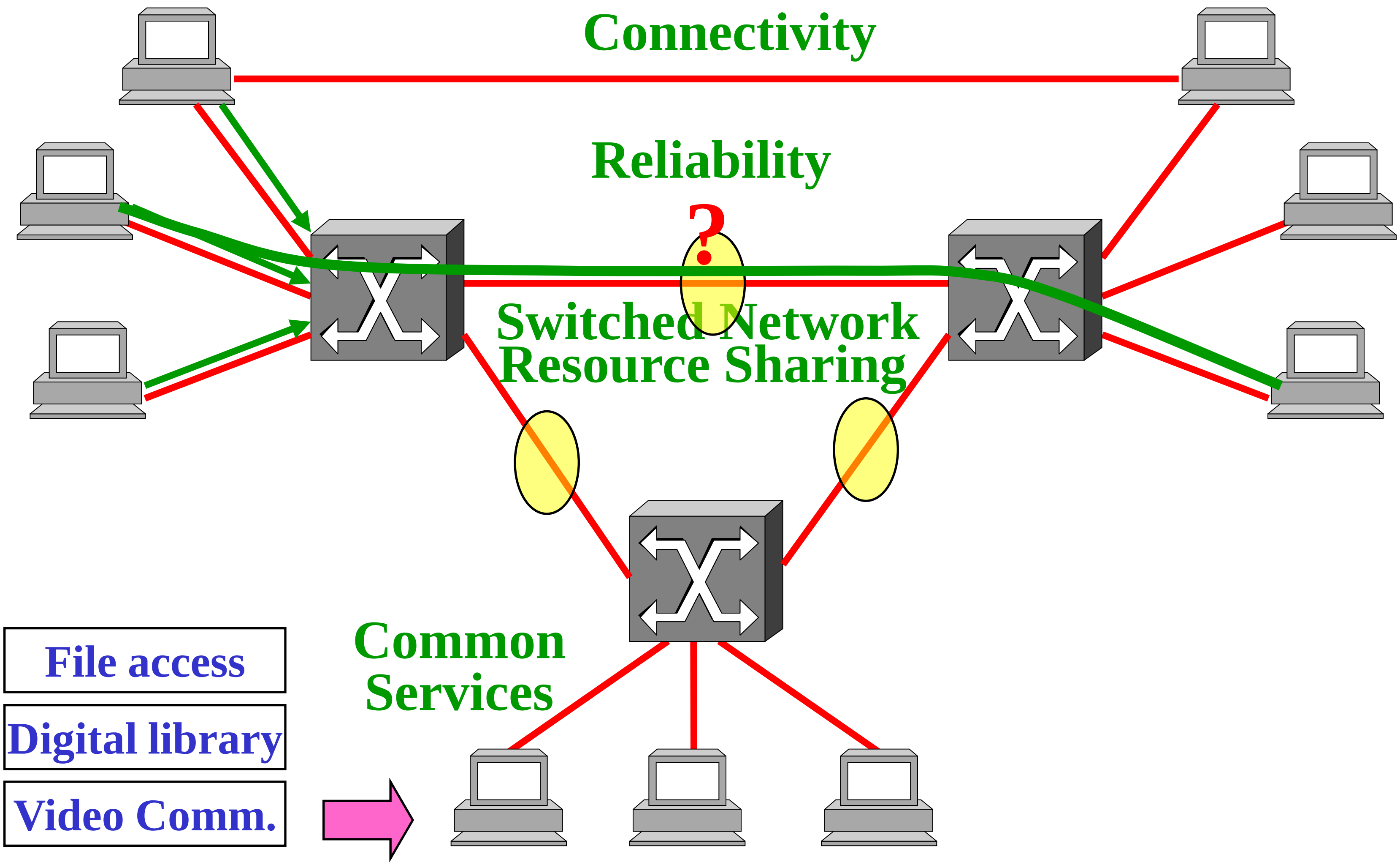
- Circuit switching: carry bit streams
  - original telephone network
- Packet switching: store-and-forward messages
  - Internet

Addressing and Routing

- Address: byte-string that identifies a node
  - usually unique
- Routing: process of forwarding messages to the destination node based on its address
- Types of addresses
  - unicast: node-specific
  - broadcast: all nodes on the network
  - multicast: some subset of nodes on the network

Requirements

Requirements



Connectivity

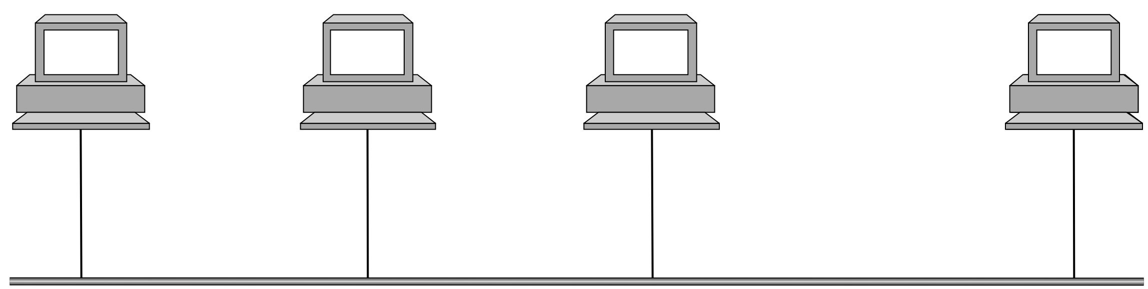
- A network must provide connectivity among a set of computers.
- **Private networks:**
  - Limit the set of machines that are connected
  - For reasons of **privacy** and **security**
  - Prevent virus infection and hacker attacks
- **Public networks** (Internet):
  - Allows them the potential to connect **all the computers** in the world

Connectivity

- Network connectivity occurs at many different levels
- A network can consist of two or more computers directly connected by some **physical medium**
  - Such as a coaxial cable or an optical fiber.
- **Link:** such a physical medium
- **Node:** the computers it connects

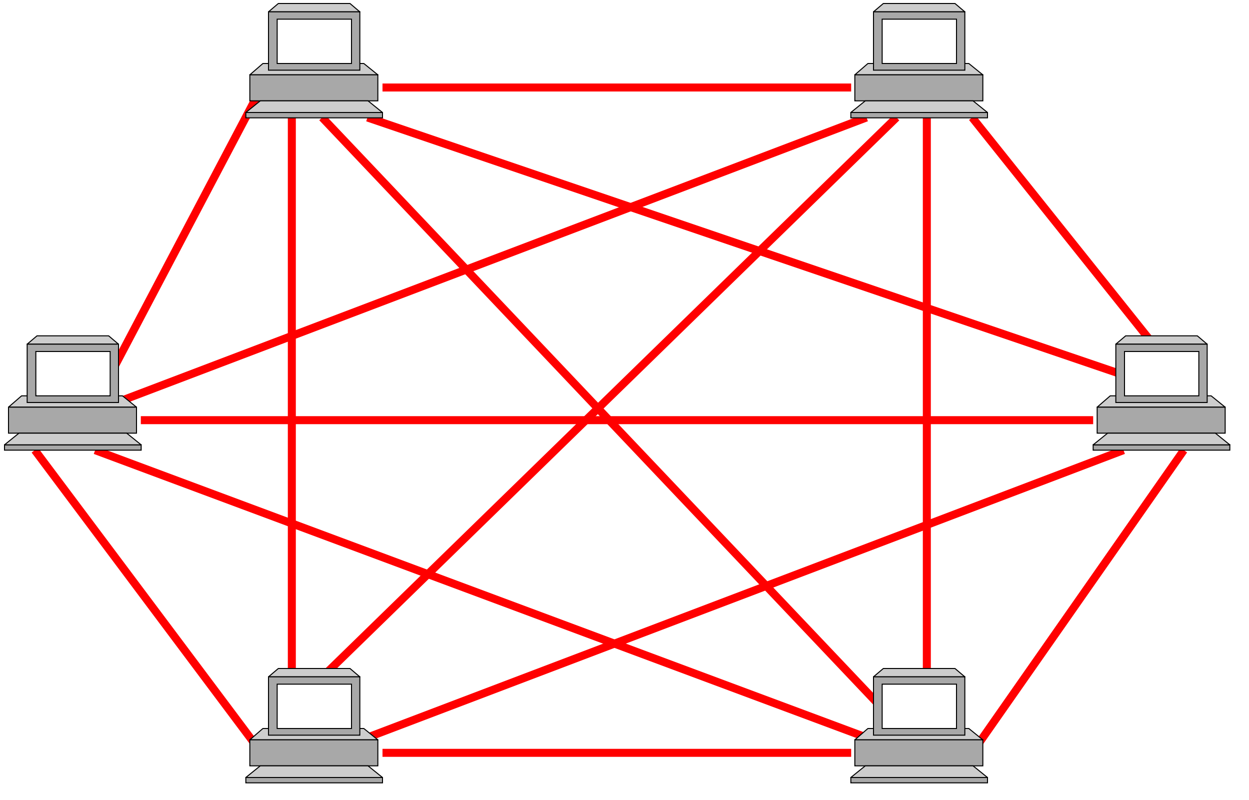
Point-to-point

Multiple-access



Connectivity

- If all nodes are **directly connected to each other** over a common physical medium



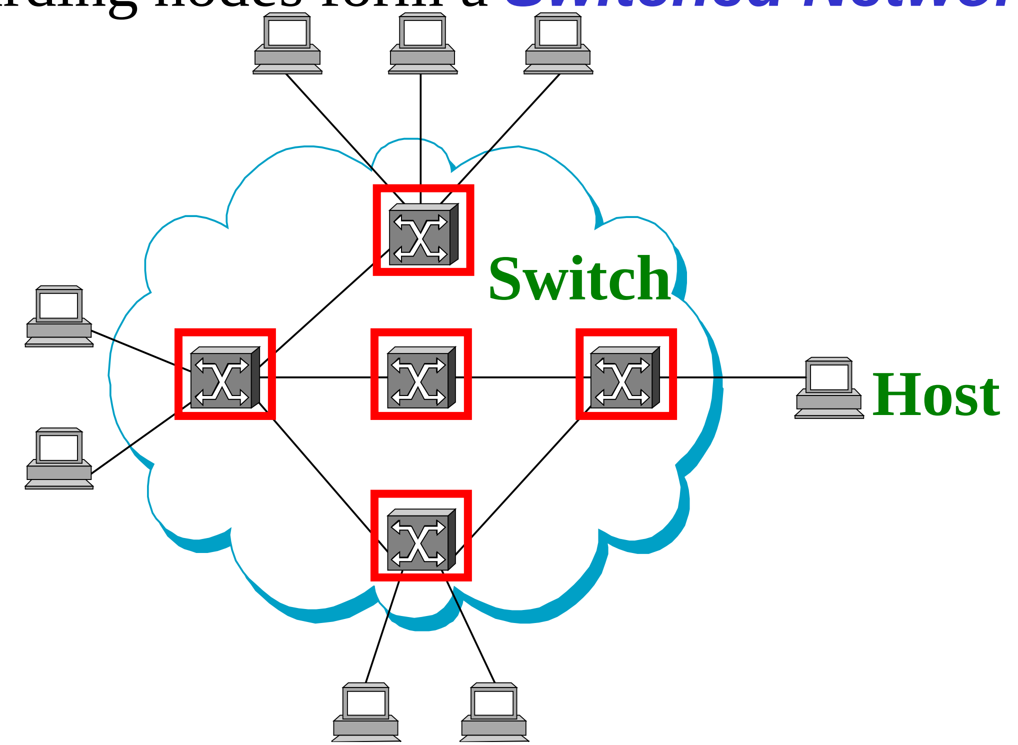
Connectivity

- A **directly connected** network
  - Networks would be very **limited** in the number of computers they could connect
  - The number of wires of each node would quickly become both **unmanageable** and very **expensive**
  - **Number of wires: $N \times (N - 1)/2$**
- **Indirect connectivity** should be achieved among a set of cooperating nodes
  - **Switched network**

Switched Network

Switched Network

- Each node is attached to **one or more** point-to-point links
- **Switch:** those nodes that are attached to at least two links
  - Forwards data received on one link out on another
- These forwarding nodes form a ***Switched Network***

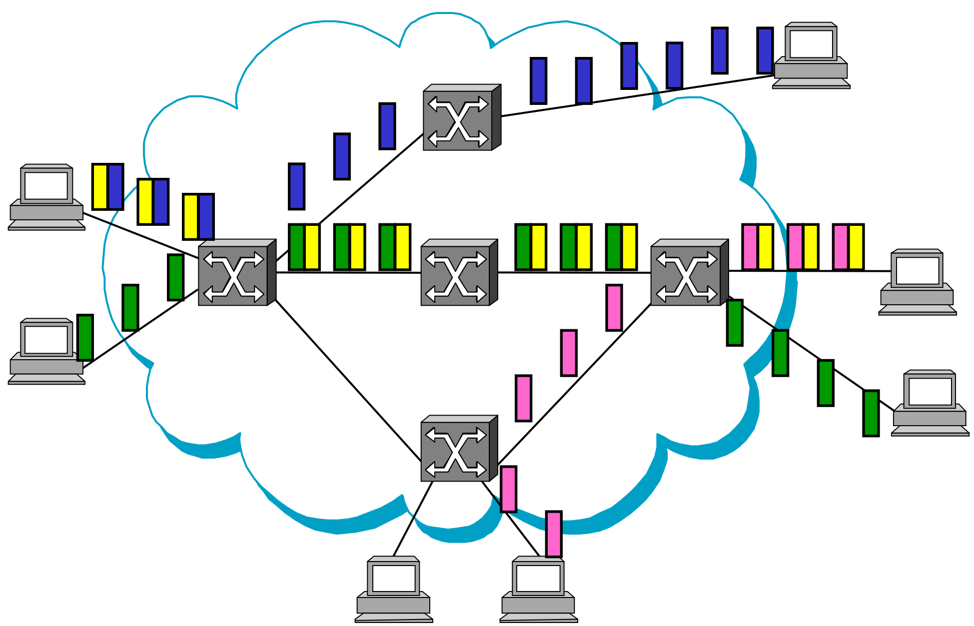


Types of Switched Network

- Two most common types of switched networks:
  - **Circuit-switched:** most notably employed by the telephony system
  - **Packet-switched:** used for the overwhelming majority of computer networks
- **Packet:**
  - A block of data
  - Corresponding to some piece of application data
    - such as a file, a piece of email, or an image
- The major reason for using packet switching
  - **Efficiency**

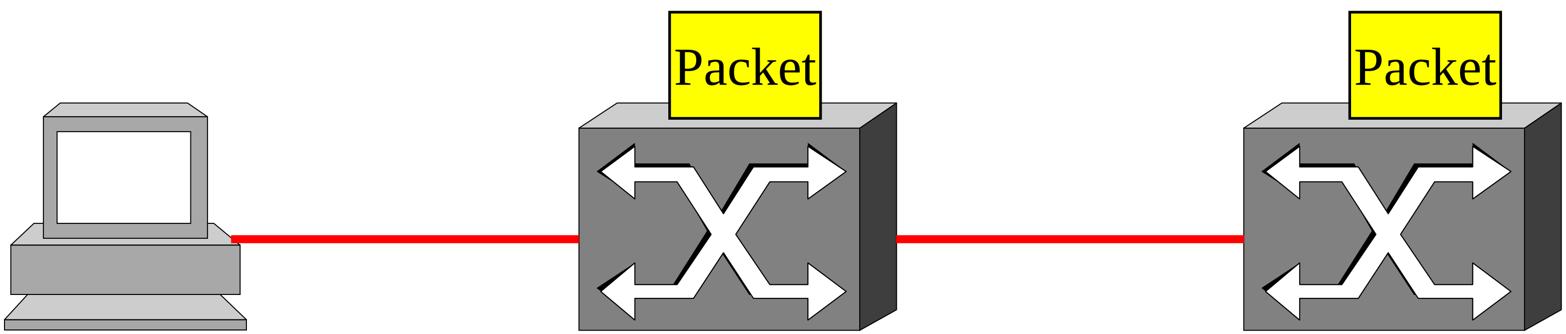
Packet Switched Network

- **Packet-switched:**
  - Send discrete blocks of data to each other
  - **Store-and-forward**



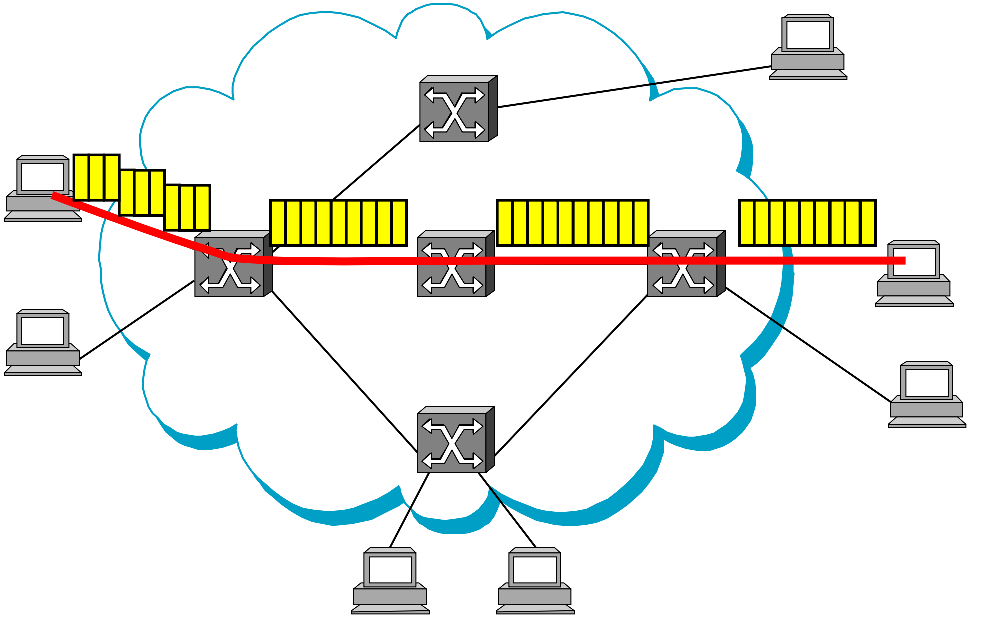
Packet Switched Network

- **Packet-switched:**
  - Store-and-forward
  - Each node receives a complete packet
  - **Store** the packet in internal memory
  - **Forward** the complete packet to the next node



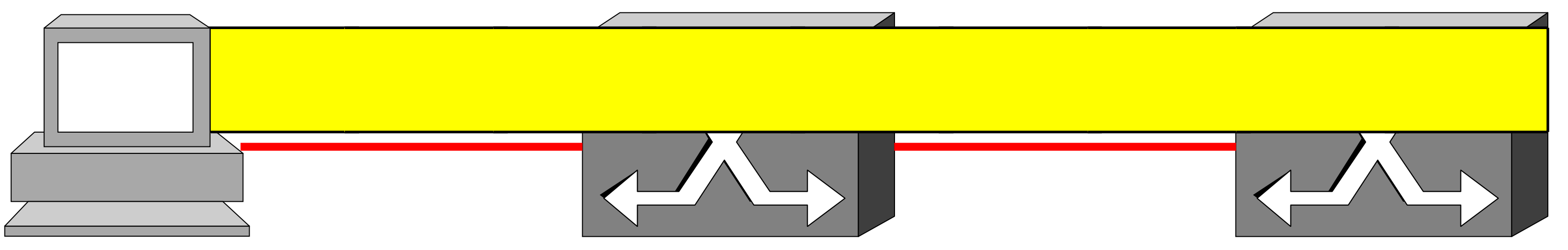
Circuit Switched Network

- **Circuit-switched:**
  - Establishes a dedicated circuit across a sequence of links
  - Allows the source node to send **a stream of bits** across this circuit to a destination node.



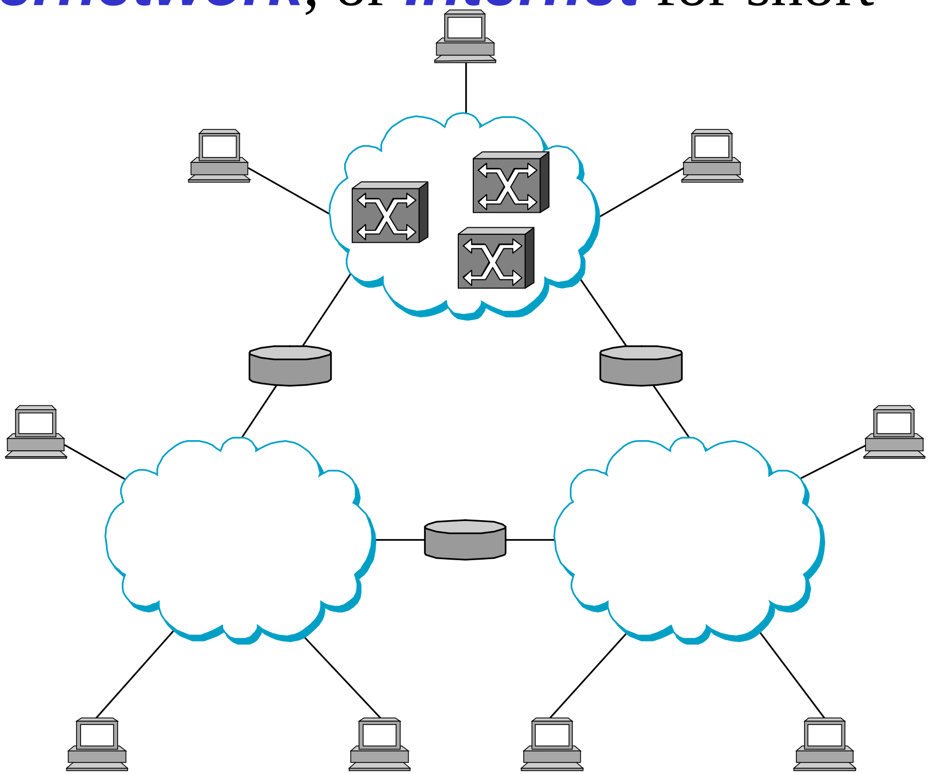
Circuit Switched Network

- **Circuit-switched:**
  - Continuous transmission
  - A stream of bits across this circuit to a destination node.



Internetwork

- A second way in which a set of computers can be **indirectly connected**
- A set of independent networks (clouds) are interconnected to form an ***Internetwork***, or ***Internet*** for short



Internetwork

- A node that is connected to **two or more networks** is commonly called a ***router*** or ***gateway***
  - Forwards messages from one network to another
- Each node must be able to say **which of the other nodes** on the network it wants to communicate with
  - Assign an ***address*** to each node
  - An address is a byte string that identifies a node
- If the sending and receiving nodes are **not directly connected**
  - The switches and routers of the network use this address to decide **how to forward the message** toward the destination
  - The process is called ***routing***

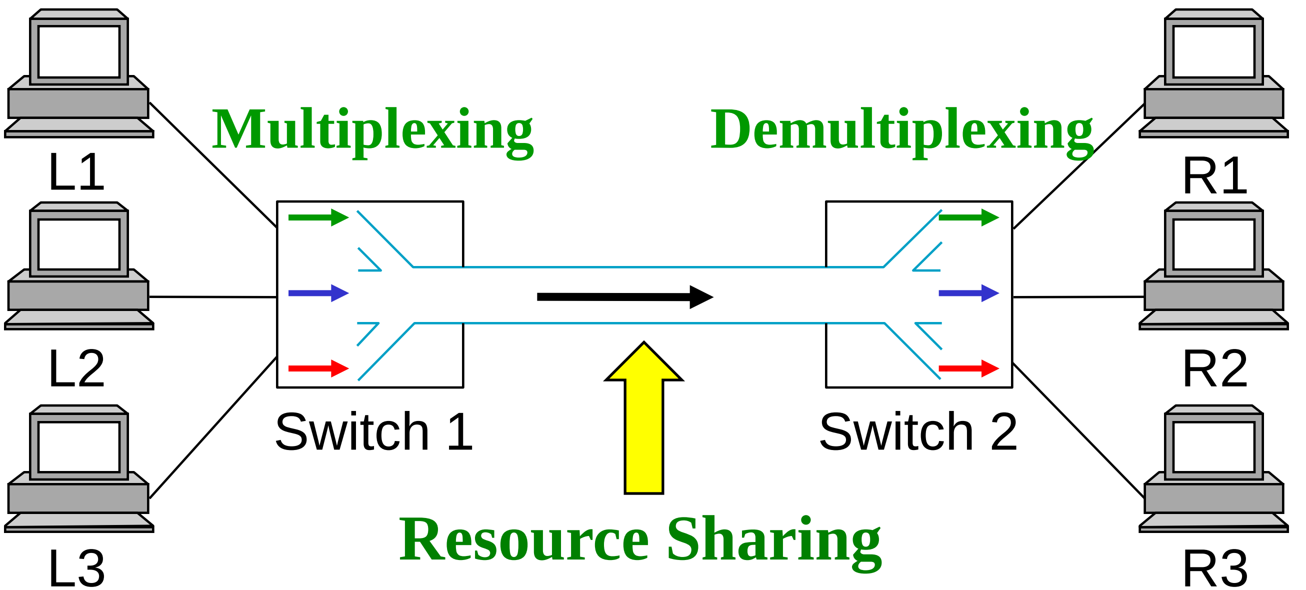
Switched Network

- **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:** The source node might want to send a message to **some subset of the other nodes**, but not all of them
- Thus, in addition to node-specific addresses, another requirement of a network is
  - **Supporting multicast and broadcast addresses**

Resource Sharing

Cost-Effective Resource Sharing

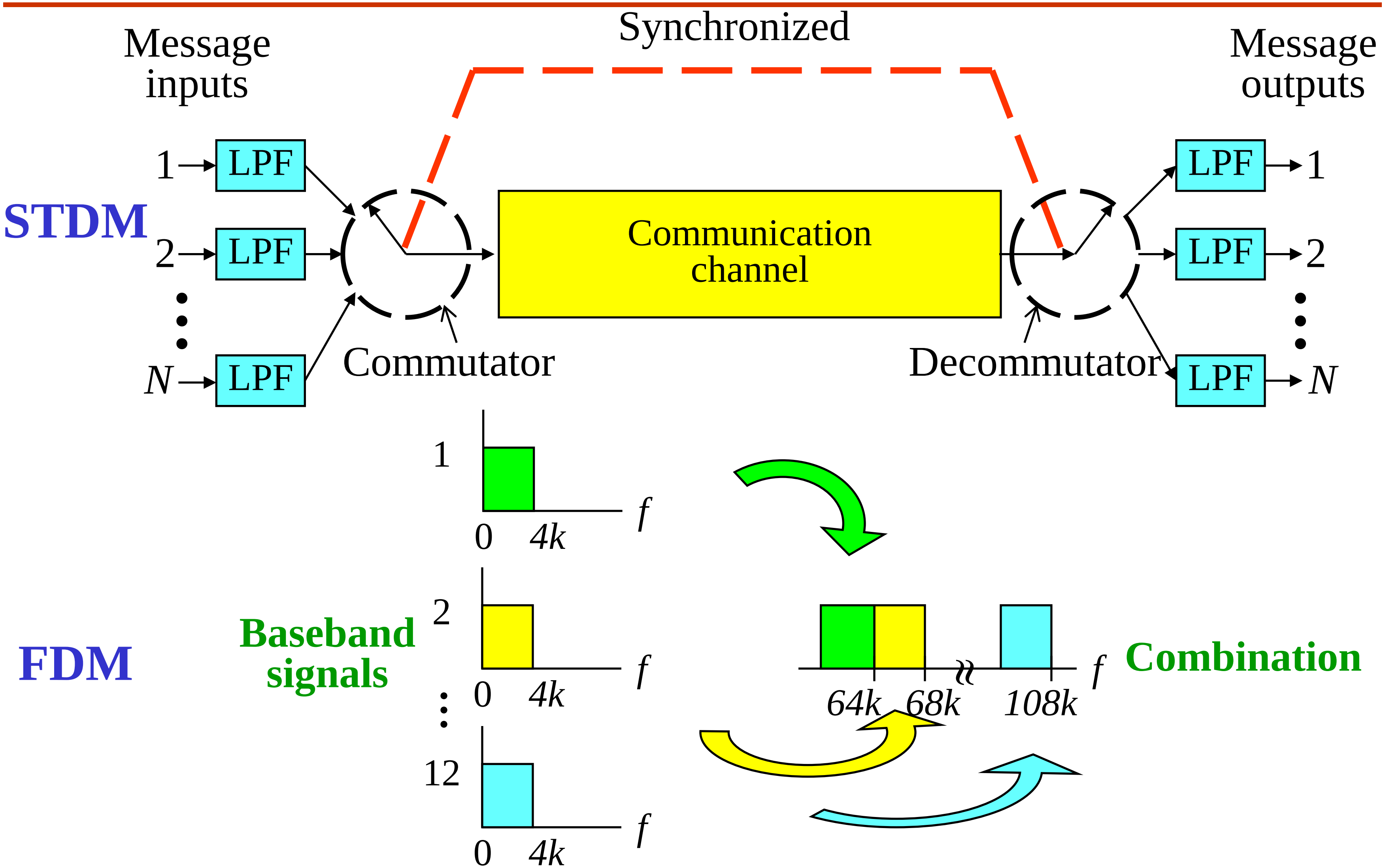
- A collection of nodes indirectly connected by a nesting of networks
  - Provide all pairs of hosts with the ability to exchange messages
- **Multiplexing:** a system resource is shared among multiple users



Multiplexing

- There are several different methods for multiplexing multiple flows onto one physical link
  - **Synchronous time-division multiplexing (STDM)**
  - **Frequency-division multiplexing (FDM)**
- **STDM:**
  - 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
- **FDM:**
  - Transmit each flow over the physical link at **a different frequency band**

STDM & FDM



Synchronized
Message inputs
Message outputs
STDM
1
2
N
LPF
Communication channel
Commutator
Decommutator
FDM
Baseband signals
1
2
12
0
4k
f
64k
68k
108k
Combination

Multiplexing

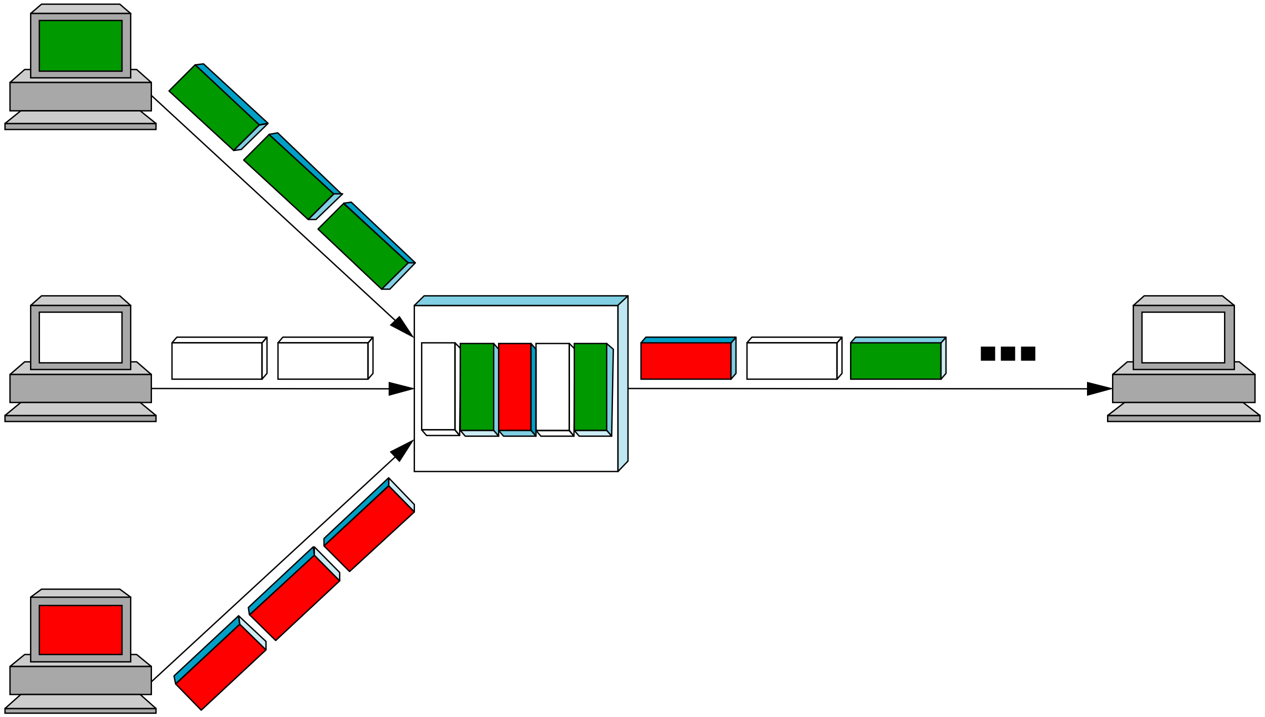
- Both STDM and FDM are limited in two ways:
  - **Efficiency:** If one of the flows (host pairs) **does not have any data to send**, its share of the physical link—that is, its time quantum or its frequency—**remains idle**
    - For computer communication, the amount of time that a link is idle can be **very large**
  - **Feasibility:** 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

- **Statistical multiplexing:** with two key ideas
  - The physical link is shared over time like STDM. However, data is transmitted from each flow **on demand** rather than during a predetermined time slot
    - The **avoidance of idle time** ⇒ **more efficient**
  - Defines 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 referred to as a **packet**
    - Ensure that all the flows eventually get their turn to transmit over the physical link ⇒ **fairness**
- The source may need to **fragment** the message into packets
- The receiver **reassembles** the packets back into the message

Statistical Multiplexing

- Each flow sends a sequence of packets over the physical link
  - With a decision made on a **packet-by-packet basis**



Statistical Multiplexing

- There are number of different ways to decide which packet to be sent next on a shared link:
  - To service packets on a **first-in-first-out (FIFO)** basis
  - To service the different flows in a **round-robin** manner, just as in STDM
- One of the issues that faces a network designer is how to make this decision **in a fair manner**
- A network that allows flows to request some treatment is said to support **quality of service (QoS)**
  - To ensure that certain flows receive a particular share of the **link's bandwidth**
  - To ensure that they never have their packets **delayed** in the switch for more than a certain length of time

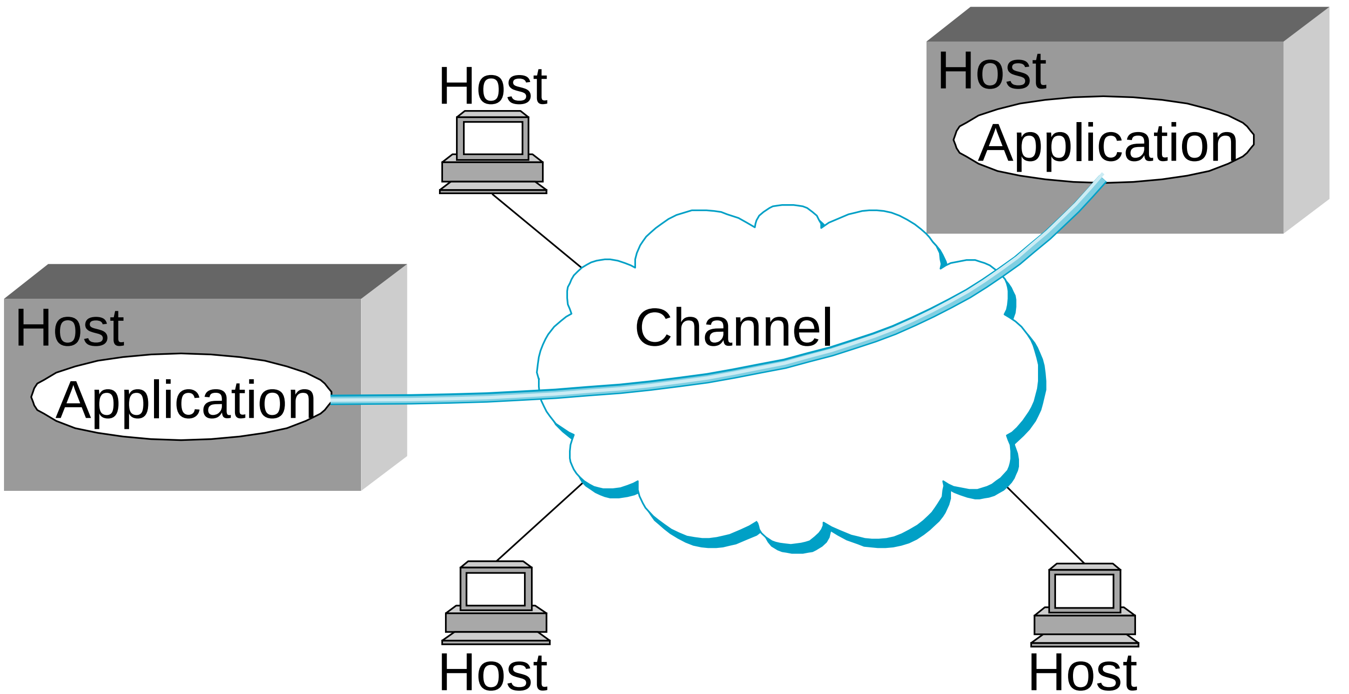
Common Services

Support for Common Services

- When two application programs need to communicate with each other
  - Simply sending a message from one host to another $\Rightarrow$ A lot of **complicated things** need to happen
- Since many applications need common services
  - Implement those common services **once**
  - Let the application designer build the application **using those services**
- Intuitively, we view the network as providing **logical channels** over which application-level processes can communicate with each other

Support for Common Services

- Each channel provides the set of services required by that application
- A network provides a variety of **different types of channels**
  - Each application selecting the type that **best meets its needs**



Examples

- Use **file access**, a **digital library**, and the **video applications** (videoconferencing and video-on-demand) as a sample
- Assume that two types of logical channels are provided:
  - **Request/reply channels** & **Message stream channels**
- The request/reply channel would be used by the **file transfer** and **digital library applications**
  - Every message sent by one side is received by the other side
  - **Only one copy** of each message is delivered
- The message stream channel could be used by both the **video-on-demand and videoconferencing applications**
  - Does not need to guarantee that all messages are delivered
  - Need to ensure that those messages that are delivered arrive **in the same order** in which they were sent

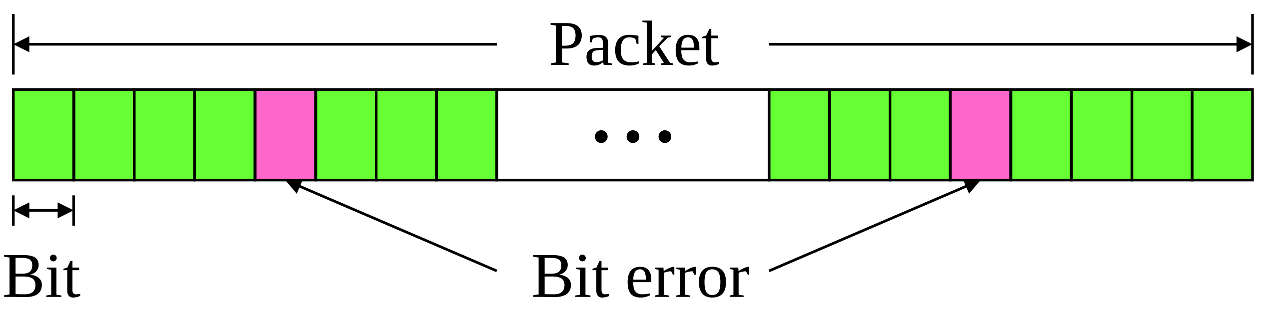
Reliability

Reliability

- **Reliable message delivery** is one of the most important functions that a network can provide
- Computer networks do not exist in a perfect world:
  - A network should mask (hide) certain kinds of failures
  - Make the network **appear more reliable** than it really is
- There are three general classes of failure:
  - **First, bit errors may be introduced into the data**
  - **The second class of failure is that a complete packet is lost by the network**
  - **The third class of failure is that a physical link is cut, or that the connected computer has crashed**

Reliability

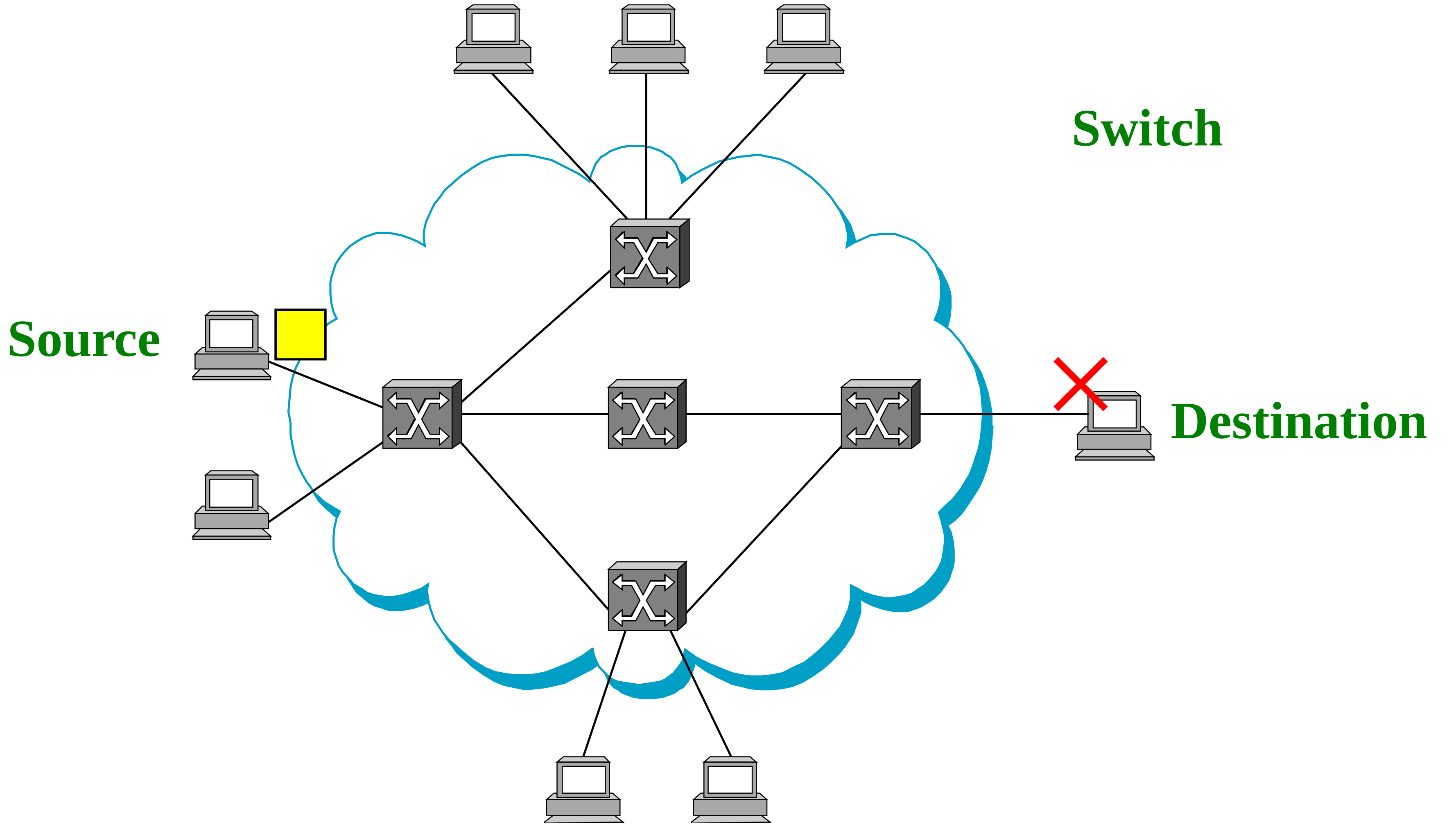
- **First, bit errors may be introduced into the data**
  - Bit errors typically occur because outside forces (**noise**)
  - Such bit errors are **fairly rare**:
    - A typical copper-based cable: BER $10^{-6} \sim 10^{-7}$
    - A typical optical fiber: BER $10^{-12} \sim 10^{-14}$
  - There are techniques that **detect** these bit errors
  - It is sometimes possible to **correct** for such errors



Reliability

- **The second class of failure is that a complete packet is lost by the network**
  - One reason is that the packet contains an **uncorrectable bit error** and therefore has to be discarded
  - A more likely reason is that **one of the nodes is forced to drop the packet** (out of capacity)
  - Less commonly, the **software** running on one of the nodes that handles the packet **makes a mistake**
    - For example, it might **incorrectly forward** a packet out on the wrong link
  - The sender may be expected to **retransmit** the packet
  - The main difficulty: to distinguish that a packet is **indeed lost** or is merely **late in arriving** at the destination

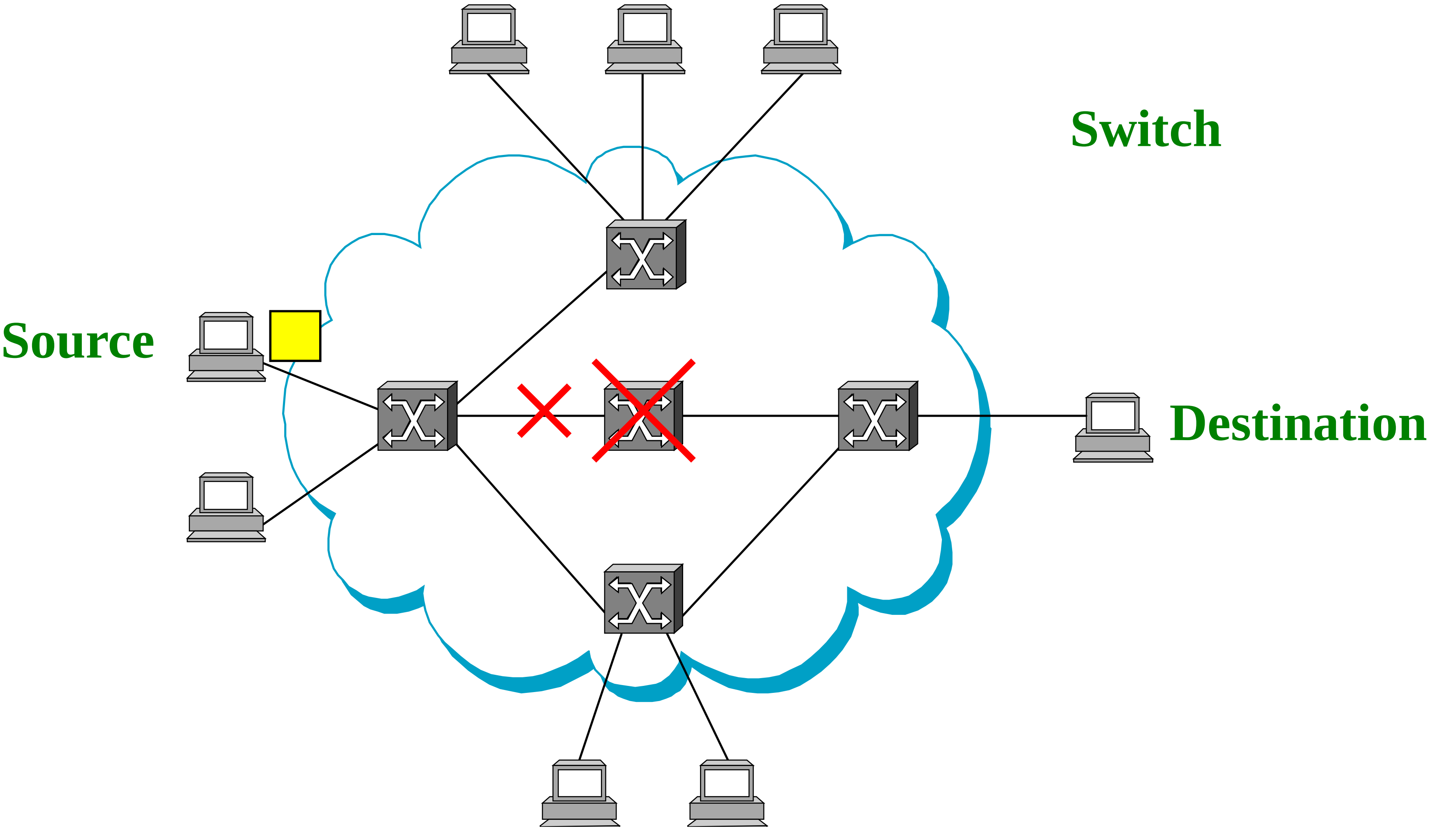
Reliability



Reliability

- **The third class of failure is that a physical link is cut, or that the connected computer has crashed**
  - Caused by software that crashes, a power failure, …
  - Such failures can have a **dramatic effect** on the network for an extended period of time
  - They need not totally disable the network: it is sometimes possible to **route around a failed node or link**
  - It is difficult to distinguish between a **failed computer** and one that is **merely slow**
  - It is difficult to distinguish between a link **has been cut** and one that is with **a high number of bit errors**

Reliability



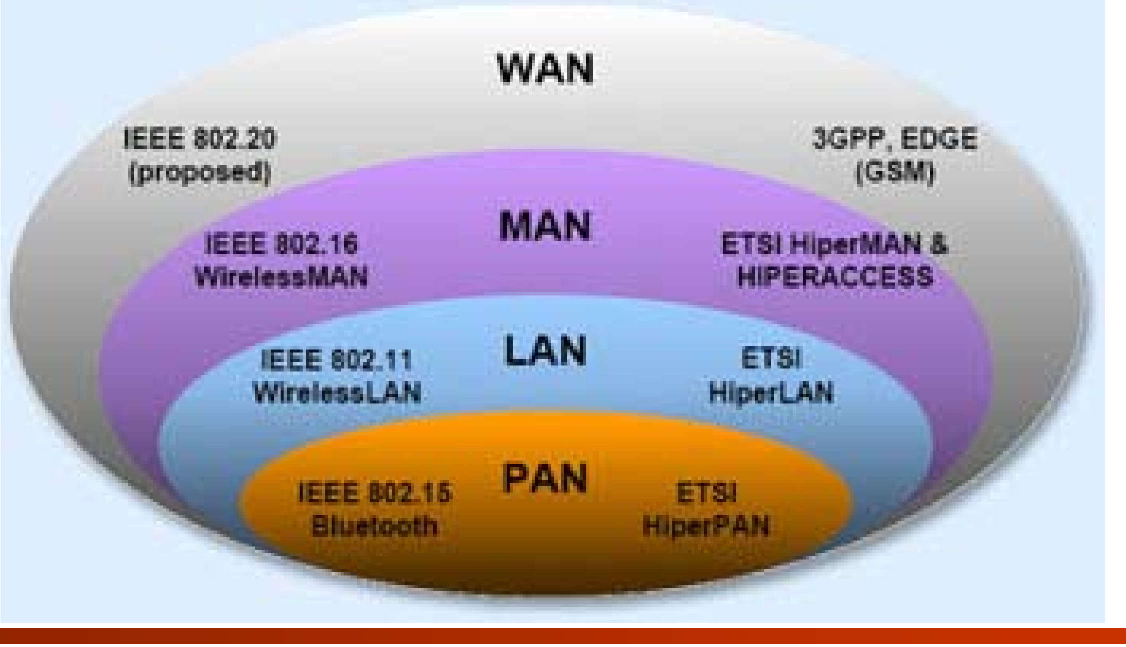
Network Architecture

SANs, LANs, MANs, and WANs

- One way to characterize networks is according to their size:
  - **LANs (local area networks):** extend less than 1 km
  - **MANs (metropolitan area networks):** span tens of kilometers
  - **WANs (wide area networks):** can be worldwide
  - **SANs (system area networks):** are usually confined to a **single room** and connect the various components of a large computing system
    - HiPPI (High Performance Parallel Interface) and Fiber Channel are two common SAN technologies

SANs, LANs, MANs, and WANs

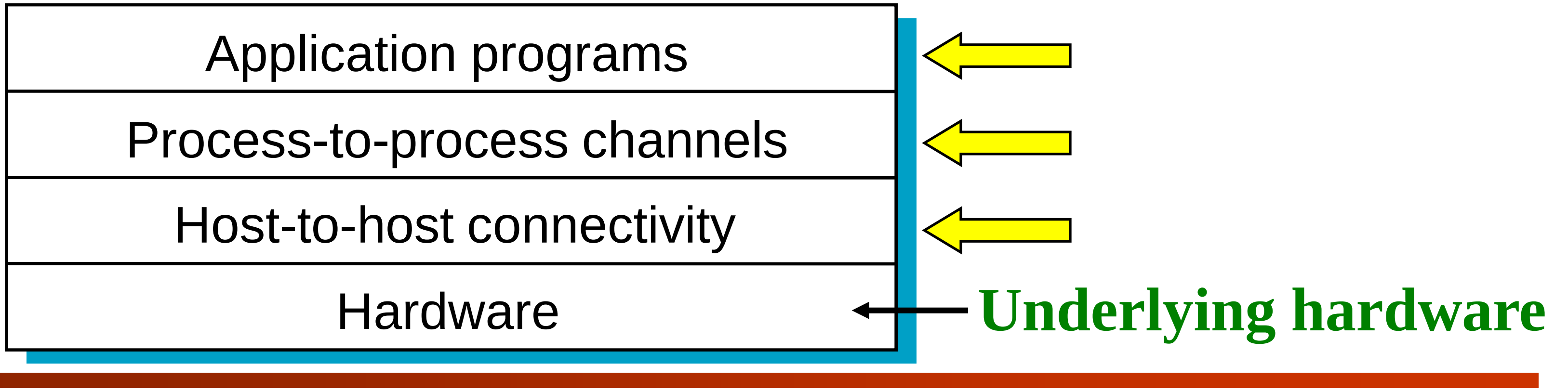
- A new class of network classification:
  - **PANs (Personal area networks):** networks that are meant for one person
    - extend to about 10 meters
  - **UWB** (Ultra Wide Band), **Bluetooth**



Layering and Protocols

Layering

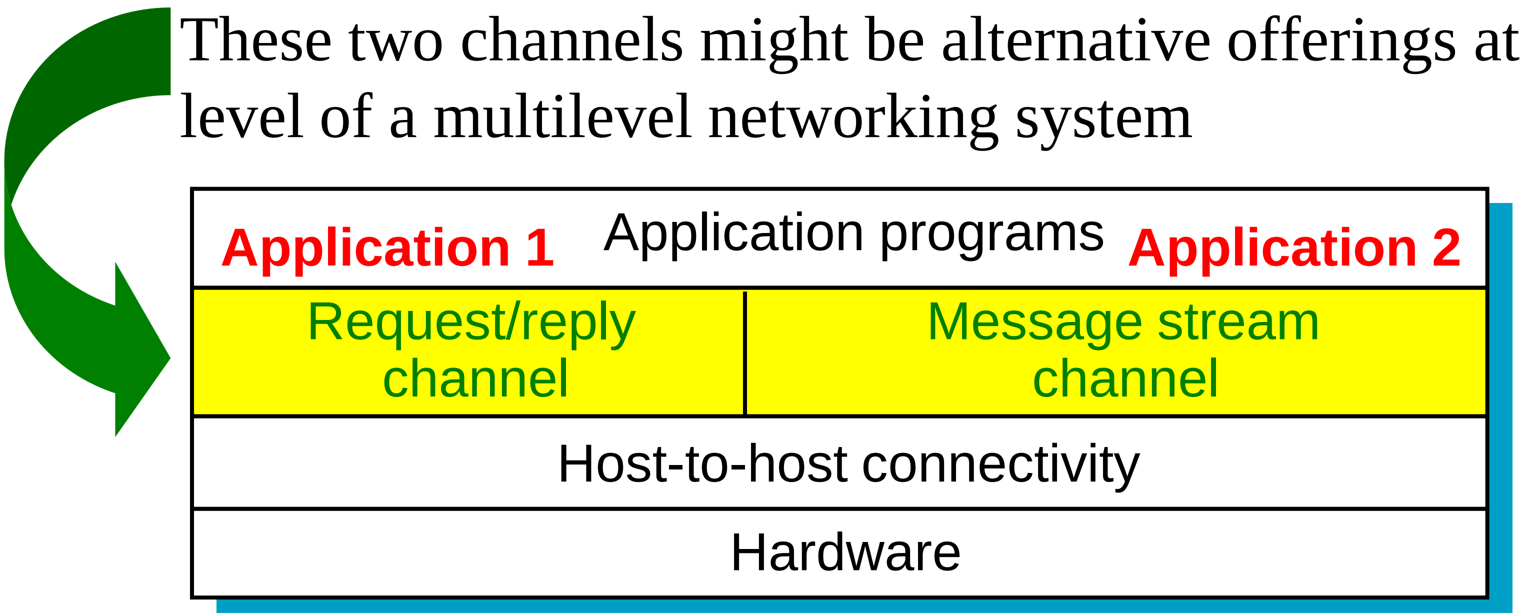
- **Layering:** Start with the services offered by the underlying hardware, and then add a sequence of layers
  - The services provided at the high layers are implemented in terms of **the services provided by the low layers**
- Layering provides two nice features
  - First, it decomposes the problem of building a network into more **manageable** components (hide complexity)
  - Second, it provides a more **modular design**



Layering

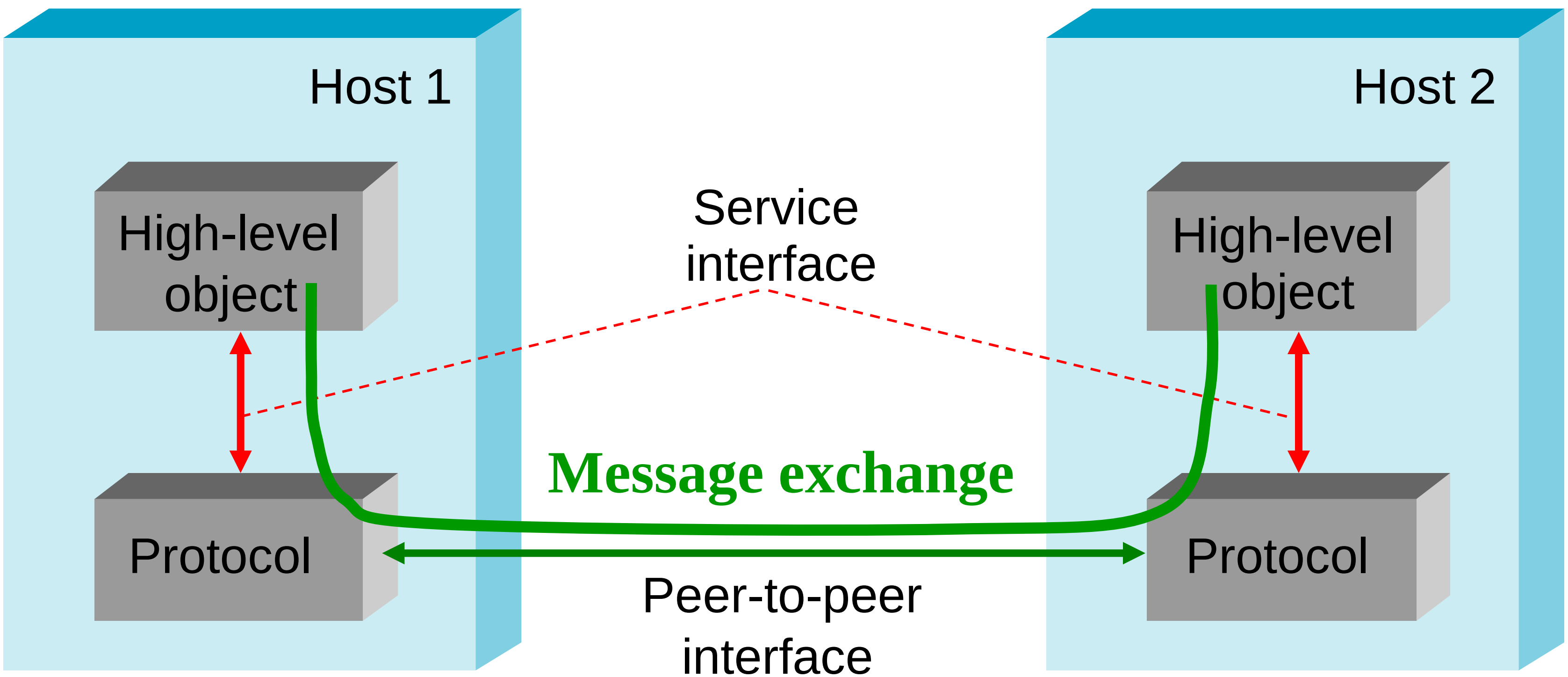
- To add some new service
  - To modify the functionality at **one layer**, reusing the functions provided at all the other layers
- Multiple abstractions may be provided at any given level
  - Each providing a **different service** to the higher layers

These two channels might be alternative offerings at some level of a multilevel networking system



Protocols

- **Protocols:** the abstract objects that make up the layers of a network system
  - A protocol provides a communication service that **higher-level objects** use to exchange messages



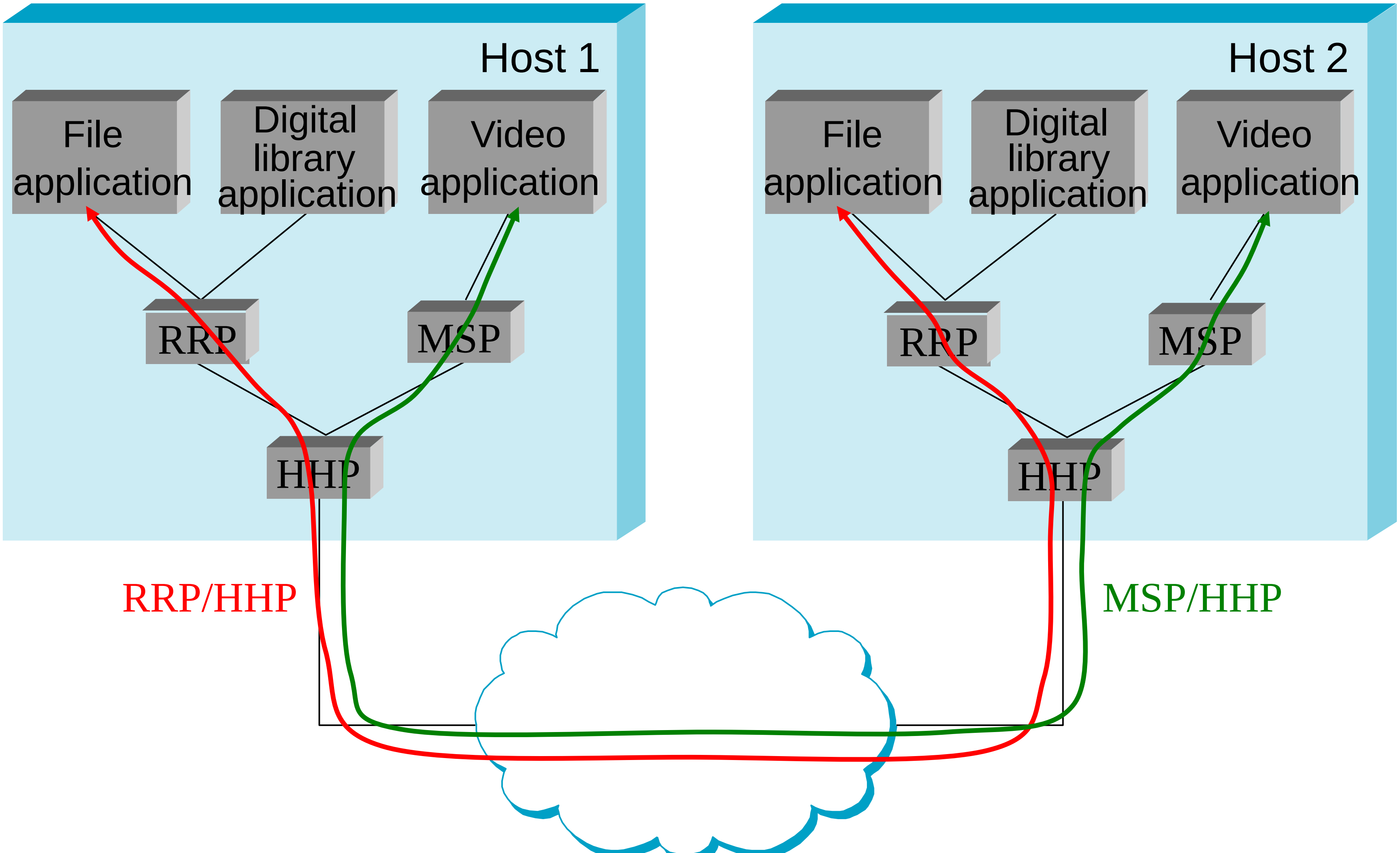
Protocols

- Each protocol defines two different interfaces
  - First, it defines **a service interface** to the other objects (program) **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
  - Second, a protocol defines **a peer interface** to its counterpart (peer) **on another machine**
    - This second interface defines the form and meaning of messages exchanged between protocol peers

Multiple Protocols in a Layer

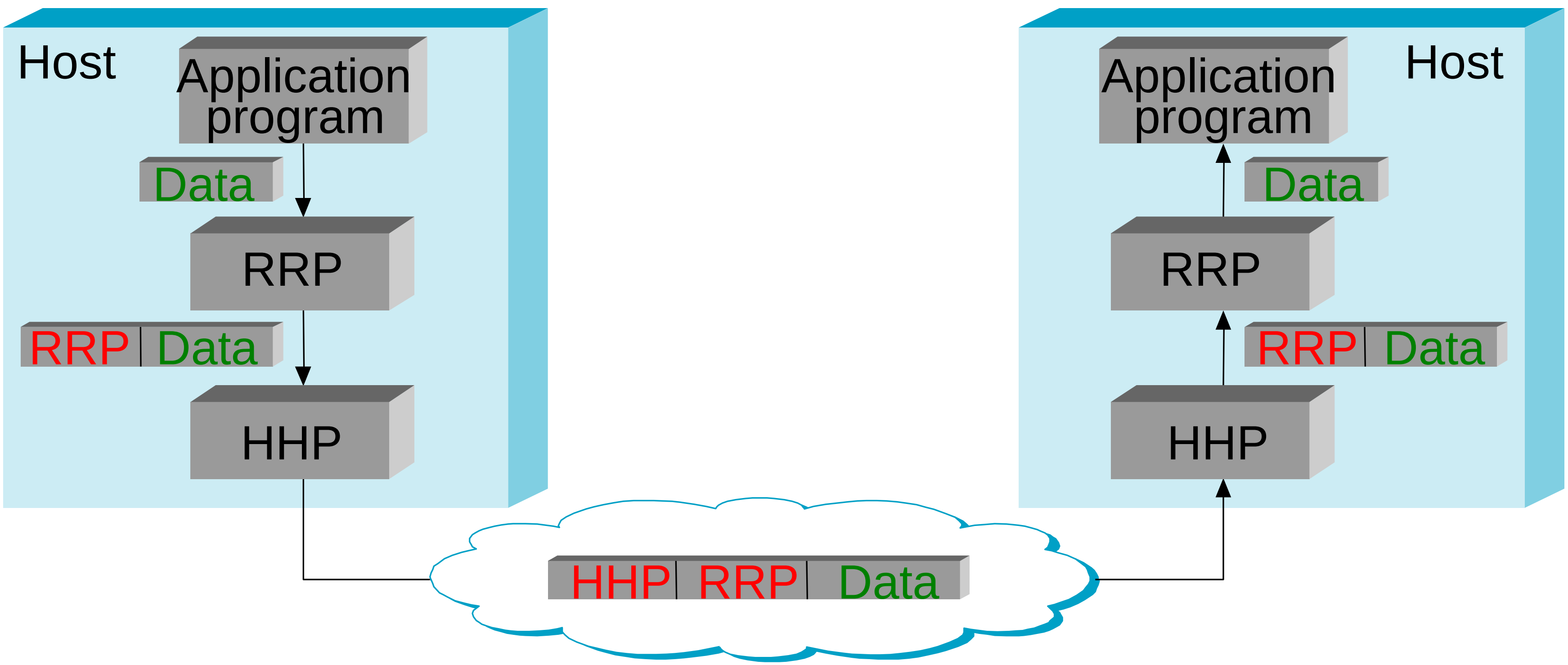
- Except the hardware level **(lowest level)**, peer-to-peer communication is **indirect**
  - Each protocol communicates with its peer by **passing messages to some lower-level protocol**
- In addition, there are potentially multiple protocols at any given level, each providing a different communication service
- Two different types of process-to-process channels, depending on HHP **(Host-to-Host Protocol)**:
  - **RRP (Request/Reply Protocol)**
  - **MSP (Message Stream Protocol)**
- The application employs the services of the protocol stack **RRP/HHP** or **MSP/HHP**

Multiple Protocols in a Layer



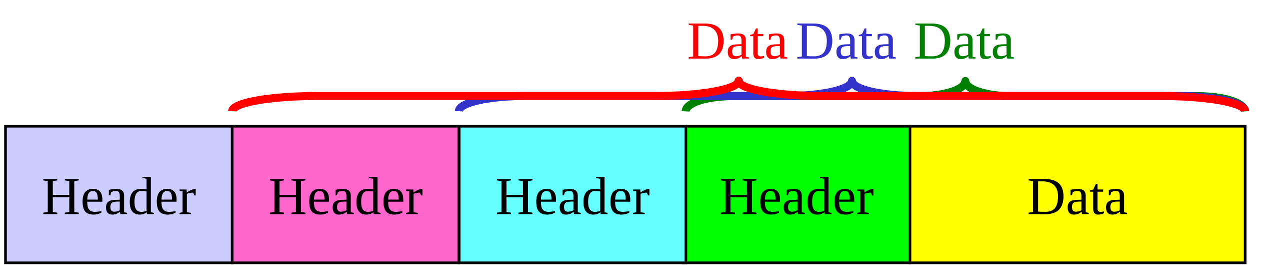
Encapsulating

- RRP must communicate control information to its peer
  - By attaching a **header** to the message
- The rest of the message is called **message body** or **payload**
  - The application data is **encapsulated** in the RRP message



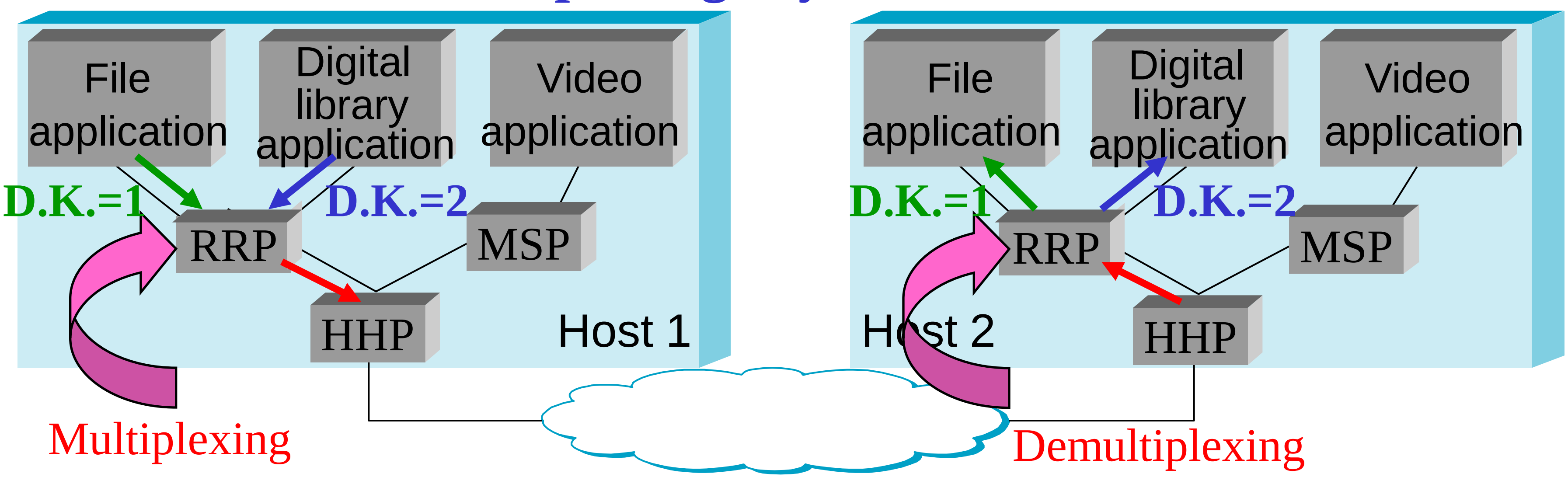
Encapsulating

- This process of encapsulation is repeated at **each level** of the protocol graph
- When the message arrives at the destination host, it is processed in the **opposite order**
  - Strips its header off the front of the message
  - Interprets **the header** (i.e., takes whatever action is appropriate given the contents of the header
  - Passes the body of the message up to RRP



Multiplexing and Demultiplexing

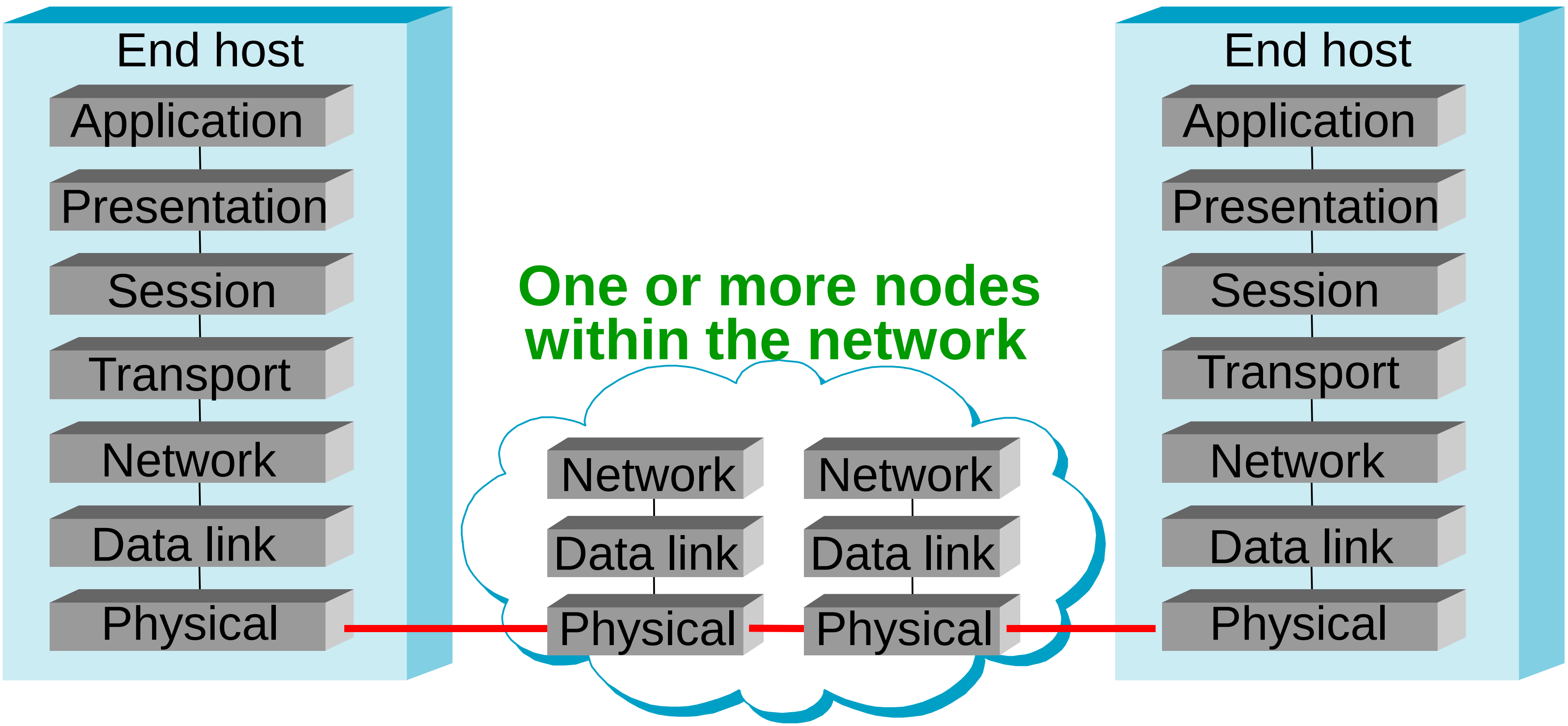
- RRP can be thought as a **logical** communication channel
  - Two different applications **multiplexed** and **demultiplexed** over this channel
- The RRP header contains an identifier that records the application to which the message belongs
  - RRP's **demultiplexing key**



OSI Architecture

OSI Architecture

- ISO (International Organization for Standardization): one of the first organizations to formally define a common way to connect
  - **Open Systems Interconnection (OSI)** architecture



OSI Architecture

- **Physical** layer: handles the transmission of raw bits over a link
- **Data link** layer: collects a stream of bits into a larger aggregate called a **frame**
  - **Network adaptors**, along with **device drivers** running in the node's OS, typically implement the data link level
  - Frames, not raw bits, are actually delivered to hosts
- **Network** layer: handles **routing** among nodes within a packet-switched network
  - The unit of data exchanged among nodes is called a **packet**
- The lower three layers are implemented on **all network nodes**
  - Including switches within the network and end hosts

OSI Architecture

- Other higher layers typically run **only on the end hosts** and not on the intermediate switches or routers
- **Transport** layer: implements a **process-to-process channel**
  - The unit of data exchanged is commonly called a **message**
- There is **less agreement** about the definition of the top three layers
- **Application** layer: **user interference**, e.g. the File Transfer Protocol (FTP), which defines a protocol by which file transfer applications can inter-operate
- **Presentation** layer: concerns with the **data format** exchanged between peers
  - For example: whether an integer is 16, 32, or 64 bits long

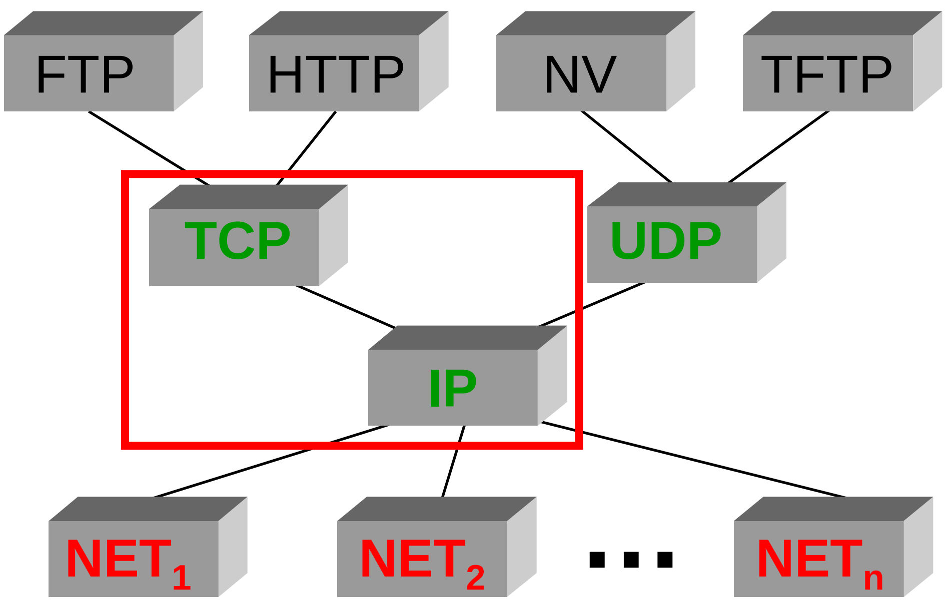
OSI Architecture

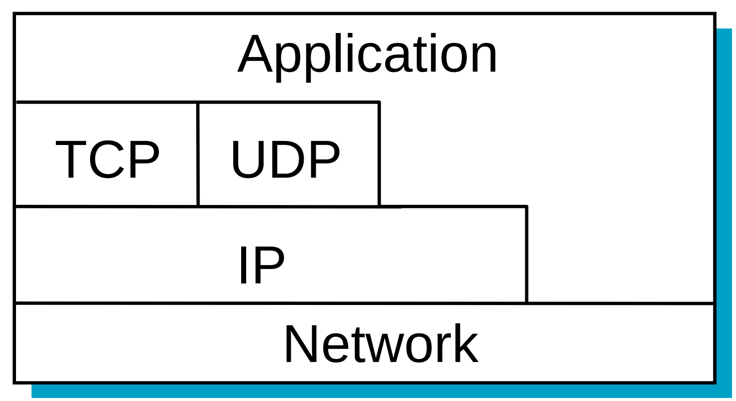
  - Whether the most significant bit is transmitted first or last
  - How a video stream is formatted
  - …
- **Session** layer: provides a name space that is used to tie together the potentially **different transport streams** that are part of **a single application**
  - It might manage an **audio stream** and a **video stream** that are being combined in a **teleconferencing application**

Internet Architecture

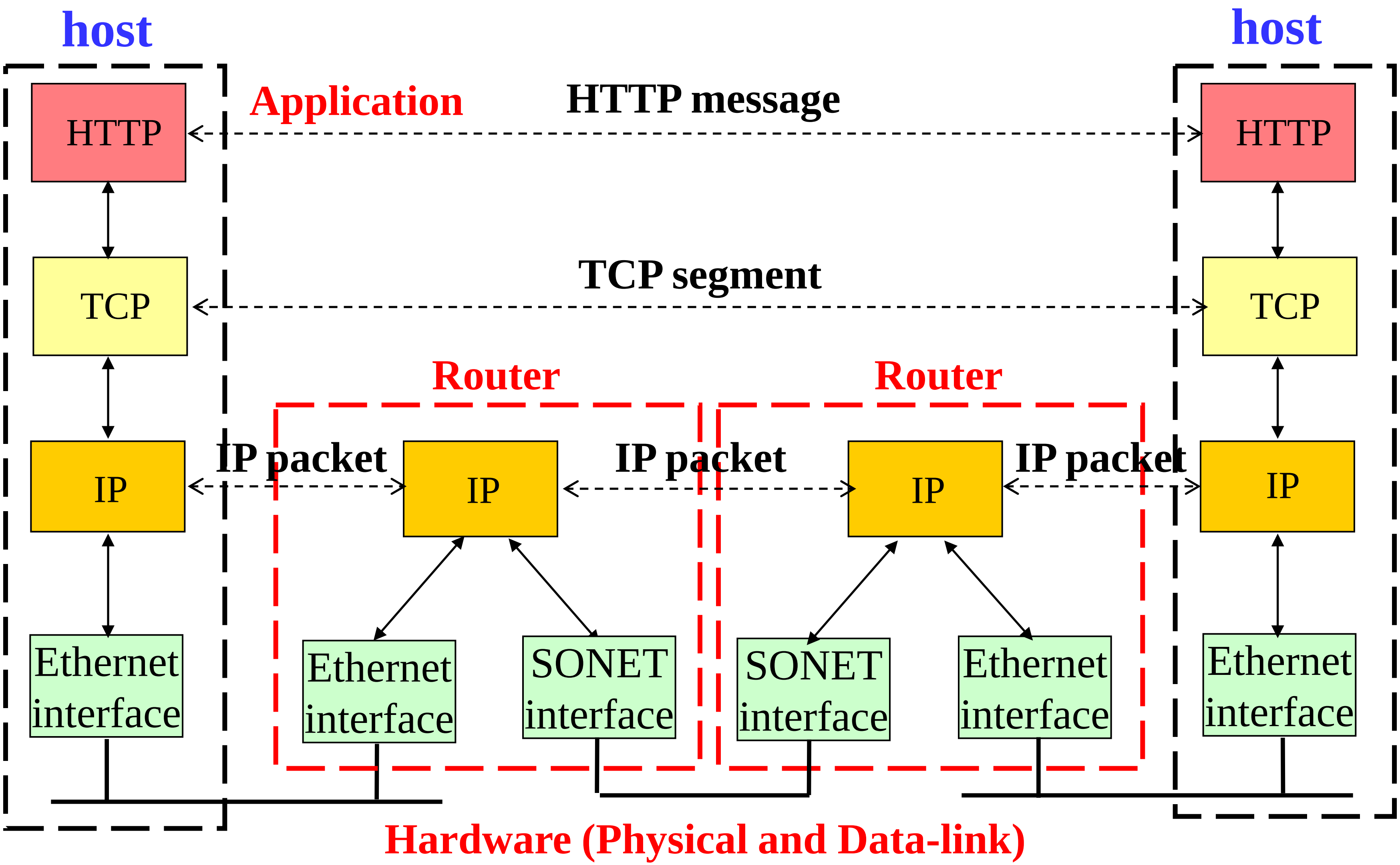
Internet Architecture

- The Internet architecture is also sometimes called the **TCP/IP architecture**
- At the **lowest layer** are a **wide variety** of network protocols
  - Implemented by a combination of hardware (e.g., a network adaptor) and software (e.g., a network device driver)





Internet Architecture



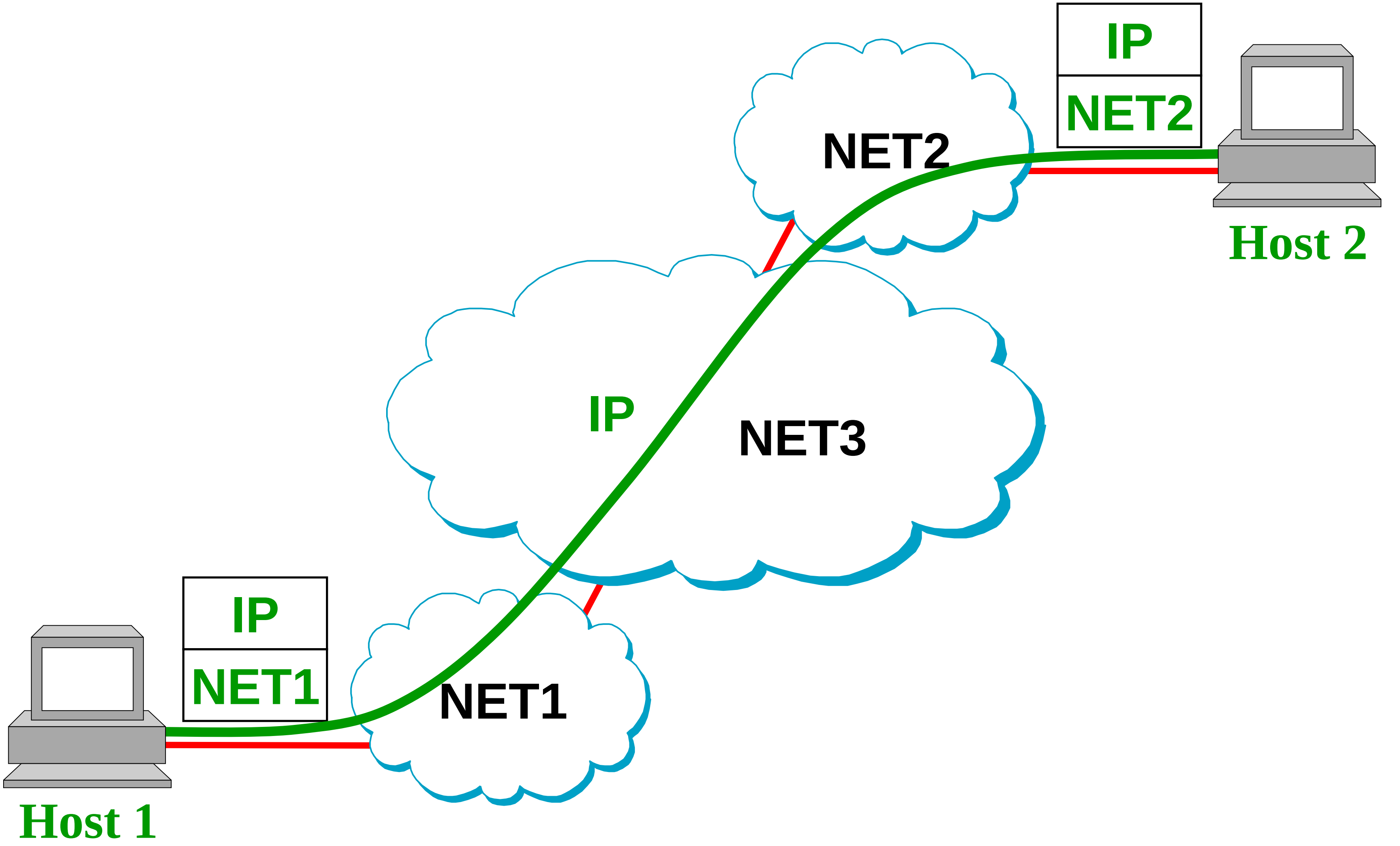
Internet Architecture

- The second layer consists of a single protocol—the **Internet Protocol (IP)**
  - Supports the interconnection of multiple networking technologies into a single, logical internetwork
- The third layer contains two main protocols
  - **Transmission Control Protocol (TCP):** provides a **reliable** byte-stream channel
  - **User Datagram Protocol (UDP):** provides an **unreliable** datagram delivery channel
  - TCP and UDP are sometimes called **end-to-end protocols**
- Above the transport layer are a range of **application protocols**
  - FTP, TFTP (Trivial File Transport Protocol), Telnet (remote login), and SMTP (Simple Mail Transfer Protocol)

Internet Architecture

- Applications vs application layer protocols
  - **Applications: WWW browsers** (Netscape, Explorer) are all based on the same **application layer protocol: HTTP**
- The Internet architecture does not imply **strict layering**
  - The application may **bypass the transport layers** and **directly use IP** or one of the underlying networks
- IP defines a common method for **exchanging packets** among a wide collection of networks
  - Delivering messages from **host to host** is **completely separated from** the **process-to-process** level
  - Below IP, **different network technologies** are allowed

Internet Architecture

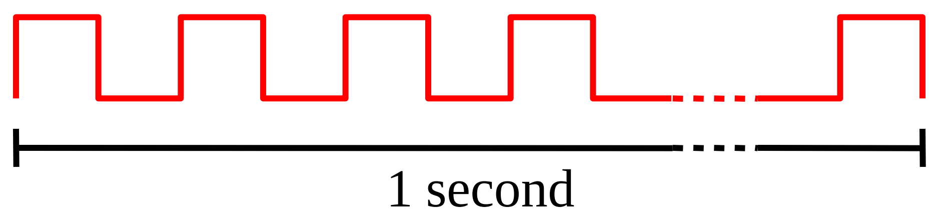


Performance

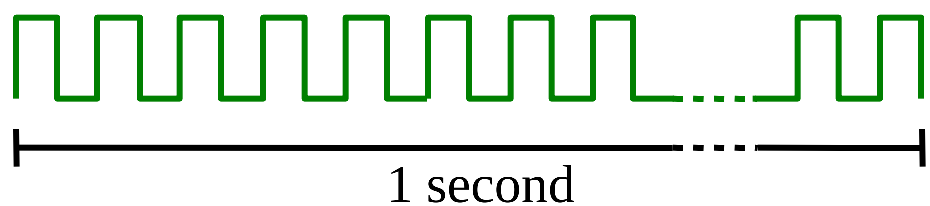
Bandwidth and Latency

- Network performance is measured in two fundamental ways:
  - **Bandwidth** (also called throughput)
  - **Latency** (also called delay)
- The bandwidth of a network: the number of bits that can be transmitted over the network in a certain period of time
- It is sometimes useful to think of bandwidth in terms of **how long it takes to transmit each bit of data**

1 Mbps (each bit 1 µs wide)

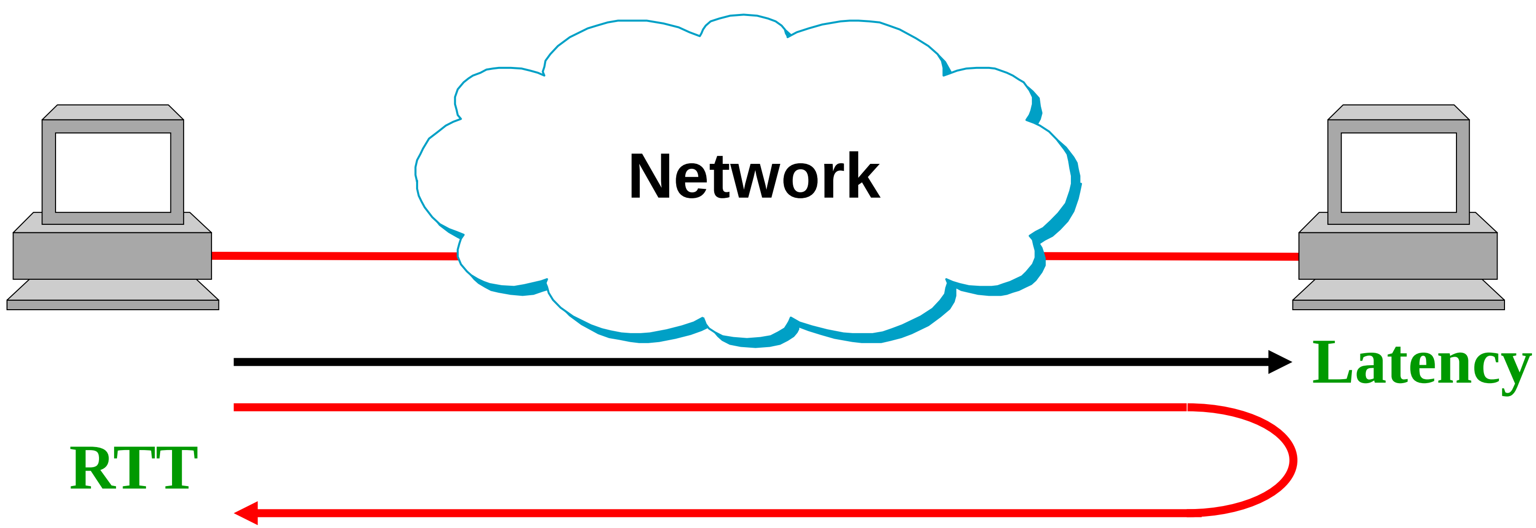


2 Mbps (each bit 0.5 µs wide)



Bandwidth and Latency

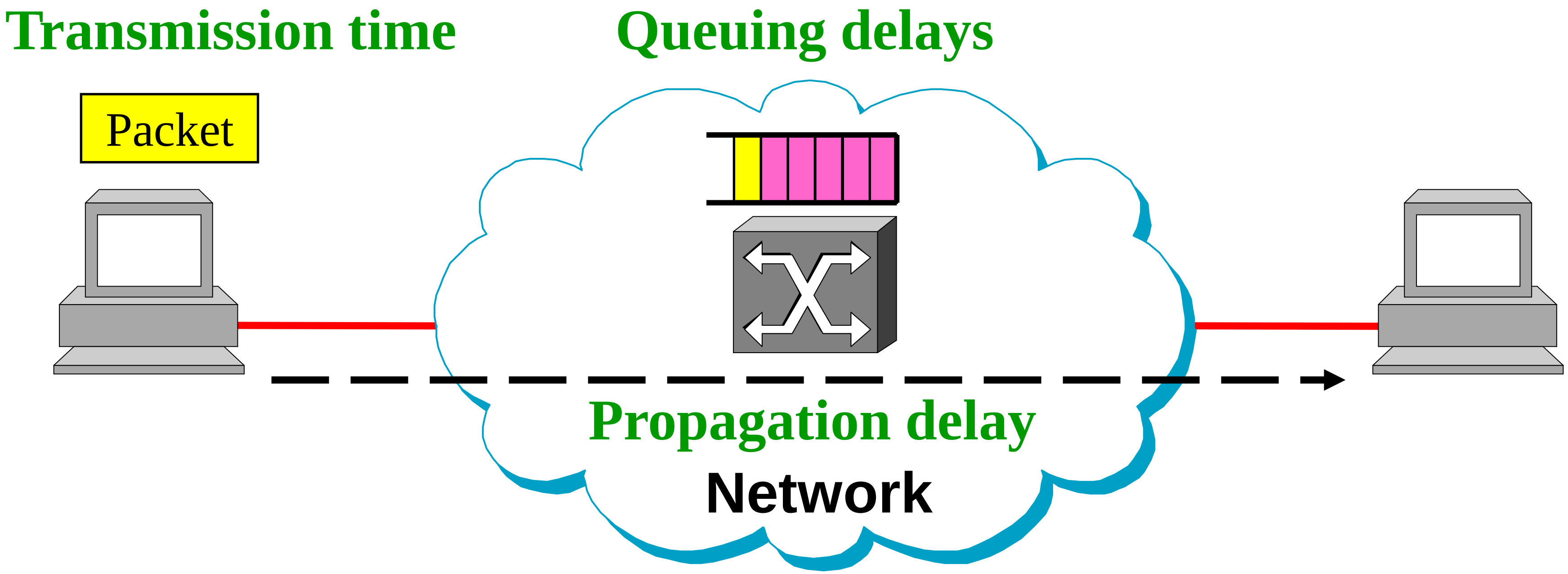
- **Latency:** corresponds to how long it takes a message to travel **from one end of a network to the other**
  - Measured strictly in terms of time
- **Round-trip time (RTT):** how long it takes to send a message from one end of a network **to the other and back**
- RTT may be approximated as 2×Latency



Bandwidth and Latency

- **Latency** consists of three components:
  - The speed-of-light **propagation delay**
    - 3.0×10^8 m/s in a vacuum, 2.3×10^8 m/s in a cable, and 2.0×10^8 m/s in a fiber
  - The amount of time it takes **to transmit a unit of data**
    - A function of the **network bandwidth** and the **size of the packet**
  - **Queuing delays** inside the network
    - Packet switches generally need to store packets for some time before forwarding them on an outbound link

Bandwidth and Latency



RTT: Queuing delays (Source → Destination) + Propagation delay

+ Queuing delays (Destination → Source) + Propagation delay

Transmission time is generally not included in RTT

Bandwidth and Latency

- The total latency is defined as

  Latency = Propagation + Transmit + Queue

  Propagation = Distance/SpeedOfLight

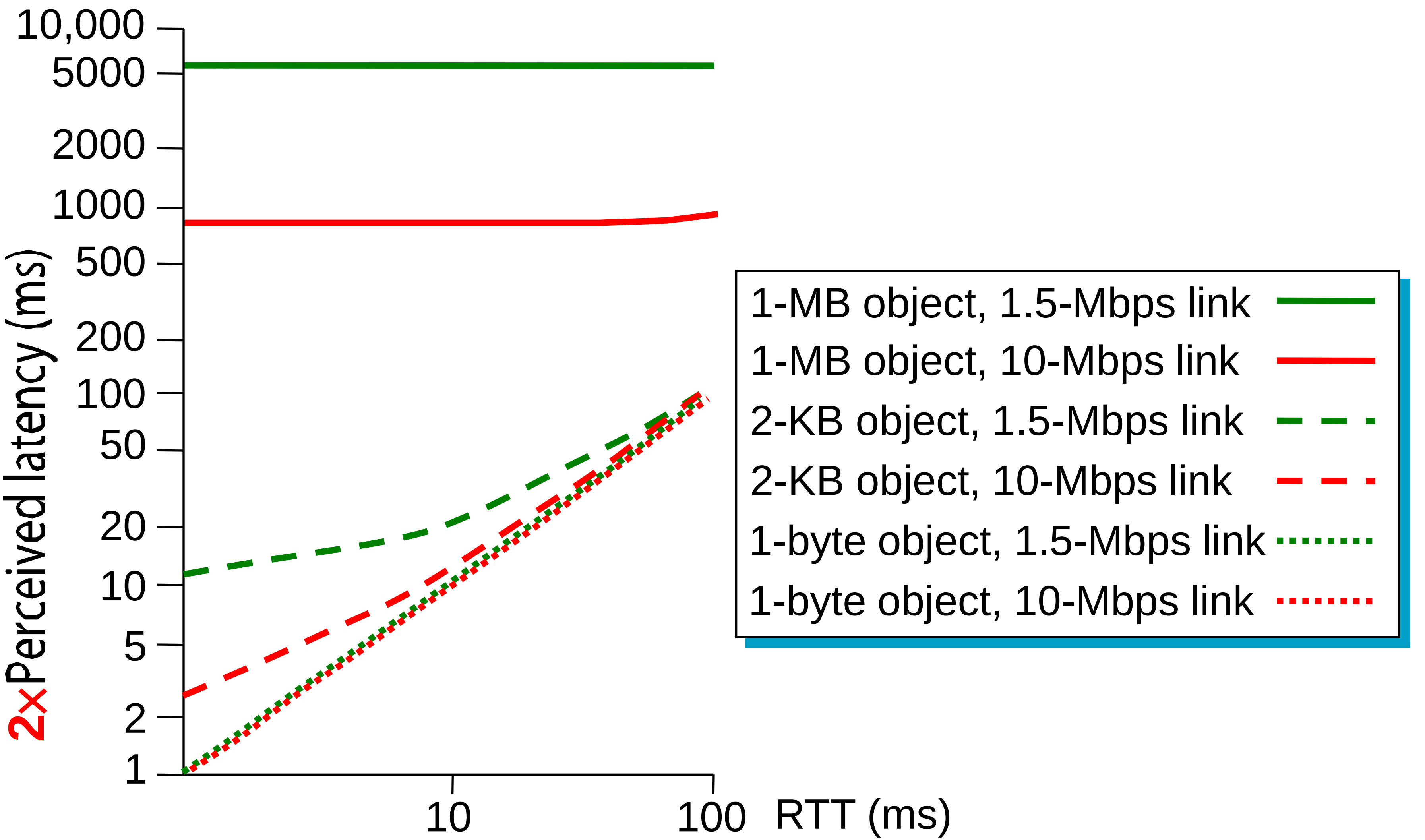
  Transmit = Size/Bandwidth

  – **Distance:** the length of the wire
  – **SpeedOfLight:** the effective speed of light
  – **Size:** the size of the packet
- **Bandwidth and latency** combine to define the **performance** characteristics of a given link or channel
- Their relative importance **depends on the application**

Link Performance

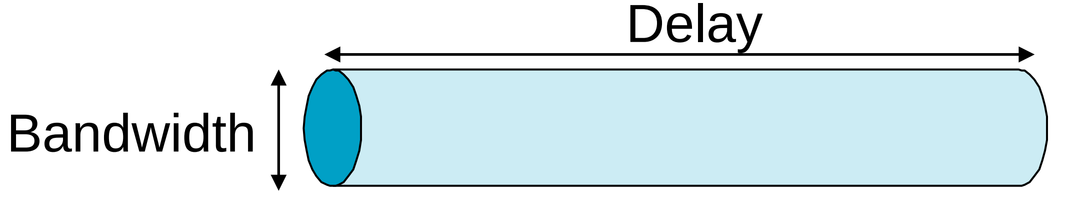
- Various message sizes: 1 byte, 2 KB, 1 MB
- RTTs: range from 1 to 100 ms (not including **Transmit time**)
- Link speeds: 1.5 or 10 Mbps
- For a 1-byte object (say, **a keystroke**):
  - 2×Latency remains almost exactly equal to the RTT
  - Cannot distinguish between 1.5-Mbps and 10-Mbps links
- For a 2-KB object (say, **an email message**):
  - Makes quite a difference on a 1-ms-RTT network
  - A negligible difference on a 100-ms-RTT network
- For a 1-MB object (say, **a digital image**):
  - The RTT makes no difference
  - It is the link speed that dominates performance

Link Performance



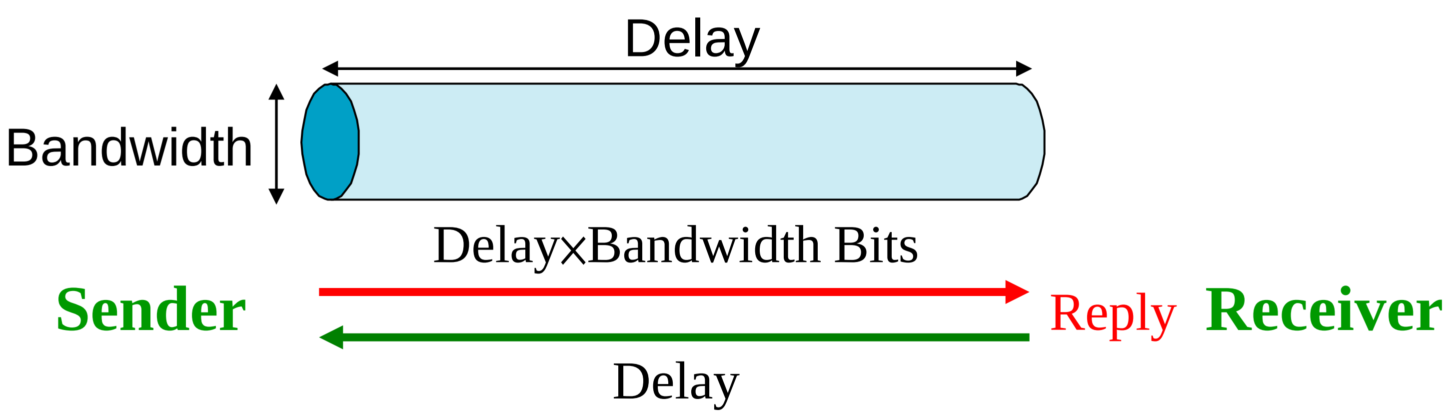
Delay × Bandwidth Product

- We think of a channel between a pair of processes as a pipe
  - The latency corresponds to the **length** of the pipe
  - The bandwidth gives the **diameter**
- The **delay × bandwidth product:** the **volume** of the pipe
  - **The number of bits it holds**
- For example, a channel with a one-way latency of 50 ms and a bandwidth of 45 Mbps is able to hold
  - 50×10^{-3} seconds $\times\ 45 \times 10^{6}$ bits/second $= 2.25 \times 10^{6}$ bits
  - Approximately 280 KB of data



Delay × Bandwidth Product

- The delay × bandwidth product is important for constructing high performance networks
  – How many bits the sender must transmit **before the first bit arrives at the receiver**
- If the sender is expecting the receiver to signal that bits are starting to arrive
  – The sender can send up to **2×delay×bandwidth** worth of data **before hearing from the receiver** that all is well



Delay × Bandwidth Product

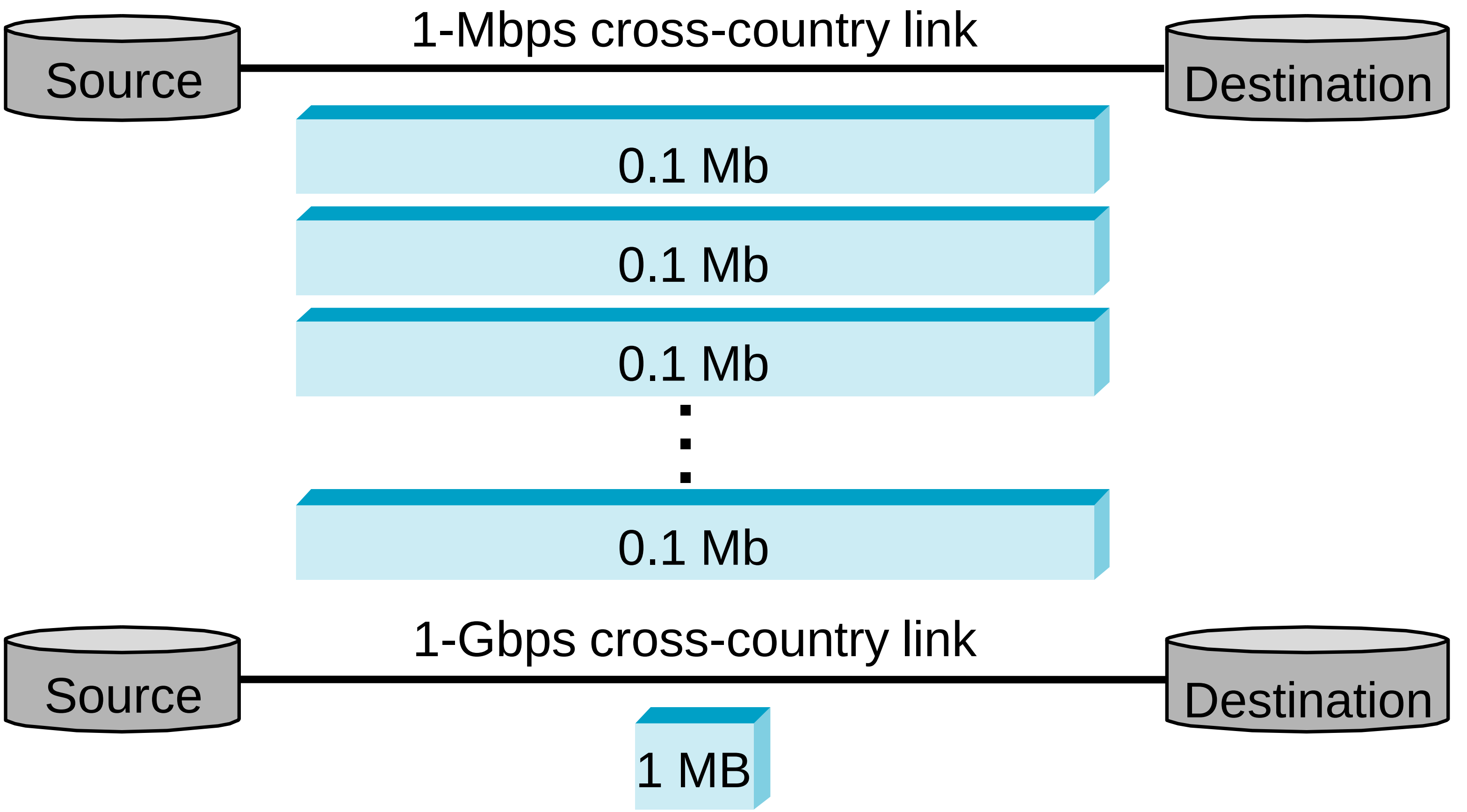
- The bits in the pipe are said to be "**in flight**"
- If the receiver tells the sender to stop transmitting
  - It might receive up to a delay × bandwidth's worth of data before the sender manages to respond
- On the other hand, if the sender does not fill the pipe, the sender will **not fully utilize the network**

High-Speed Networks

- The bandwidths available on today's networks are **increasing** at a dramatic rate
- "**High speed**" does not mean that latency improves at the same rate as bandwidth
  - **The RTT of a 1-Gbps link is the same as that of a 1-Mbps link**
- Consider what is required to transmit a **1-MB** file over a **1-Mbps** network versus over a **1-Gbps** network; RTT = 100 ms
  - 1-Mbps network: it takes 80 round-trip times to transmit the file; 1.25% of the file is sent in 1 RTT
  - 1-Gbps network: it takes 1/12.5 round-trip times to transmit the file; delay $\times$ bandwidth product = 12.5 MB

High-Speed Networks

- It looks like a stream of data transmitted across a 1-Mbps link
- It looks like a single packet on a 1-Gbps network



High-Speed Networks

- The effective end-to-end throughput that can be achieved is **Throughput = TransferSize/TransferTime**

  TransferTime = RTT + 1/Bandwidth × TransferSize

  - **RTT:** accounts for a **request message** being sent across the network and the **data** being sent back
- Transmit a 1-MB file across a 1-Gbps network with a round-trip time of 100 ms
  - The TransferTime = 100-ms RTT+ 8 ms = 108 ms
  - The effective throughput will be 1 MB/108 ms = 74.1 Mbps
  - **Transfer a larger amount of data will help improve the effective throughput** (fill the pipe)

Application Performance Needs

- Some applications are able to state an **upper limit** on how much bandwidth they need
- Just knowing the **average bandwidth** needs of an application will not always suffice
  - It is possible to put an **upper bound** on how big of a burst an application likely to transmit
- If this **peak rate** is higher than the available channel capacity
  - The excess data will have to be buffered somewhere
- In the case of **delay**, it sometimes **doesn’t matter** whether the one-way latency of the network is 100 ms or 500 ms
  - It does matter the **variation in latency**
    - The latency varies from packet to packet

Application Performance Needs

- **Interpacket gap:** the spacing between when packets arrive at the destination
- Such variation is generally not introduced in a single physical link, but it happen for a **multihop** packet-switched network **(queuing delay)**
- Suppose that the packets contain video frames
  - If a frame arrives **early**, it can simply be saved by the receiver until it is time to display it
  - If a frame arrives **late**, the video quality will suffer; it will not be smooth